MoM: Linear Sequence Modeling with Mixture-of-Memories

Routing tokens into persistent recurrent states

Paper by Jusen Du, Weigao Sun, Disen Lan, Jiaxi Hu, and Yu Cheng

arXiv:2502.13685v4 | Revised November 18, 2025
Independent reading presentation | September 11, 2026

Expand memory through selective updates; retain linear sequence processing.

Primary materials: the 18-page v4 PDF, matching author TeX, and original figures. Main talk: about 25 minutes; backup follows slide 22.

Fixed states trade historical detail for efficiency

Transformer

Stores a key and value for each past token.

Historical storage and attention work grow with context length.

Linear recurrent model

Compresses history into a fixed state.

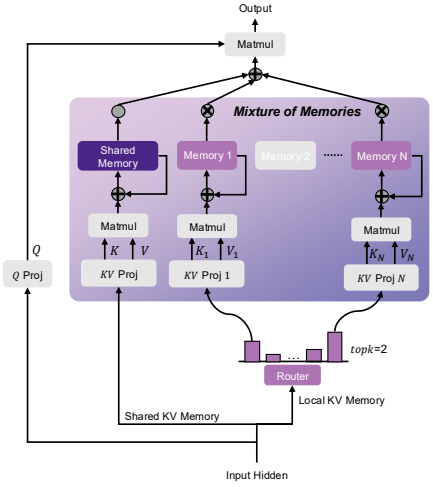
Later writes can interfere with information already stored there.

$$\mathbf{M}_t = \mathbf{M}_{t-1} + \mathbf{k}_t^\top \mathbf{v}_t, \qquad \mathbf{o}_t = \mathbf{q}_t \mathbf{M}_t$$

MoM asks which state a token should update, as well as how to update it.

Paper, Secs. 1–3.1, PDF pp. 1–3; Eq. (2). The displayed rule is the simplest illustrative update.

A router selects states; a shared state sees every token



- Each routed memory has its own KV projections and recurrent state.
- Only selected states update; all others remain unchanged.
- The query projection is shared.
- An extra shared memory provides a path through the full causal history.

Main configuration

4 routed memories, top-2 activation,
plus 1 shared memory.
Update rule: Gated DeltaNet.

Original Fig. 1, PDF p. 3; Secs. 3.2 and 4.1, PDF pp. 3–6.

Top-k routing chooses where to write and read

$$\text{scores}_t = \text{TopK}(\text{softmax}(\mathbf{x}_t \mathbf{W}_g)), \qquad \mathbf{g}_t = \frac{\text{scores}_t}{\sum \text{scores}_t}$$

- $\mathbf{x}_t \in \mathbb{R}^d$: current token representation; $\mathbf{W}_g \in \mathbb{R}^{d \times M}$: learned router.
- Keep the selected indices S_t, with $|S_t| = k$.
- Weights sum to one over the selected routed memories.

Teaching example: $M = 4$, $k = 2$

Softmax scores $(0.50, 0.30, 0.15, 0.05)$ select memories 1 and 2.
Their mixing weights are $(0.625, 0.375)$.

The paper uses these weights for mixing; Eq. (6) does not weight the write by $g_t^{(m)}$.

Paper, Sec. 3.2.1, Eqs. (3)–(4), PDF p. 4. Example constructed for this presentation.

Unselected memories skip the entire state update

$$\mathbf{k}_t^m = \mathbf{x}_t \mathbf{W}_k^m, \qquad \mathbf{v}_t^m = \mathbf{x}_t \mathbf{W}_v^m$$

$$\mathbf{M}_t^m = \begin{cases} F_m(\mathbf{M}_{t-1}^m, \mathbf{x}_t), & m \in S_t, \\ \mathbf{M}_{t-1}^m, & m \notin S_t. \end{cases}$$

- The generic F_m can use additive, gated, or delta-style updates.
- The main experiments instantiate F_m with Gated DeltaNet.
- A memory receiving tokens 1, 5, and 9 evolves on that subsequence: tokens 2–4 do not decay or overwrite its state.

