Where You Put Attention Matters More Than How Much You Use: Attention Placement in Hybrid Sequence Models

Anonymous authors
Paper under double-blind review

Abstract

Hybrid sequence models mix recurrent blocks with a small number of attention blocks, but most design studies ask how much attention to use, not where to put it. We compare six six-block models with exactly matched parameter counts. The models differ only in the positions of their attention blocks. Each configuration is trained with five seeds on an NVIDIA RTX A6000 and tested on parity state tracking, the non-commutative A_5 task, and multi-query associative recall. On parity, placing one attention block at the front or middle gives a median L_{90} of 192, compared with 96 for the recurrent baseline and 128 for back placement. On recall, the middle layout reaches 0.968 accuracy while the front layout reaches only 0.091. A paired knockout test explains the difference: removing attention after training changes parity accuracy from 1.000 to 0.990 ($p = 0.2500$), but reduces recall from 0.578 to 0.008 ($p = 1.9 \times 10^{-6}$). Thus, attention is needed during inference for recall, while on parity it mainly changes what the recurrent blocks learn during training. Middle placement is the only layout that is strong on both tasks.

1 Introduction

Hybrid sequence models combine two kinds of blocks. Recurrent blocks process long sequences efficiently and can store a running state. Attention blocks can retrieve information from earlier tokens, but they are more expensive. Deployed systems therefore use only a few attention blocks, often near the middle of the network Lieber et al. (2025); Ren et al. (2025); NVIDIA (2025b); Yang et al. (2025); Falcon-LLM Team (2025).

This common design was chosen using language-model loss, few-shot accuracy, retrieval, and speed NVIDIA (2025a); Wang et al. (2025). It was not tested on state tracking, where a model must update and preserve a compact state at every token. At the same time, work on recurrent architectures is explicitly trying to improve this capability Merrill et al. (2024); Sarrof et al. (2024); Grazzi et al. (2025); Siems et al. (2025); Lahoti et al. (2026).

The missing question is simple: if a hybrid has one attention block, does its position change what the recurrent blocks learn? Prior work identifies this as an open problem Jiang (2026).

1.1 Research questions

RQ1 With parameters, depth, width, and training budget fixed, does the position of attention change state-tracking accuracy at lengths longer than those seen in training? (§5)

RQ2 Does the same position work best for state tracking and associative recall? (§6)

RQ3 After training, is attention still needed to run the state-tracking computation, or did it mainly help the recurrent blocks learn? (§7)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1.2 Contributions

1. The knockout test separates the two roles of attention (RQ3). Removing every attention mixer after training changes parity accuracy only from 1.000 to 0.990 (paired sign-flip $p = 0.2500$), but reduces mqar recall from 0.578 to 0.008 ($p = 1.9 \times 10^{-6}$). Attention is required at inference for recall. On parity, its larger role is during training, when it changes what the recurrent blocks learn (§7).
2. Position changes parity extrapolation (RQ1). Median L_{90} is 96 for the recurrent baseline, 192 with attention at the front, 192 in the middle, and 128 at the back. Adding a second attention block reaches only 96. We report all exact pairwise tests with Holm correction instead of presenting only a winner (§5).
3. The best position depends on the task (RQ2). Front attention performs well on parity but reaches only 0.091 on recall. Middle attention reaches 0.968 on recall and also performs well on parity. It is the only layout near the top of both axes (§6).

We expected attention to hurt length extrapolation by giving the optimizer a short, length-specific solution Liu et al. (2023); Phan et al. (2025). The parity results show the opposite at this scale. We also report the results that limit this conclusion: no layout solves A_5, and device choice changes some seed-level outcomes.

2 Related Work

Hybrids are what ship. Meanwhile, essentially every deployed linear-recurrent language model is a hybrid: Jamba Lieber et al. (2025), Samba Ren et al. (2025), Nemotron-H NVIDIA (2025b), Gated DeltaNet with sliding-window attention Yang et al. (2025), and the parallel-head design of Falcon-H1 Falcon-LLM Team (2025). The systematic design studies that fixed the recipe, namely attention ratio around 1:5 with attention blocks placed in the middle rather than at the front NVIDIA (2025a), and the parallel sweep of Wang et al. (2025), optimise perplexity, few-shot accuracy, needle-in-a-haystack recall and throughput. Neither contains a formal-language, parity or state-tracking evaluation. The mechanistic account of why a few attention heads are added, a gather-and-aggregate circuit that recurrent backbones lack Bick et al. (2025), is likewise a story about recall.

