MS-GLA: Multi-Scale Gated Linear Attention

Addressing Representational Bottlenecks via Multi-Temporal Resolution

Paper authors: Prasoon Dev, Anirudh Sankar, Vasudeva Varma

IIIT Hyderabad | COLM 2026

Research paper discussion | 12 October 2026

Distribute a fixed recurrent-state budget across update timescales.

arXiv:2609.35664v1 | 20–30 minute discussion

The question: can one update rate serve every timescale?

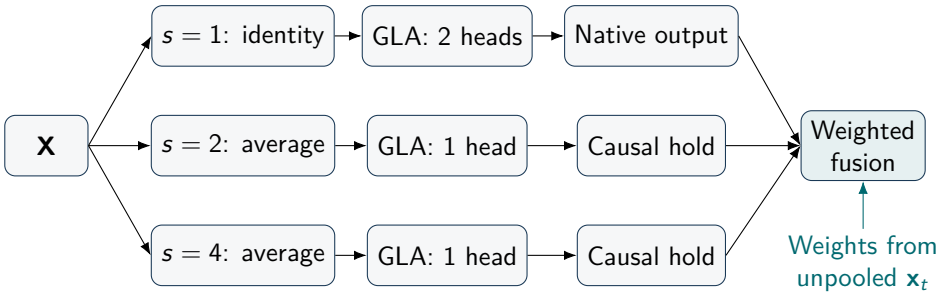
The architectural bottleneck

Every GLA head updates at every token, while a fixed-size state must retain both local detail and longer-range information.

- Input-dependent gates control retention at the current update rate.
- MS-GLA also changes how often each group of heads updates.
- Coarse summaries trade temporal detail for fewer recurrent updates.

Paper: Secs. 1–2, PDF pp. 1–4.

Three branches view the same sequence at different rates



All branches run; the fusion is dense over a small set of scales.

Teaching diagram based on paper, Sec. 3, PDF pp. 4–5.

GLA writes associations and decays the previous state

$$\mathbf{S}_t = \mathrm{Diag}(\alpha_t)\mathbf{S}_{t-1} + \mathbf{k}_t^{\top}\mathbf{v}_t, \qquad \mathbf{o}_t = \mathbf{q}_t\mathbf{S}_t$$

State and read/write vectors

$\mathbf{S}_t \in \mathbb{R}^{d_k \times d_v}$

$\mathbf{q}_t, \mathbf{k}_t \in \mathbb{R}^{1 \times d_k}$

$\mathbf{v}_t, \mathbf{o}_t \in \mathbb{R}^{1 \times d_v}$

Paper: Sec. 2.1, Eqs. (2), (5)–(6), PDF pp. 2–3.

- α_t: input-dependent, row-wise retention.
- $\mathbf{k}_t^{\top}\mathbf{v}_t$: new association.
- State size stays fixed as the sequence grows.

Pooling changes the time axis before GLA is applied

$$\mathbf{X}_i^{(s)} = \frac{\sum_{j=0}^{s-1} \mathbf{X}_{is+j} \mathbf{M}_{is+j}}{\max\left(1, \sum_{j=0}^{s-1} \mathbf{M}_{is+j}\right)}$$

- Input: $\mathbf{X} \in \mathbb{R}^{L \times d}$; mask $\mathbf{M} \in \{0,1\}^L$.
- Non-overlapping blocks; output length $\lceil L/s \rceil$, width d.
- Right padding handles incomplete blocks; the mask excludes padding.

Causality requires delayed release

The average of x_0, x_1, x_2, x_3 cannot affect predictions at positions 0, 1, or 2.

Paper: Sec. 3.1, Eq. (8), PDF p. 4.

The head budget is redistributed across scales

$$\sum_{s\in\mathcal{S}} h_s = H, \qquad \sum_{s} h_s d_k d_v = H d_k d_v$$

Variant	Scales	Heads per scale
Two-scale	$\{1,2\}$ or $\{1,4\}$	$[2,2]$
Coarse-only	$\{2,4\}$	$[2,2]$
Main model	$\{1,2,4\}$	$[2,1,1]$
Four-scale	$\{1,2,4,8\}$	$[1,1,1,1]$

Independent projections and gates; per-head state dimensions stay fixed.

State equality is exact under these dimensions; total parameters include small branch and fusion overheads.

Paper: Secs. 3.2, 4.1, PDF pp. 4–6. Parameter qualification: author code @ 45fa2abc4e3e.

A coarse output becomes visible only at the block end

Position t	0	1	2	3	4	5	6	7
Input	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$s = 4$ output	0	0	0	y_0	y_0	y_0	y_0	y_1

$$\widehat{\mathbf{Y}}_t^{(s)} = \begin{cases} \mathbf{0}, & t < s-1, \\ \mathbf{Y}_{\lfloor (t-s+1)/s \rfloor}^{(s)}, & t \geq s-1. \end{cases}$$

Repeat each coarse output s times, then shift right by $s-1$.