Paper, Sec. 3.2.2, Eqs. (5)–(6), Table 1, PDF p. 4. Piecewise form makes the prose rule explicit.

Read a mixture without collapsing the stored states

$$\widetilde{\mathbf{M}}_t = \sum_{m \in S_t} g_t^{(m)} \mathbf{M}_t^m, \qquad \mathbf{o}_t = \mathbf{q}_t \widetilde{\mathbf{M}}_t$$

- Mixing is a temporary readout; the individual states persist.
- The same query can read each selected state separately:

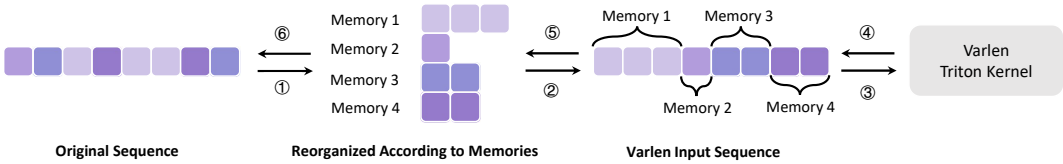
$$\mathbf{q}_t \left(\sum_m g_t^{(m)} \mathbf{M}_t^m \right) = \sum_m g_t^{(m)} (\mathbf{q}_t \mathbf{M}_t^m).$$

Shared path

Fig. 1 adds an always-active shared-memory readout to the routed mixture. It covers the full causal history, but remains a compressed state.

Paper, Eqs. (7)–(8), PDF p. 4; shared memory, p. 5; readout equivalence, Sec. 3.3, p. 5.

Stable grouping makes routed recurrence a varlen operation



1. Group by **sample and memory**, preserving time order within each group.
2. Concatenate groups with segment boundaries; run independent causal kernels.
3. Scatter outputs to original positions and combine with routing weights.

Top-k creates k routed copies of each token; segment boundaries prevent state leakage.

Original Fig. 2 and Sec. 3.3, PDF pp. 5–6. Figure shows top-1 and omits QKV projections; main experiments use top-2.

Linear in context length does not mean free memory expansion

Quantity	Simplified dependence
Training work vs. sequence length T	$O(T)$ for fixed M, k, d
Core state operations	$O(T(k+1)d^2)$
Router score computation	$O(TdM)$, plus selection and packing
Persistent recurrent state	$O((M+1)d^2)$

- Single-step decoding is constant in *past context length*; prefill still processes the prompt.
- Inactive states still occupy memory.
- Fixed activation ratio k/M means more active computation as M grows.

Reading analysis of Secs. 3.2–3.3, PDF pp. 3–6. Single-state dimension convention; not a full-model FLOPs estimate.

The size labels specify configurations, not equal budgets

	"380M"	"1.3B"
Layers / hidden size	24 / 1024	24 / 2048
Training tokens	15B	100B
Batch tokens	0.5M	2M
Pretraining context	2K	2K

- SlimPajama data; Mistral tokenizer; AdamW, learning rate 3×10^{-4}.
- Sparse KV projections add parameters. Main-table models are not exactly parameter matched.
- Several 1.3B baselines use public checkpoints marked †; authors report matched training configurations.

Paper, Sec. 4.1, PDF p. 6; Appendix C, p. 15. Exact 400M activated-parameter comparison appears in Table 9.

MoM narrows the recall gap at both reported scales

Model	380M / 15B	1.3B / 100B
Transformer++	31.70	37.31[†]
Gated DeltaNet	24.78	32.30
MoM	**28.16**	**36.04**
MoM minus Gated DeltaNet	+3.38	+3.74
MoM minus Transformer++	−3.54	−1.27

Scope of this result

Mean over FDA, SWDE, SQuAD, NQ, TriviaQA, and DROP.
All inputs are truncated to at most 2K tokens. Higher is better.

The paper does not fully specify each task's scoring metric; differences are reported in score points.

Paper, Table 2, PDF p. 7. Differences calculated from published means. No confidence intervals reported.