The gap. Jiang Jiang (2026) states explicitly that whether hybrids inherit the layerwise state-encoding behaviour observed in pure models is left open; Chung et al. Chung et al. (2026) find that expressive capacity does not predict state-tracking robustness, again across pure models only; and Rawat et al. Rawat et al. (2026) ask in their own limitations for the controlled analysis we run. The closest prior work is Phan et al. Phan et al. (2025), whose recurrent backbone is an LSTM (already outside TC^0 and already at 100% on Chomsky-hierarchy parity), whose state-tracking task is a commutative modular sum, and whose contribution is a training-schedule fix rather than a measurement of the design trade-off. No prior study holds parameter count exactly constant while varying placement, and none reports an inference-time knockout probe.

3 Experimental Setup

Blocks. We use two block types in a pre-norm residual stack with RMSNorm Zhang & Sennrich (2019). The R-block is a diagonal input-dependent linear recurrence $h_t = a_t \odot h_{t-1} + b_t \odot v_t$ with state-transition eigenvalues starting from $a_{\text{raw}} = 2\sigma(z_t) - 1$ and then applying the magnitude floor described below. The sign can therefore be positive or negative, which is the feature Grazzi et al. Grazzi et al. (2025) show unlocks parity tracking. A gated output projection follows the recurrence. The A-block is 4-head causal softmax attention Vaswani et al. (2017) with rotary position embeddings Su et al. (2024), followed by an MLP.

Iso-parameter matching. An R-block's mixer has $5d^2$ parameters against an A-block's $4d^2$. We compensate in the MLP width, setting $d_{\text{ff}}^A = d_{\text{ff}}^R + d/2$, so that a block of either type has

