LCRA: LOCAL ATTENTION WITH HIERARCHICALLY ROUTED COARSE-RESIDUAL MEMORY FOR LONG-CONTEXT RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Recurrent language models compress distant history into fixed-size states, but such compression can make individually retrievable associations difficult to preserve as new information accumulates. Existing gated or delta-rule memories primarily focus on how a shared recurrent state is updated, rather than explicitly partitioning memory across associations. Routing-based memories provide an organizational mechanism, but coarse aggregation within a route can still merge distinct values. We introduce **LCRA**, a Local-Coarse-Residual Architecture that combines fixed-window local FoX attention for recent context with hierarchically routed coarse and residual recurrent memory for distant context. Route-conditioned value-and-mass statistics provide coarse predictions, while grouped fast-weight states use residual targets relative to pre-write coarse estimates. The resulting remote readout is fused with the local branch, while the size of the remote recurrent state remains independent of context length. Across seven approximately 720M-parameter checkpoints, LCRA remains competitive on language modeling and general zero-shot tasks while showing its clearest gains in long-context retrieval. It leads all four evaluated 8K RULER-style synthetic retrieval tasks and retains **76.95%** accuracy on 16K S-NIAH-1, compared with **19.53%** for GDN-2. LCRA also achieves the lowest fixed-target PG-19 loss through 16K and the highest real-world in-context retrieval mean among the evaluated fixed-state models, while trailing the KV-access references on that mean. A separate 3B-token ablation study supports the complete remote-memory configuration and reveals task- and length-dependent trade-offs in the residual write target. Together, these results position LCRA as a promising fixed-state architecture for long-context retrieval.

1 INTRODUCTION

Long-context retrieval asks a language model to recover information from distant text while retaining precise access to nearby tokens. Self-attention provides direct content-based access, but dense training attention is quadratic in context length and its key–value cache grows with the sequence (Vaswani et al., 2017). Recurrent state-space and linear-attention models instead summarize history in fixed-size state (Katharopoulos et al., 2020; Gu & Dao, 2024). With bounded memory, the challenge is to keep useful associations addressable as new information arrives without storing every preceding token.

The fast-weight view of linear attention treats recurrent state as an online key–value map; delta-rule updates correct the value predicted by that map (Schlag et al., 2021). Gated DeltaNet adds learned forgetting, and Gated DeltaNet-2 (GDN-2) separates retention, erase, and write controls (Yang et al., 2025; Hatamizadeh et al., 2026). Such controls govern how a shared state is edited but do not themselves partition it among associations. Learned codebooks offer a complementary way to organize keys (van den Oord et al., 2017; Lingle, 2024), although a single mean for each route can merge distinct values. This motivates combining soft routing with corrective memory inside each routed group.

We introduce LCRA, a Local-Coarse-Residual Architecture that combines fixed-window local FoX attention (Lin et al., 2025) for recent context with hierarchically routed coarse and residual recurrent memory for distant context. A hierarchical codebook softly routes remote keys and queries through

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.

parent and child prototypes. Coarse value-and-mass states provide a route-weighted estimate, while grouped residual fast-weight states correct deviations from the pre-write coarse estimate. A token enters remote memory as it leaves the local window; the local and remote readouts are then normalized separately and combined by a learned, position-independent channelwise fusion gate. LCRA uses no explicit positional embeddings and requires no positional rescaling when evaluated beyond its training context. Figure 1 summarizes this division of labor.

We distinguish models that retain token-level KV access to distant history from those that compress it into fixed-size state. GDN-2, Mamba-3, and Liger share LCRA's fixed-state constraint; Transformer++ (Gu & Dao, 2024), original NHA, and NAtS-L provide KV-access references. State capacity and computational budgets can still differ within each group.