Paper: Sec. 3.3, PDF p. 5. Timeline and equivalent index formula are teaching aids.

The current token chooses the mixture of branch outputs

$$\mathbf{w}_t = \mathrm{Softmax}(\mathbf{W}_{\mathrm{fuse}}\mathbf{x}_t + \mathbf{b}_{\mathrm{fuse}}), \qquad \mathbf{O}_t = \sum_{s\in\mathcal{S}} w_{t,s}\widehat{\mathbf{Y}}_t^{(s)}$$

- $\mathbf{W}_{\mathrm{fuse}} \in \mathbb{R}^{|\mathcal{S}|\times d}$; the router reads unpooled $\mathbf{x}_t$.
- Each branch projects back to width d before fusion.
- Weights are nonnegative and sum to 1 at each position.

A different role from MoM

MS-GLA mixes temporal resolutions. MoM routes token updates to a sparse subset of memories.

Paper: Secs. 2.2, 3.3; Eqs. (10)–(11), PDF pp. 3, 5.

A slower update clock can extend the retention horizon

Teaching approximation: constant scalar retention $0 < a < 1$

After D original tokens, old information is multiplied approximately by

$$\underbrace{a^{D}}_{s=1} \qquad \text{versus} \qquad \underbrace{a^{D/s}}_{\text{scale } s}.$$

$$T_{1/2}^{(s)} \approx s\,\frac{\ln(1/2)}{\ln a}$$

- Assumes the same per-update retention; learned gates need not match.
- Pooling loses detail and introduces delay, motivating a fine branch.

Reader derivation for intuition, based on paper Eq. (6) and Secs. 3.1–3.3.

Shorter branch sequences preserve chunkwise training

Training

Each branch applies the existing GLA chunkwise algorithm to a contiguous sequence of length $\lceil L/s \rceil$.

Idealized head-update count for $[2, 1, 1]$:

$$\frac{2L + L/2 + L/4}{4L} = 68.75\%.$$

Streaming decoding

Keep recurrent state, pending tokens, and the most recent output for each branch.

Update a coarse branch only when its block is complete; otherwise hold its output.

Update counts exclude projections, MLPs, fusion, and execution overhead.

Paper: Sec. 3.4, PDF p. 5. Decode details: author code, ms_gla_layer.py @ 45fa2abc4e3e.

The experiment compares six approximately 340M models

- 24 layers, hidden width 1024, four total heads; GLA is the baseline.
- FineWeb-Edu; context 2,048; batch 32; accumulation 1; seed 42.
- Language modeling, zero-shot tasks, recall, position curves, routing, and training cost.

Training-budget records disagree

Main text: all models use 7B tokens. Appendix B:

$$\begin{aligned} \text{GLA}: &\quad 68{,}664 \times 32 \times 2{,}048 \approx 4.50\text{B}, \\ \text{MS-GLA}: &\quad 106{,}813 \times 32 \times 2{,}048 \approx 7.00\text{B}. \end{aligned}$$

Both public Table 1 CSVs point to `step-106813`; the appendix may be stale. Actual run metadata is needed to resolve this.

Paper: Sec. 4, PDF pp. 5–6; Appendix B, p. 13. Author result CSVs @ 45fa2abc4e3e.

Three scales give the lowest mean perplexity

Model	Wikitext PPL ↓	LAMBADA PPL ↓	Mean PPL ↓
GLA	19.63	22.26	20.94
$\{1, 2\}$	19.25	20.99	20.12
$\{1, 4\}$	**17.43**	21.69	19.56
$\{2, 4\}$	21.66	31.47	26.56
$\{1, 2, 4\}$	18.33	**19.59**	**18.96**
$\{1, 2, 4, 8\}$	18.80	20.18	19.49

- Mean PPL: 20.94 $\rightarrow$ 18.96, a 9.46% relative reduction.
- Four-task mean score: 41.94 $\rightarrow$ 42.85 ($+0.91$ pp).
- $\{1, 4\}$ wins Wikitext; the advantage depends on the task.

Paper: Sec. 5.1, Table 1, PDF pp. 6–7; means use the reported table values.