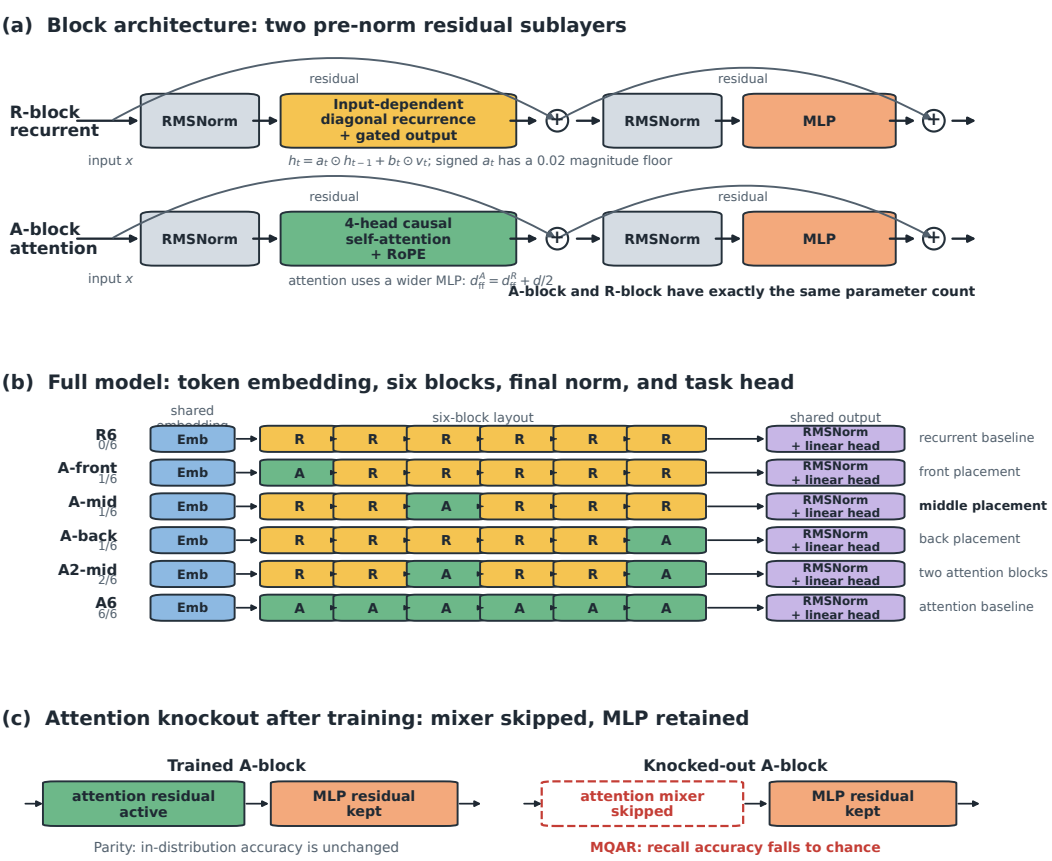

Figure 1: The source-verified model architecture. (a) Each block has two pre-norm residual sublayers. The first uses either an input-dependent diagonal recurrence with a gated output or 4-head causal attention with rotary position embeddings. The second uses an MLP. The attention MLP is wider by $d/2$, which gives the two block types the same parameter count. (b) Every model maps token IDs through a shared embedding, one of six block layouts, a final RMSNorm, and a linear task head. Only the positions and number of A-blocks change. (c) The knockout skips only the attention mixer after training; the A-block's MLP residual remains active. The figure matches models.py.

exactly the same parameter count. Every configuration in Fig. 1consequently has 226,112 parameters, not approximately but exactly (226,112 on A_5; the value differs between tasks only through the embedding and head, which depend on the task vocabulary), so the only thing that varies across the grid is where attention sits.

Tasks. Task instances are created during each run, so no dataset download is needed. Parity is a regular task from the Chomsky-hierarchy suite Delétang et al. (2023), with the running parity as the per-position target. The A_5 word problem composes a sequence of generators of the 60-element alternating group and asks for the running product; it is non-commutative, and is the task the expressivity literature Merrill et al. (2024); Grazzi et al. (2025); Siems et al. (2025) identifies as discriminating. MQAR is multi-query associative recall from Zoology Arora et al. (2024a), the task that defines the recall–efficiency frontier this design space is usually judged on Arora et al. (2024b). It is our recall axis, with 8 key–value pairs and a vocabulary of 256.

Protocol. Following Delétang et al. (2023); Grazzi et al. (2025) we train on lengths drawn uniformly from $[3, 40]$ and evaluate on unseen lengths up to 256, reporting final-token accuracy. Optimisation is AdamW Loshchilov & Hutter (2019) with a one-cycle schedule, peak

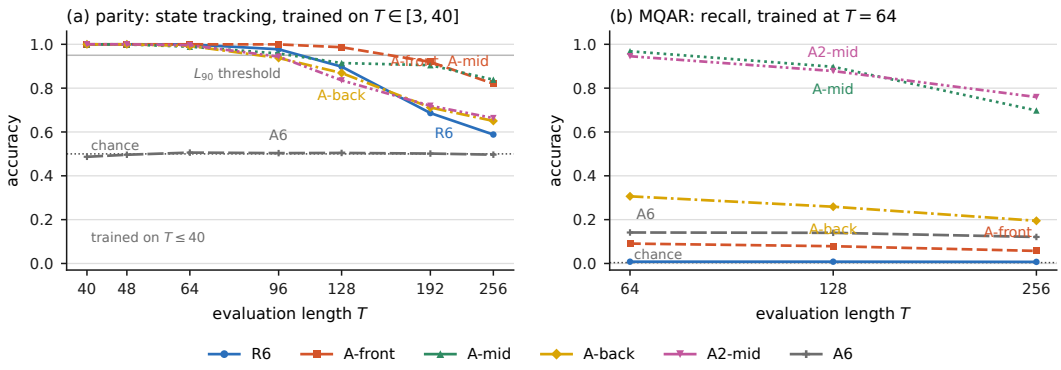

Figure 2: Accuracy against evaluation length, mean over seeds. (a) parity: all six configurations containing at least one R-block are at ceiling in distribution, so the entire comparison is extrapolation. Attention at the front or middle holds accuracy furthest; at the back it does not, and the pure transformer never learns the task at all. (b) MQAR: the ordering is close to reversed. A-front is barely above the pure recurrent model, which fails outright.