The remaining gap depends strongly on the recall task

1.3B / 100B tokens	MoM	Transformer++†	Difference
FDA	41.14	44.32	−3.18
SWDE	34.30	32.43	+1.87
SQuAD	37.08	42.59	−5.51
NQ	24.11	24.49	−0.38
TriviaQA	58.59	58.47	+0.12
DROP	21.03	21.56	−0.53

- MoM beats the listed linear models on all six tasks at both scales.
- At 380M, FDA remains 22.98 vs. 46.14 for Transformer++.
- Excluding FDA, the 380M means become 29.20 vs. 28.81.

Paper, Table 2, PDF p. 7. All six-task comparisons use the same maximum input length of 2K.

Recall gains survive matching activated parameters

Model	Activated params	Recall ↑	Common ↑
Gated DeltaNet	400M	24.78	41.44
MoM	400M	**26.51**	**41.50**
Difference		+1.73	+0.06

- Reduce MoM's MLP hidden ratio to 2; otherwise follow the main training setup.
- All six recall scores improve in this comparison.
- The common-sense mean changes very little; two individual tasks decline.

What this control establishes

The recall benefit is not explained solely by more activated parameters. Total parameters, total state, FLOPs, and wall time are not jointly matched.

Paper, Appendix G.1, PDF pp. 16–17; Table 9, p. 17. Recall differences: points; common-sense accuracy: percentage points.

Routed memories beat a single expanded active state

Architecture	Listed Params	Recall ↑	Common ↑
GLA expanded	425M	22.87	38.90
GLA MoM	395M	**23.53**	**39.99**
Gated DeltaNet expanded	550M	26.32	41.32
Gated DeltaNet MoM	444M	**28.16**	**41.97**

- Expand the baseline value dimension to match **activated state capacity**.
- Benefits appear with both GLA and Gated DeltaNet updates.
- MoM still retains additional inactive states; this is not equal total state storage.

Paper, Table 4 and Sec. 4.2.3, PDF pp. 7–8; Appendix G.2, p. 17. "Params" retained from the table; Appendix G discusses activated parameters.

LongBench gains concentrate in few-shot tasks and code

Model	Summary	Few-shot	Synthetic	Code	Avg.*
GSA†	**7.75**	20.29	1.92	42.83	14.61
Gated DeltaNet	7.14	18.00	2.10	41.52	13.98
MoM	6.89	**21.26**	**2.63**	**47.79**	**15.64**

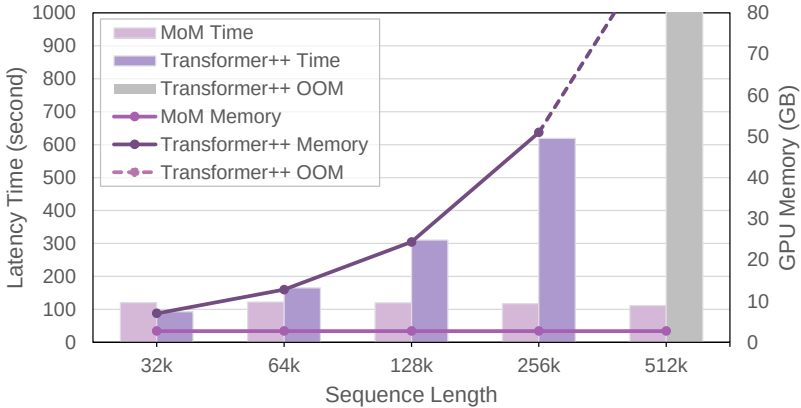
- Code improves by 6.27 points over Gated DeltaNet; Transformer++ is not included.
- It is not best on summary, single-doc QA, multi-doc QA, or synthetic tasks across the full table.

Aggregation needs clarification

Table 10 reports MoM Zh-Avg 17.33, En-Avg 15.71, but Avg. 15.64. These cannot be a simple partition-weighted average on the same basis.

Paper, Tables 3 and 10, PDF pp. 7 and 18. *Author-reported overall average; weights are not reconstructible. Input limits and model scale are not separately specified here.