Across seven approximately 720M-parameter checkpoints, LCRA has the lowest FineWeb-Edu test perplexity and an eleven-task mean of 54.88%, close to Transformer++ at 55.10%. Its clearest advantages occur in long-context evaluation: it leads all four tested 8K RULER-style synthetic in-context retrieval tasks (Hsieh et al., 2024) and reaches 76.95% on 16K S-NIAH-1, compared with 19.53% for GDN-2. LCRA also has the lowest loss on matched PG-19 targets through 16K (Rae et al., 2020). On real-world in-context retrieval, it leads the fixed-state group in mean score but trails the KV-access references.

Our contributions are threefold: a local–remote architecture combining hierarchical routing with coarse and residual memory; an evaluation across both long-range storage groups; and a separate six-variant, 3B-token ablation study of local attention, remote memory, and the residual write target. Together, these experiments characterize the design's gains and task-dependent trade-offs.

2 Related Work

Recurrent sequence models and fast weights. Linear attention admits a fixed-size recurrent state (Katharopoulos et al., 2020), and its fast-weight interpretation motivates delta-rule corrections to previously stored key–value mappings (Schlag et al., 2021). Gated DeltaNet adds learned forgetting; Gated DeltaNet-2 (GDN-2) further separates retention, key-side erasure, and value-side writing (Yang et al., 2025; Hatamizadeh et al., 2026). Selective state-space models pursue a different recurrence, with Mamba-3 improving state tracking and inference-oriented computation (Gu & Dao, 2024; Lahoti et al., 2026). These approaches edit or evolve a bounded state. LCRA retains the fast-weight update view but partitions its remote state by learned routes and models the residual of a routed coarse estimate.

Hybrid local and remote computation. BASED combines sliding-window attention with linear attention to trade recurrent-state size against recall (Arora et al., 2024a). Infini-attention places masked local attention and a compressive long-term memory within one Transformer block (Munkhdalai et al., 2024), while Hymba runs attention and state-space heads in parallel (Dong et al., 2025). Liger Attention combines softmax and gated recurrent attention within a layer (Lan et al., 2025). Native Hybrid Attention (NHA) combines recurrent slots and recent tokens in one softmax (Du et al., 2026); NAtS-L adaptively selects linear or softmax attention for tokens (Deng et al., 2026). FoX instead introduces a forget gate into softmax attention and a complementary Pro block (Lin et al., 2025). LCRA uses FoX only for a fixed recent window. Thus the local–remote split is shared with prior hybrid designs, whereas LCRA's distinguishing design choice is to organize remote memory using hierarchically routed coarse statistics together with grouped residual fast-weight states.

Discrete and routed memory. Vector quantization learns reusable prototypes for continuous representations (van den Oord et al., 2017). Product-key memory addresses a large learned parameter memory with structured keys (Lample et al., 2019); Transformer-VQ quantizes attention keys and accumulates sufficient statistics for linear-time softmax attention (Lingle, 2024). LCRA's parent–child codebook instead provides soft, data-dependent routes into recurrent value-and-mass states, with one residual fast-weight matrix per parent group. The coarse statistics and residual correction therefore serve different roles within the same routed memory rather than acting as alternative readouts of a single code average.