Table 1: Multi-query associative recall (mqar), the recall axis. 8 key–value pairs, vocabulary 256, trained at $T = 64$ and evaluated in and out of distribution. Cells are mean query-position accuracy over 5 seeds.

Config	T=64	T=128	T=256	L_{90}	knockout
R6	0.008	0.008	0.007	0	0.008
A-front	0.091 $\pm$0.01	0.079 $\pm$0.01	0.057 $\pm$0.01	0	0.008
A-mid	0.968 $\pm$0.03	0.897 $\pm$0.06	0.698 $\pm$0.09	128	0.008
A-back	0.306 $\pm$0.35	0.259 $\pm$0.34	0.195 $\pm$0.29	0	0.008
A2-mid	0.946 $\pm$0.03	0.878 $\pm$0.03	0.759 $\pm$0.04	64	0.007
A6	0.141	0.140 $\pm$0.01	0.121	0	0.008

learning rate 3×10^{-3}, weight decay 0.1, gradient-norm clipping at 1.0, batch size 32, and 2000 steps (1500 for MQAR). We use five seeds for every task and layout. We report mean accuracy at each length and the full per-seed L_{90} values, where L_{90} is the longest tested length with scaled accuracy at least 0.9. The main summary is the median L_{90} across seeds, so one lucky run cannot define a layout.

Statistical protocol. Configuration contrasts on parity use exact two-sided Mann–Whitney U tests on final-token accuracy at $T = 256$. The knockout probe uses an exact paired sign-flip test on the within-run accuracy change (Table 3). We use exact tests because five observations per layout are too few for a reliable large-sample approximation.

The knockout probe. After training, we replace every A-block's attention output with a residual passthrough and re-measure in-distribution accuracy. If a hybrid's recurrence has genuinely learned the automaton, knockout should cost little; if attention has displaced the recurrent algorithm with a shortcut, in-distribution accuracy should collapse. This is a forward-pass-only diagnostic and costs seconds.

Compute. The primary analysis is 90 runs trained on a NVIDIA RTX A6000 with CUDA 12.4 and PyTorch 2.6.0+cu124. The sum of the per-run training times is 89 minutes. The earlier CPU records are retained as a separate cross-device comparison and are never pooled with the GPU runs.

Table 2: The trade-off, read off the two axes. Median L_{90} on parity (5 seeds) and mean parity accuracy at T=256 are the state-tracking axis; in-distribution mqar accuracy is the recall axis. Values after ± are the seed standard deviation, which is large for A-mid on parity and A-back on mqar: those two orderings are not resolved by this data.

Config	parity L_{90}	parity @256	mqar @64
R6	96	0.588 ±0.06	0.008 ±0.00
A-front	192	0.820 ±0.13	0.091 ±0.01
A-mid	192	0.839 ±0.20	0.968 ±0.03
A-back	128	0.650 ±0.16	0.306 ±0.35
A2-mid	96	0.663 ±0.19	0.946 ±0.03
A6	0	0.496 ±0.03	0.141 ±0.00

Table 3: The knockout probe uses paired evidence. Exact two-sided paired sign-flip test on in-distribution accuracy before and after replacing every attention mixer with a residual passthrough, paired within run and pooled over the four hybrid configurations. This test asks whether the observed within-run changes could arise if their signs were randomly reversed. Tied runs contribute zero.

Task	n runs	full	knocked out	exact p
parity	20	1.000	0.990	0.2500
mqar	20	0.578	0.008	1.9×10^{-6}

4 Validation Against Published Baselines

Before comparing layouts, we check that the two endpoints reproduce the expected qualitative behavior because both tasks and the recurrent layer are reimplemented here.