Recall improves by 1.40 F1 points in the three-task mean

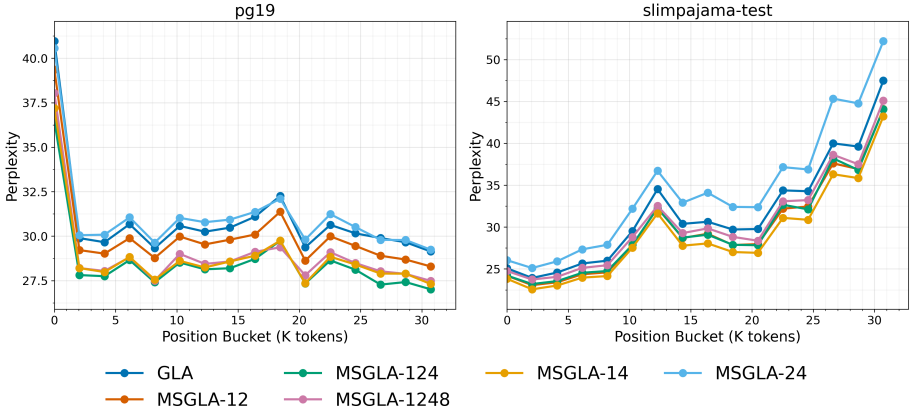
Model	SWDE	FDA	SQuAD	Mean F1 ↑
GLA	7.86	5.52	8.84	7.41
$\{1, 2\}$	9.26	5.95	9.79	8.33
$\{1, 4\}$	8.28	**6.06**	10.32	8.22
$\{2, 4\}$	5.96	5.42	8.21	6.53
$\{1, 2, 4\}$	**9.30**	5.98	**11.15**	**8.81**
$\{1, 2, 4, 8\}$	7.59	5.27	11.05	7.97

$$\frac{8.81 - 7.41}{7.41} \times 100\% \approx 18.9\%$$

Relative gain in the mean; absolute F1 remains low. Four scales reduce SWDE and FDA below GLA despite strong SQuAD.

Paper: Sec. 5.2, Table 2, PDF p. 7; all task columns report F1, higher is better.

Position curves show lower PPL, with shared fluctuations



PG19: lower curves, without sustained baseline drift.

SlimPajama: all variants rise at late positions; coarse-only is worst.

Paper: Fig. 1, PDF p. 8; original author figure, not redrawn.

Document position does not establish the available context

Public evaluator: independent sliding windows

Default `max_ctx = 2048`; each window calls

`model(chunk, use_cache=False)`

with no recurrent state passed from the previous window.

- A document can reach 32K while each prediction sees at most 2K.
- Saved curve JSONs omit the actual window/stride settings.
- Default settings cannot establish 15× continuous-state extrapolation; the exact run settings remain unresolved.

Needed: a full long-sequence forward pass or state-carrying chunks.

Paper: Sec. 5.3, PDF pp. 7–8. Supplement: author evaluator and README @ 45fa2abc4e3e.

The router's regional means stay almost unchanged

Region of sequence	$s = 1$	$s = 2$	$s = 4$
First 10%	52.64	18.41	28.95
Middle 10%	52.91	18.34	28.76
Last 10%	52.57	18.41	29.03

- Weights in percent for $\{1, 2, 4\}$; first-to-last shift ≤ 0.08 pp.
- Fine resolution has the largest mean weight in variants using $s = 1$.
- Coarse-only routing is also stable, despite poor model quality.

Interpretation

Stable regional averages support sustained average scale usage. They do not measure per-token routing diversity or causal contribution.

Paper: Sec. 5.4, Table 3, PDF p. 8.

The main model retains 92% of training throughput

Model	Tokens/second ↑	Peak memory, GiB ↓
GLA	31,989	70.2
$\{1, 2\}$	27,619	86.7
$\{1, 2, 4\}$	29,437	80.4
$\{1, 2, 4, 8\}$	27,201	89.2
$\{2, 4\}$	34,123	71.5

- Main configuration: 7.98% less throughput; 14.53% more peak memory.
- Fixed recurrent-state size does not fix training activation memory.
- Total training times require reconciled token counts and timing scope.

Paper: Sec. 5.5, Table 4, PDF p. 9; single RTX PRO 6000 Max-Q, Appendix B.

Scale coverage competes with branch capacity

The strongest ablation pattern

Removing $s = 1$ worsens mean PPL and recall F1:

$$18.96 \rightarrow 26.56$$

$$8.81 \rightarrow 6.53$$

$\{1, 2, 4\}$ versus $\{2, 4\}$.

Adding a scale changes two things

$\{1, 2, 4\} : [2, 1, 1]$

$\{1, 2, 4, 8\} : [1, 1, 1, 1]$

The fine branch loses a head while $s = 8$ is added. The experiment does not isolate these effects.

A token-level pathway is valuable in the tested designs; more scales are not automatically better.

Paper: Secs. 6.1–6.2, Tables 1–2, PDF pp. 7, 9–10.

The design is useful; the strongest claims need sharper evidence

1. **Contribution:** a concrete multi-rate GLA layer with fixed head/state budget and causal output alignment.
2. **Reported benefit:** lower mean PPL and higher recall F1 for a fine-plus-coarse allocation, with modest training overhead.
3. **Evidence limits:** one seed, GLA-only direct comparison, training-record inconsistency, and unresolved long-context protocol.