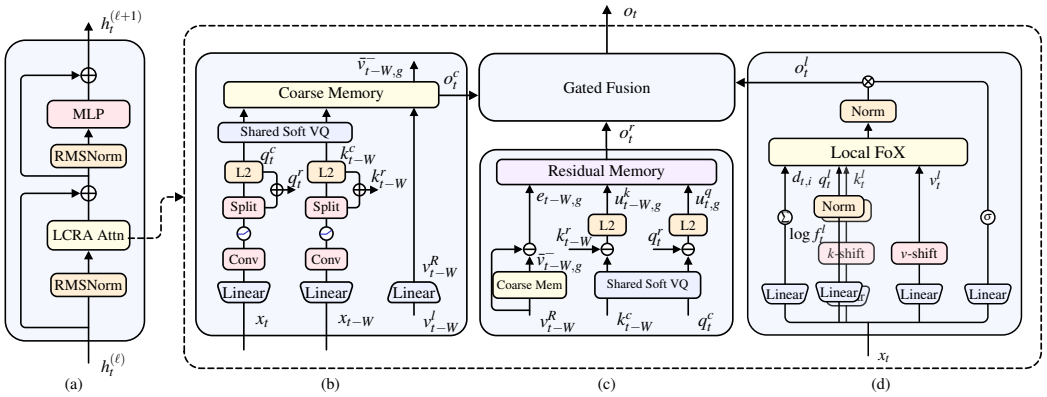

Figure 1: LCRA architecture. The local FoX branch reads a causal window of W tokens; the remote branch stores earlier tokens in hierarchically routed coarse and grouped residual memories. Their readouts enter gated fusion, whose exact combination is given in Eq. (10).

Long-context evaluation. RULER extends synthetic in-context retrieval with varied needle types and multi-key tasks (Hsieh et al., 2024); we evaluate four of its needle tasks rather than the full benchmark. PG-19 supplies long-form book text (Rae et al., 2020), which we use for matched-target fixed-window perplexity rather than continuous streaming. For real-world in-context retrieval, SQuAD-Completion, FDA, and SWDE follow BASED, while NQ, TriviaQA, and DROP use Just Read Twice document-QA adapters (Arora et al., 2024a;b). We report per-task literal-containment scores and their equal-weight mean.

3 METHOD

LCRA replaces the mixer in a pre-normalized decoder block (Figure 1). For a length-T sequence, $h_t^{(\ell)}$ is the representation entering layer ℓ at position t, and $x_t = \mathrm{RMSNorm}(h_t^{(\ell)})$. We describe one head, omitting batch and head indices. Local attention reads the latest $W = 64$ tokens; remote memory stores older history. We use $G = J = 8$ groups and children per group, remote key/value dimensions $d_k = 32, d_v = 64$, local head dimension $d_l = 128$, and 12 heads (eight in the 3B ablations). Throughout, σ is sigmoid, $\odot$ is elementwise multiplication, and diag forms a diagonal matrix; vector sigmoid and exponential act elementwise.

Neither branch uses explicit positional embeddings (NoPE). Sequence order enters through causal attention, token shifts, causal convolutions, and recurrent state updates.

3.1 HIERARCHICAL ROUTING AND COARSE MEMORY

For each head, LCRA learns G normalized parent prototypes $p_g \in \mathbb{R}^{d_k}$ and J normalized child prototypes $c_{g,j}$ per parent. Each child is obtained by adding a learned, centered tangent offset to its parent before normalization. Remote queries and keys pass through separate causal depthwise convolutions and are split into coarse and residual features. Their ℓ_2-normalized coarse features $q_t^c, k_t^c \in \mathbb{R}^{d_k}$ route independently through the shared codebook. With fixed positive parent and child routing temperatures τ_p, τ_c, for $z \in \{q, k\}$ define

$$\rho_{t,g}^z = \mathrm{softmax}_g(\langle z_t^c, p_g\rangle/\tau_p), \quad \pi_{t,g,j}^z = \mathrm{softmax}_j(\langle z_t^c, c_{g,j}\rangle/\tau_c), \quad w_{t,g,j}^z = \rho_{t,g}^z \pi_{t,g,j}^z. \tag{1}$$

Parent probabilities allocate mass across groups, and conditional child probabilities allocate it within each group. The remote value $v_t^R \in \mathbb{R}^{d_v}$ is a headwise linear adaptation of the shifted value shared with the local branch.

At position t, the remote writer processes $n = t - W$ when $n \geq 1$. This token has just left the local window $[t - W + 1, t]$. The coarse numerator $A_{t,g,j} \in \mathbb{R}^{d_v}$ and mass $Z_{t,g,j} \in \mathbb{R}$, both initially zero, receive the key-routed write

$$A_{t,g,j} = A_{t-1,g,j} + w_{n,g,j}^k v_n^R, \qquad Z_{t,g,j} = Z_{t-1,g,j} + w_{n,g,j}^k. \tag{2}$$

For $t \le W$, the states remain unchanged. After the write, the current query reads a weighted mean,

$$o_t^c = \mathrm{SafeDiv}\left(\sum_{g,j} w_{t,g,j}^q A_{t,g,j},\ \sum_{g,j} w_{t,g,j}^q Z_{t,g,j}\right), \tag{3}$$

where $\mathrm{SafeDiv}(a,b) = a/b$ for $b > \epsilon_m$ and zero otherwise, with positive mass threshold ϵ_m. The pre-write version of this state supplies the baseline for the residual branch.