Our pure-transformer stack A6 reaches in-distribution parity accuracy 0.506 against the 52.0% reported for a Transformer on the Chomsky-hierarchy Parity Check task Delétang et al. (2023); our pure recurrent stack R6 reaches 1.000, matching the scaled accuracy of 1.000 reported for Mamba with eigenvalues in $[-1, 1]$ Grazzi et al. (2025); and R6 attains mqar accuracy 0.008, the total failure Zoology reports for recurrent models at small model dimension Arora et al. (2024a). The two endpoints therefore sit where the literature places them, in opposite corners. We did not run a learning-rate sweep in the A6000 replication, so all comparisons are conditional on the shared schedule.

5 The State-Tracking Axis

Fig. 2(a) gives the parity accuracy-against-length curves.Every configuration containing at least one R-block reaches perfect in-distribution accuracy, so the comparison is entirely about extrapolation.

5.1 The ordering

The table below reports the median L_{90} across five seeds and also shows every seed. L_{90} is the longest tested length where scaled accuracy is at least 0.9.

Configuration	median L_{90}	per-seed L_{90}
R6	96	$\{64, 128, 128, 96, 96\}$
A-front	**192**	$\{192, 96, 128, 192, 256\}$
A-mid	**192**	$\{256, 192, 256, 192, 48\}$
A-back	128	$\{64, 128, 128, 256, 64\}$
A2-mid	96	$\{64, 96, 192, 192, 64\}$
A6	0	$\{0, 0, 0, 0, 0\}$

Front and middle placement double median L_{90} from 96 to 192. Back placement reaches 128, and adding a second attention block reaches 96. Mean accuracy at $T = 256$ shows the same descriptive order, with substantial seed spread: R6 reaches 0.588 ($\pm$0.061), A-front 0.820 ($\pm$0.133), A-mid 0.839 ($\pm$0.197) and A-back 0.650 ($\pm$0.161). R6 is not at chance at $T = 256$; it degrades gradually, averaging 0.898 at $T = 128$ and 0.588 at $T = 256$. Since ratio is held at 1/6 across A-front, A-mid and A-back, and parameters are identical to the last weight, position in the stack is the only variable separating those three.

5.2 What five seeds support

We ran exact two-sided Mann–Whitney tests for all fifteen configuration pairs at $T = 256$. Five seeds lower the resolution floor to $p = 0.008$, but the estimates are still noisy and the family contains 15 comparisons. After Holm correction, no placement contrast reaches 0.05. The correct reading is therefore descriptive: front and middle placement have higher median L_{90} and higher mean accuracy at $T = 256$, but this experiment does not establish a unique winner. The front-versus-middle contrast is especially small (0.820 versus 0.839).

5.3 Ratio versus placement

Doubling the attention budget to 2/6 with one block at the back (A2-mid) reaches median $L_{90} = 96$ and mean accuracy 0.663 at $T = 256$, below both single-block front and middle placements but above back placement alone. We therefore state the ratio claim in its weak form: at fixed parameter count, spending a second block on attention did not recover what moving the first block to the front or middle achieved. We do not claim that ratio is irrelevant, and note that A2-mid confounds two changes at once, since it both doubles the ratio and adds a back-placed block, so a ratio-versus-placement decomposition would require a A2-front or A2-symmetric arm we did not run.

6 The Recall Axis

Table 1 and Fig. 2(b) give mqar, and the ordering is close to the reverse of the state-tracking one. A-mid attains 0.968 in-distribution accuracy and A2-mid 0.946; A-back averages 0.306 but is bimodal across its five seeds ($\{0.999, 0.210, 0.115, 0.105, 0.103\}$) and should not be read as a stable midpoint; A-front manages only 0.091, and the pure recurrent R6 0.008.

The consequence for design is stated in Table 2. A-front is among the best state-tracking configurations and is recall-pessimal among hybrids. A-back is the reverse, at lower quality on both and with high seed variance. A-mid, the configuration the hybrid design literature recommends on the basis of perplexity and retrieval alone NVIDIA (2025a), is the only configuration near the top of both axes. We think this is the most useful thing in the paper: the deployed recipe survives contact with a capability axis it was never tested against, and its two obvious perturbations do not. That is a result the design sweeps could not have produced, because their metric could not see it. Even with five seeds, the A-mid versus A2-mid contrast (0.968 against 0.946) should be treated as descriptive rather than as a resolved difference.