Do next

Reconcile checkpoint budgets; evaluate continuous long-context state; separate head allocation, pooling, and fusion in ablations.

Paper: Secs. 7–8, Appendix A, PDF pp. 10, 13; protocol qualifications from the preceding evidence.

Sources for the paper discussion

Primary paper

Prasoon Dev, Anirudh Sankar, Vasudeva Varma.
MS-GLA: Multi-Scale Gated Linear Attention for Addressing Representational Bottlenecks via Multi-Temporal Resolution.
arXiv:2609.35664v1, 28 September 2026. PDF marked COLM 2026.
https://arxiv.org/abs/2609.35664v1

Author implementation and supplementary records

github.com/prasoondev/msgla, pinned commit `45fa2abc4e3e`.
Layer/config files, Table 1 CSVs, long-document evaluators and saved curves. Accessed 12 October 2026.

Code observations supplement the paper; a repository snapshot does not identify every experimental run that produced it.

Backup: zero-shot gains are task dependent

Model	LAMBADA	PIQA	HellaSwag	Winogrande	Mean
GLA	19.33	64.20	33.94	50.28	41.94
$\{1, 2\}$	19.68	65.18	34.89	49.17	42.23
$\{1, 4\}$	19.89	63.87	**35.49**	51.07	42.58
$\{2, 4\}$	16.15	65.18	34.17	**52.57**	42.01
$\{1, 2, 4\}$	**20.69**	**65.83**	35.01	49.88	**42.85**
$\{1, 2, 4, 8\}$	20.34	65.34	34.53	49.01	42.31

- All values are percentage scores; higher is better.
- Public CSVs identify HellaSwag as `acc_norm`.
- Mean covers all four tasks, including Winogrande.
- $\{2, 4\}$ leads Winogrande despite its poor language-model PPL.

Paper: Table 1, PDF p. 7; metric clarification from pinned author CSVs.

Backup: optimization and hardware settings

Setting	Reported value
Optimizer	AdamW, $\epsilon = 10^{-15}$, weight decay 0.1
Peak learning rate	3×10^{-4}
Schedule	3,400 warmup steps; cosine to ratio 0.1
Gradient clipping	Maximum norm 1.0; skip non-finite steps
Tokens per step	$32 \times 2,048 = 65,536$; accumulation 1
Random seed	42; one training run per configuration
Tokenizer	`fla-hub/transformer-1.3B-100B`
Hardware	One NVIDIA RTX PRO 6000 Max-Q
Framework	`flash-linear-attention` + `flame`

Paper: Appendices A–B, PDF pp. 13–14; token-budget inconsistency discussed in the main slides.

Backup: matrix-state equality leaves small extra costs

$$d_k = \frac{1024 \times 0.5}{4} = 128, \qquad d_v = \frac{1024 \times 1}{4} = 256$$

$$\text{state elements per layer} = 4 \times 128 \times 256 = 131{,}072$$

- Branch s has h_s such states; their total is unchanged.
- Independent low-rank gate input projections add parameters.
- A three-scale fusion projection adds $3(1024 + 1) = 3,075$ parameters per layer.
- Cached decoding also holds incomplete blocks and previous outputs.

Derived from author config, ms_gla_layer.py and GLA layer source @ 45fa2abc4e3e.

Backup: wall-clock time and “TFLOPs” need context

Model	“TFLOPs”	Tokens/s	GiB	Training time
GLA	91.2	31,989	70.2	3d 1:49:25
$\{1, 2\}$	78.9	27,619	86.7	2d 18:23:08
$\{1, 2, 4\}$	84.0	29,437	80.4	2d 22:48:41
$\{1, 2, 4, 8\}$	77.8	27,201	89.2	2d 23:43:25
$\{2, 4\}$	97.4	34,123	71.5	2d 9:11:04

- No $\{1, 4\}$ cost row is reported.
- “TFLOPs” lacks a clear total/per-step/rate definition.
- Lower throughput with shorter total time is not a matched-work speedup without consistent budgets and timing scope.

Paper: Table 4, PDF p. 9; hardware issues noted in Appendix B, pp. 13–14.

Backup: key code evidence and how to interpret it

Training budget GLA CSV and MS-GLA CSV both name `step-106813`; this qualifies the appendix discrepancy.

Available context MS-GLA evaluator and GLA evaluator use independent windows; the default maximum is 2,048.

Run identification Saved long-document curves do not include window size or stride; exact Figure 1 settings are unresolved.

Parameter budget Layer code preserves per-head state dimensions while constructing independent branch projections and a fusion layer.

Supplementary evidence at author commit 45fa2abc4e3e; accessed 12 October 2026.