3.2 Grouped residual memory

Even a child-level mean can merge distinct values routed to the same prototype. LCRA therefore maintains one fast-weight matrix $M_{t,g} \in \mathbb{R}^{d_v \times d_k}$ per parent group. Before writing token $n = t - W$, it computes the group baseline from the *pre-write* coarse state:

$$\bar{v}_{n,g}^- = \mathrm{SafeDiv}\left(\sum_j \pi_{n,g,j}^k A_{t-1,g,j},\ \sum_j \pi_{n,g,j}^k Z_{t-1,g,j}\right), \qquad e_{n,g} = v_n^R - \mathrm{sg}(\bar{v}_{n,g}^-). \tag{4}$$

Here, "residual" refers to the write target defined relative to the group-specific coarse estimate before the current token is written; it does not imply an exact decomposition of the later coarse and residual readouts. sg stops gradients through the baseline. With learned per-head scalar scales s_k, s_q and convolutional offset features $\Delta k_t, \Delta q_t \in \mathbb{R}^{d_k}$ (Appendix A.2), the residual features are $k_t^r = k_t^c + s_k \Delta k_t$ and $q_t^r = q_t^c + s_q \Delta q_t$. Relative to parent p_g, the smooth ℓ_2-normalized write and read addresses are

$$u_{n,g}^k = \mathrm{L2Norm}(k_n^r - p_g), \qquad u_{t,g}^q = \mathrm{L2Norm}(q_t^r - p_g). \tag{5}$$

Here $\mathrm{L2Norm}(a) = a/\sqrt{\|a\|_2^2 + \epsilon_2}$, where $\epsilon_2 > 0$ stabilizes normalization.

Following the retention, erase, and write decomposition of Gated DeltaNet-2 (Hatamizadeh et al., 2026), a learned controller produces vector log-retention $\ell_{n,g} \le 0$ and erase gate $\eta_{n,g} \in (0,1)^{d_k}$ from the relative key, and a write gate $\beta_{n,g} \in (0,1)^{d_v}$ from the relative key and writer value. The parent responsibility scales these controls:

$$F_{n,g} = \exp(\rho_{n,g}^k \ell_{n,g}), \quad E_{n,g} = \rho_{n,g}^k \eta_{n,g}, \quad B_{n,g} = \rho_{n,g}^k \beta_{n,g}. \tag{6}$$

With $D_{n,g} = M_{t-1,g}\,\mathrm{diag}(F_{n,g})$, the grouped state update is

$$M_{t,g} = D_{n,g} + \left[B_{n,g} \odot e_{n,g} - D_{n,g}\left(E_{n,g} \odot u_{n,g}^k\right)\right](u_{n,g}^k)^\top. \tag{7}$$

When no remote writer exists, $M_{t,g} = M_{t-1,g}$. The query uses its parent probabilities to aggregate the updated group predictions,

$$o_t^r = \sum_g \rho_{t,g}^q M_{t,g} u_{t,g}^q. \tag{8}$$

Per head, the remote state stores $GJ(d_v+1)$ coarse-state scalars and Gd_vd_k residual-state scalars, independent of sequence length.