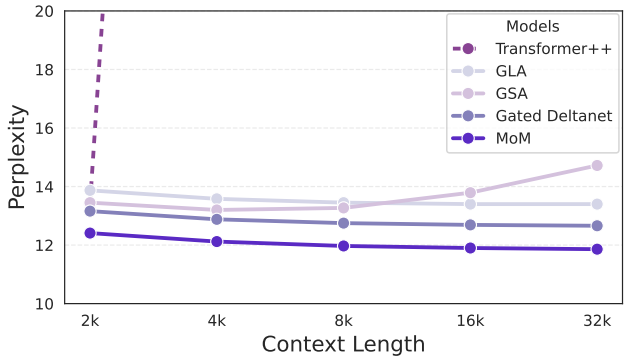
Long-history decoding benefits; the shortest point does not



- Generate 1K tokens: MoM remains roughly flat over 32K–512K history.
- Transformer++ is faster at 32K, then grows; 512K is marked OOM.

Original Fig. 3, PDF p. 8. Bars: seconds (left); lines: GB (right). Exact inference batch, precision, and prefill timing are not fully reported. OOM markers are not measured values.

Perplexity extrapolates well beyond the training length



- Train at 2K on SlimPajama; evaluate 2K–32K contexts on FineWeb.
- MoM has the lowest plotted PPL. This does not establish exact long-range retrieval or compare all long-context Transformer adaptations.

Original Fig. 4 and Sec. 4.2.5, PDF p. 8. Lower PPL is better. Transformer++ leaves the displayed range.

A shared state helps; the best balance weight is task dependent

Auxiliary loss scale

Weight	Recall	Common
10^{-2}	27.59	**42.10**
5×10^{-3}	26.55	41.71
10^{-3}	**28.16**	41.97
0	27.23	41.58

Shared memory

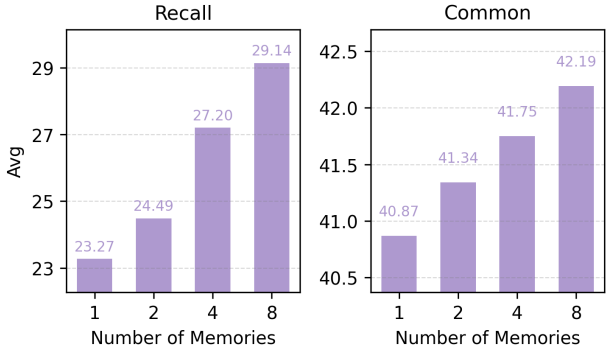
	Recall	Common
With	28.16	41.97
Without	26.06	40.38

Shared path: +2.10 recall points.

- Routing heatmaps show much better load balance with the auxiliary loss.
- Better balance does not guarantee higher recall for every loss weight.
- Removing shared memory also removes capacity and computation.

Paper, Table 6, PDF p. 10; Figs. 5 and 9, pp. 9 and 18. 380M configuration, 15B training tokens; higher scores are better.

Memory scaling helps through eight states in the main plot



- At activation ratio 0.5, more memories also mean more active updates.
- The 1-memory point is a special starting baseline.
- Appendix Fig. 8 shows a recall decline from 8 to 16 at 50% activation.

Original Fig. 6, PDF p. 10; Fig. 8, p. 17. The 4-memory scores (27.20 / 41.75) differ from the main model (28.16 / 41.97); the paper does not fully explain this configuration difference.

Common-sense gains are modest against the strongest baseline

1.3B / 100B tokens	Wiki PPL ↓	Lambada PPL ↓	Common ↑
Transformer++[†]	17.61	19.29	49.93
GSA[†]	16.69	16.02	50.89
Gated DeltaNet	17.14	18.80	49.58
MoM	**16.64**	**14.83**	**50.97**

- Mean accuracy is 1.39 percentage points above Gated DeltaNet, but only 0.08 above GSA.
- Individual task leaders differ; no uncertainty estimates are reported.
- Six-task mean: ARC-e/c, HellaSwag, LAMBADA, PIQA, WinoGrande. PPL is excluded from this mean.

Paper, Appendix D, Table 7, PDF p. 15. Zero-shot via lm-evaluation-harness; ARC-c and HellaSwag use length-normalized accuracy scoring.