An honest caveat on the transformer baseline. Our A6 reaches only 0.141 on mqar, whereas Zoology reports that attention solves the task perfectly at model dimension 64 Arora et al. (2024a). Our setup differs from theirs in vocabulary (256 versus 8192), depth (6 versus 2 layers) and training budget, and we did not tune further. The comparisons in Table 2 are therefore internally valid, since all six configurations share one protocol, but our A6 column should not be read as the published attention ceiling.

7 Knockout: What Is the Attention Block Actually Doing?

After training we replace every A-block's attention output with a residual passthrough and re-measure in-distribution accuracy, leaving all other weights untouched. The result is a sharp dissociation, visible in the last column of Table 1 and tested in Table 3. Pooled over

the four hybrid configurations, knockout moves in-distribution parity accuracy from 1.000 to 0.990, and an exact paired sign-flip test over the 20 parity runs does not reject ($p = 0.2500$). It moves mqar accuracy from 0.578 to 0.008, which is chance, with $p = 1.9 \times 10^{-6}$ over the 20 mqar runs. The test enumerates every possible sign reversal of the nonzero within-run changes.

This is the one comparison in the paper that the sample size does resolve, and it does so because it is paired within run rather than contrasted across configurations. Attention is therefore load-bearing at inference for recall and disposable at inference for state tracking, in the same models. On mqar the attention block is the gather-and-aggregate circuit Bick et al. (2025); delete it and nothing remains. On parity, the trained recurrence carries the automaton by itself, and the attention block's entire contribution was made during optimisation. That is a training-dynamics effect, not an architectural-capacity effect, and it is the mechanism that makes the placement hypothesis of §5 plausible: what should matter is whether attention sits upstream of the recurrence during learning.

8 Limitations

Five seeds do not give a precise ranking. The parity L_{90} results vary substantially across seeds: A-mid has $\{256, 192, 256, 192, 48\}$, while A2-mid has $\{64, 96, 192, 192, 64\}$. On mqar, the five A-back accuracies are $\{0.999, 0.210, 0.115, 0.105, 0.103\}$. The smallest unadjusted parity p-value is 0.008. After Holm correction, however, none of the 15 pairwise comparisons is below 0.05. The observed ordering is therefore suggestive, not a confirmed ranking. More seeds are needed for narrower estimates.

Results vary across devices. The earlier CPU study and the A6000 study share 48 matched runs. The two devices have the same L_{90} in 34 cases. Their mean absolute differences are 0.0140 for in-distribution accuracy and 0.0616 for terminal accuracy. We analyze the devices separately because the two sets of results do not always agree. We therefore do not claim that the layout ranking is independent of hardware.

The experiments are small and controlled. The A_5 models have 226,112 parameters. The parity models have 222,272 parameters, and the mqar models have 254,784 because the task vocabularies differ. This scale is common in controlled studies of capability and expressivity Delétang et al. (2023); Merrill et al. (2024); Sarrof et al. (2024); Grazzi et al. (2025); Chung et al. (2026); Jiang (2026). Our conclusions apply only to the matched models, tasks, and training budget studied here. They do not predict how a 3-billion-parameter hybrid will behave.

9 Conclusion

Holding parameters, depth and training budget exactly fixed, we measured how attention placement inside a six-block linear-RNN hybrid affects two capabilities the hybrid design literature has never jointly evaluated. The clearest result is mechanistic and is resolved inferentially: the attention block performs the recall computation at inference ($0.578 \rightarrow 0.008$ under knockout, $p = 1.9 \times 10^{-6}$) but contributes to state tracking only during training, leaving the recurrence able to run the automaton without it ($1.000 \rightarrow 0.990$, $p = 0.2500$). Consistent with that mechanism, front and middle placement extrapolate furthest on parity while back placement buys nothing over the pure recurrent baseline, and the placement that is best for state tracking is the worst for recall, with middle placement, the deployed recipe, the only configuration strong on both. We report the placement ordering as a controlled observation rather than a significance result, and we state the seed uncertainty and multiple-testing limit explicitly. The primary study is 90 training runs on an NVIDIA RTX A6000, totaling 89 recorded minutes.