3.3 Local FoX

The local branch adopts the forgetting-attention and Pro-block components of FoX (Lin et al., 2025): normalized queries and keys, shifted keys and values, a per-head forget gate, and an output gate. LCRA restricts that attention to the causal window $\mathcal{W}_t = \{\max(1, t-W+1), \dots, t\}$, leaving earlier history to remote memory. For $i \in \mathcal{W}_t$, define the forgetting bias $d_{t,i} = \sum_{m=i+1}^t \log f_m^l$; the normalized attention weight is proportional to

$$\alpha_{t,i} \ \propto\ \exp\left(\frac{\langle q_t^l, k_i^l\rangle}{\sqrt{d_l}} + d_{t,i}\right), \qquad o_t^l = g_t^l \odot \mathrm{RMSNorm}_0\left(\sum_{i\in\mathcal{W}_t} \alpha_{t,i} v_i^l\right). \tag{9}$$

Here $q_t^l, k_i^l, v_i^l \in \mathbb{R}^{d_l}$ are local queries, shifted keys, and shifted values. $\mathrm{RMSNorm}_0$ denotes non-affine RMS normalization with positive stabilizer ϵ_r, whereas the input RMSNorm has learned gains. $f_m^l \in (0,1)$ is the FoX forget gate and $g_t^l \in (0,1)^{d_l}$ is its token-dependent output gate. Local attention and remote memory share a shifted value source to couple their content representations, while retaining separate query and key projections. Appendix D compares this configuration with independent value projections.

3.4 GATED FUSION AND BLOCK OUTPUT

The coarse and residual outputs first form a remote readout. A headwise linear map lifts it to the local head dimension, and a non-affine RMS normalization aligns its scale with the already normalized local branch:

$$r_t = o_t^c + o_t^r, \qquad \widetilde{r}_t = \mathrm{RMSNorm}_0(W_{\mathrm{lift}} r_t), \qquad o_t = (1-\gamma) \odot o_t^l + \gamma \odot \widetilde{r}_t, \quad \gamma = \sigma(\theta). \tag{10}$$

Unlike the token-dependent FoX output gate g_t^l, $\gamma \in (0,1)^{d_l}$ is a position-independent gate with learned logits $\theta \in \mathbb{R}^{d_l}$ for each head. The lift matrix $W_{\mathrm{lift}} \in \mathbb{R}^{d_l \times d_v}$ maps remote values to the local head dimension. Concatenating all head outputs and applying the mixer output projection gives $m_t = W_o \,\mathrm{concat}_h(o_{t,h})$. The pre-norm decoder block then completes the two residual paths,

$$y_t = h_t^{(\ell)} + m_t, \qquad h_t^{(\ell+1)} = y_t + \mathrm{MLP}(\mathrm{RMSNorm}(y_t)) \,. \tag{11}$$

Algorithm 1 summarizes the recurrence; Appendix A.1 gives its mathematical form and feature and controller definitions.

Algorithm 1 LCRA forward flow for one head

Input: normalized representations $x_{1:T}$, window W, learned parameters.
Output: head outputs $o_{1:T}$, before concatenation and output projection.

```
Construct causal local and remote features; initialize memories A, Z, M to zero.
for each position t = 1, ..., T do
    Read the recent window with Local FoX (Eq. 9).
    n ← t − W; if n ≥ 1 then
        Route the key of the token leaving the window (Eq. 1).
        Predict its value in each group using the pre-write coarse memory.
        Form residual targets, stopping gradients through the predictions (Eq. 4).
        Update residual memory using writer keys, values, and targets (Eqs. 5–7).
        Add routed writer values and weights to coarse memory (Eq. 2).
    end if
    Route the current query; read both updated memories (Eqs. 3, 8).
    Sum, lift, and normalize the remote readout; fuse with the local output (Eq. 10).
end for; return o_{1:T}.
```

4 EXPERIMENTS

Setup. We compare LCRA with six sequence-modeling baselines: Transformer++ (Gu & Dao, 2024), Gated DeltaNet-2 (GDN-2) (Hatamizadeh et al., 2026), Mamba-3 (SISO) (Lahoti et al., 2026), Liger Attention (Lan et al., 2025), Native Hybrid Attention (NHA) (Du et al., 2026), and NAtS-L (Deng et al., 2026). The main models have approximately 720M parameters and are evaluated at their approximately 20B-token endpoints. The matched training runs use the same non-repeated FineWeb-Edu prefix (Penedo et al., 2024), 4K sequences, 131,072 tokens per update, and seed 42. We group models by whether distant history retains token-level KV access or is compressed into fixed-size recurrent state. NHA† is a converted, continued variant; Appendix B details its training protocol and the evaluation settings.