Three global-attention layers unlock further recall gains

24 layers = 3 repeats of [7 MoM layers + 1 Transformer layer]

Model	Recall ↑	Common ↑
Transformer++	31.70	41.35
MoM	28.16	41.97
MoM Hybrid	**37.80**	**42.38**

- Hybrid recall: +9.64 points over pure MoM, +6.10 over Transformer++.
- FDA and SWDE account for much of the gain; not every task improves.

The complexity claim changes

Global-attention layers retain growing KV caches. The full hybrid does not inherit pure MoM's constant-history-state guarantee.

Paper, Appendix F, Table 8, PDF p. 16. Baseline rows match the small-scale main experiment; full hybrid compute and parameter budgets are not separately listed.

The evidence supports useful separation, not lossless memory

1. **Mechanism:** unselected states avoid the current write; interference within a state and routing mistakes remain possible.
2. **Budgets:** activated parameters, active state capacity, total state, and runtime are different controls.
3. **Reliability:** no multi-seed error estimates; some common-sense margins are very small.
4. **Reporting:** LongBench aggregation, scaling configuration differences, and inference details need clarification before reproduction.

Reading assessment based on Sec. 4 and Appendices C–I, PDF pp. 6–10 and 15–18. No training or evaluation was rerun for this presentation.

Selective persistent state is a useful design axis

Take-home message

MoM routes tokens into independently updated recurrent states, improving recall while preserving linear sequence processing for a fixed architecture.

- Strongest result: recall gains remain after matching activated parameters.
- Main trade-off: more stored states, routing overhead, and extra active work.
- Practical complement: a few attention layers can add substantial recall.

Primary reference

Jusen Du, Weigao Sun, Disen Lan, Jiaxi Hu, and Yu Cheng.
MoM: Linear Sequence Modeling with Mixture-of-Memories.
arXiv:2502.13685v4, November 18, 2025.

Prepared from the matching PDF, author TeX, and original figures. Equations and examples marked as teaching explanations are this reading's reconstructions. Backup slides follow.

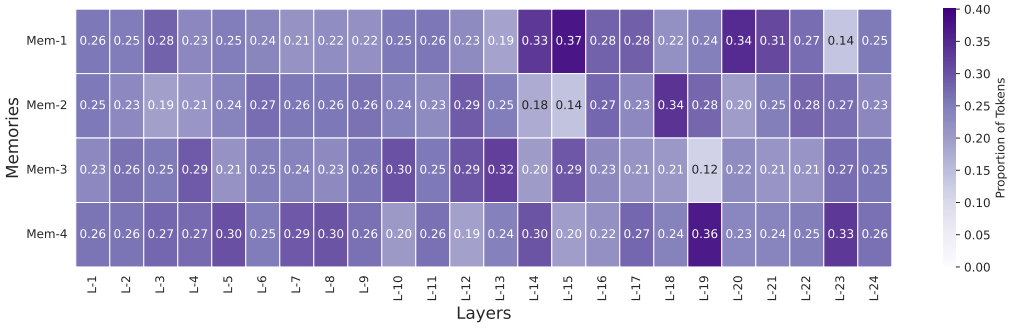
Backup: token preferences suggest possible specialization

Memory	Observed token types	Proposed function
1	Basic nouns / verbs / prepositions	Basic semantics / causal logic
2	Proper nouns / scientific terms	Specialized knowledge
3	Technical terms / adjectives	Detailed memory
4	Questions / incomplete nouns	Fragmented or open information

- Qualitative observation of an intermediate layer.
- Token-selection procedures and cross-seed stability are not established.
- A functional claim would require targeted state intervention or controlled routing experiments.

Paper, Sec. 4.2.6 and Table 5, PDF pp. 8–9. Proposed functions are author interpretations, not causally demonstrated modules.