Reproducibility statement

Training examples for all three tasks are created during each run. The main analysis covers 90 runs, with five seeds for each task and layout. Training used a NVIDIA RTX A6000 with CUDA 12.4 and PyTorch 2.6.0+cu124. The run logs record 89 minutes of training in total. All tables, figures, tests, and numerical values are computed directly from the JSON records. The earlier CPU baseline is analyzed separately. Across 48 matched CPU and GPU runs, L_{90} is the same in 34 cases, and the mean absolute difference in terminal accuracy is 0.0616. We never combine observations from the two devices.

AI use statement

Generative AI tools were used only to improve the grammar, clarity, and organization of the manuscript. The author independently created and verified the data, code, analyses, figures, and scientific conclusions.

References

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In Proc. ICLR, 2024a. arXiv:2312.04927.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall–throughput tradeoff. In Proc. ICML, 2024b. arXiv:2402.18668.

Aviv Bick, Eric P. Xing, Albert Gu, et al. Understanding the skill gap in recurrent language models: The role of the gather-and-aggregate mechanism. In Proc. ICML, 2025. arXiv:2504.18574.

Jiwan Chung, Heechan Choi, and Seon Joo Kim. Rethinking state tracking in recurrent models through error control dynamics. arXiv preprint arXiv:2605.07755, 2026.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. Neural networks and the Chomsky hierarchy. In Proc. ICLR, 2023. arXiv:2207.02098.

Falcon-LLM Team. Falcon-H1: A family of hybrid-head language models redefining efficiency and performance. arXiv preprint arXiv:2507.22448, 2025.

Riccardo Grazzi, Julien Siems, Arber Zela, Jörg K. H. Franke, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear RNNs through negative eigenvalues. In Proc. ICLR, 2025. arXiv:2411.12537.

Yuhang Jiang. Task structure reverses layerwise state encoding in sequence models. arXiv preprint arXiv:2606.00926, 2026.

Aakash Lahoti, Kevin Y. Li, Berlin Chen, Caitlin Wang, Aviv Bick, J. Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In Proc. ICLR, 2026.

Opher Lieber, Barak Lenz, Hofit Bata, et al. Jamba: A hybrid Transformer–Mamba language model. In Proc. ICLR, 2025. arXiv:2403.19887.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In Proc. ICLR, 2023. arXiv:2210.10749.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In Proc. ICLR, 2019.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In Proc. ICML, 2024. arXiv:2404.08819.

NVIDIA. Hybrid architectures for language models: Systematic analysis and design insights. arXiv preprint arXiv:2510.04800, 2025a.

NVIDIA. Nemotron-H: A family of accurate and efficient hybrid Mamba–Transformer models. arXiv preprint arXiv:2504.03624, 2025b.

Buu Phan, Reza Ebrahimi, Sanjay Haresh, and Roland Memisevic. Delayed attention training improves length generalization in Transformer–RNN hybrids. arXiv preprint arXiv:2510.00258, 2025.

Shivam Rawat, Lucie Flek, Florian Mai, and Nicholas Kluge Corrêa. Reasoning primitives in hybrid and non-hybrid LLMs: Do architectural differences yield advantages in state-tracking and recall? arXiv preprint arXiv:2604.21454, 2026.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In Proc. ICLR, 2025. arXiv:2406.07522.

Yash Sarrof, Yana Veitsman, and Michael Hahn. The expressive capacity of state space models: A formal language perspective. In Proc. NeurIPS, 2024. arXiv:2405.17394.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. DeltaProduct: Improving state-tracking in linear RNNs via Householder products. arXiv preprint arXiv:2502.10297, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Proc. NeurIPS, 2017.

Dustin Wang, Rui-Jie Zhu, Steven Abreu, et al. A systematic analysis of hybrid linear attention. arXiv preprint arXiv:2507.06457, 2025.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In Proc. ICLR, 2025. arXiv:2412.06464.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In Proc. NeurIPS, 2019.