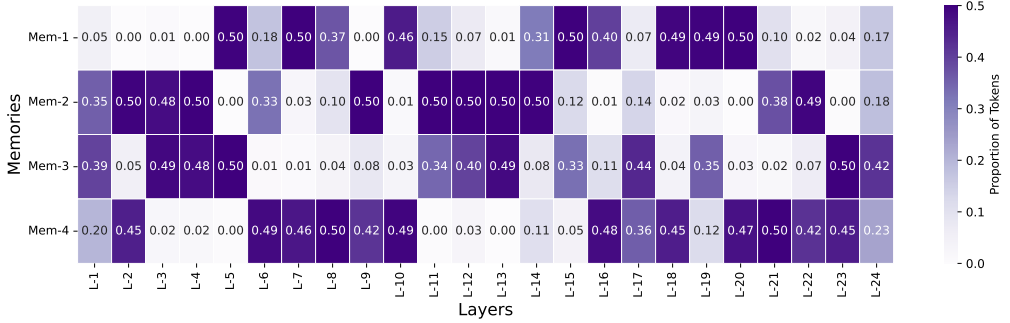
Backup: auxiliary loss spreads routing across the memories



- Columns: 24 layers. Rows: four routed memories.
- Roughly 300k ARC-easy tokens; many shares are near 0.25.
- Near-balanced shares still vary: the figure includes values from 0.12 to 0.37.

Original Fig. 5, PDF p. 9; Sec. 4.2.6, p. 8. Columns sum to about 1, suggesting shares of routing assignments rather than per-token selection probabilities. The denominator is not explicitly defined.

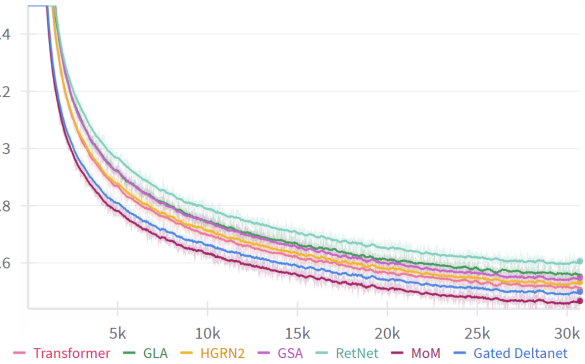
Backup: without auxiliary loss, some states are nearly unused



- Layer 2 shares: 0.00, 0.50, 0.05, 0.45.
- Color scale reaches 0.50 here, versus 0.40 in Fig. 5; compare numbers.
- The main text labels both heatmaps "with" loss; this figure's caption explicitly says **without**.

Original Fig. 9, PDF p. 18, and Sec. 4.2.6, p. 8. Unbalanced usage can reduce effective capacity, but the heatmap alone is not a downstream causal test.

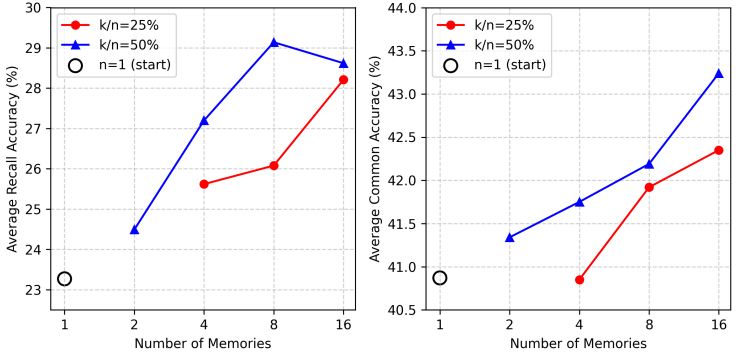
Backup: the training-loss advantage is a single reported run



- 380M configuration, 15B training tokens, random seed 42.
- Lower training loss supports this optimization trajectory; the plot is not a throughput measurement or a multi-seed confidence interval.

Original Fig. 7, Appendix E, PDF pp. 15–16. The horizontal scale reaches about 30k, consistent with 15B / 0.5M batch tokens; the original axis has no explicit unit title.

Backup: recall is not monotonic through sixteen memories



- Red: 25% activation; blue: 50%; open circle: one-memory start.
- The blue recall curve drops from 8 to 16 memories; the common-sense curve rises. Every setting includes an additional shared memory.

Original Fig. 8, Appendix H, PDF p. 17. The legend's *n* denotes memory count. Points without numeric labels are not treated as exact tabulated results.