Evaluation. We evaluate perplexity on held-out FineWeb-Edu and WikiText-103 (Merity et al., 2016), eleven zero-shot likelihood tasks (Biderman et al., 2024), four RULER-style synthetic in-context retrieval tasks (Hsieh et al., 2024), and six real-world in-context retrieval tasks. PG-19 (Rae et al., 2020) measures loss on fixed targets as context length grows from 2K to 16K, beyond the 4K training length. For real-world in-context retrieval, generation is zero-shot and greedy. Appendix B specifies tasks, metrics, batch sizes, and model-specific evaluation paths.

Ablations. We separately train six LCRA variants, ranging from approximately 207M to 255M parameters, for 3B FineWeb-Edu tokens under a matched 4K training protocol. The study removes the remote branch or grouped residual memory, includes two remote-only variants without local attention, and replaces the residual write target with the full remote value. Parameter-matched controls widen the FFN where needed. We evaluate language modeling, general tasks, synthetic in-context retrieval, and real-world in-context retrieval. Table 6 summarizes the variants; Appendices A–B specify state ordering and the training and evaluation settings.

5 RESULTS

In the tables, bold and underlined values mark the best and second-best point estimates within each long-range storage group (or across variants in the ablation tables); ties share the corresponding rank marker, and these ranks do not imply statistical significance.

5.1 LANGUAGE MODELING AND GENERAL TASKS

As shown in Table 1, LCRA remains competitive on language modeling and general zero-shot evaluation. It achieves the lowest FineWeb-Edu perplexity at 12.29 and the highest eleven-task mean among fixed-state models at 54.88%, close to the overall best mean of 55.10% achieved by Transformer++. Transformer++ also performs best on WikiText-103.

Table 1: Language modeling and zero-shot performance at the approximately 20B-token training endpoint. Lower perplexity is better; task scores are percentages and higher is better. acc_n denotes normalized accuracy, and Avg. is the eleven-task mean. † denotes the converted NHA variant described in Appendix B.

Model	FW-Edu ppl↓	Wiki ppl↓	LAMB. acc↑	PIQA acc_n↑	Hella. acc_n↑	BLiMP acc↑	Wino. acc↑	ARC-E acc_n↑	SciQ acc_n↑	OBQA acc_n↑	BoolQ acc↑	SIQA acc↑	ARC-C acc_n↑	Avg. acc↑
Long-Range with KV Access														
Transformer++	**12.31**	**37.03**	**38.77**	**71.00**	**49.04**	82.63	**55.33**	58.38	**80.30**	**37.60**	**61.41**	39.61	32.00	**55.10**
NHA	13.92	43.29	33.69	67.41	43.22	**84.23**	52.09	55.51	75.90	34.00	60.73	39.00	29.27	52.28
NAtS-L	12.47	45.54	38.04	69.80	48.01	82.97	54.22	**59.30**	78.00	35.60	59.66	**39.92**	**32.08**	54.33
Long-Range with Fixed-State Memory														
GDN-2	12.56	44.92	**38.62**	70.08	48.36	**84.16**	54.30	59.51	**79.20**	35.00	56.45	40.02	32.51	54.38
Mamba-3	12.50	45.64	37.59	**70.24**	49.23	83.39	53.20	59.55	78.10	36.20	60.31	39.82	32.68	54.57
Liger	13.01	41.97	37.71	70.18	48.36	81.91	53.59	59.43	75.90	**37.20**	**62.05**	39.41	32.00	54.34
NHA†	16.96	68.72	19.31	66.49	38.86	81.26	51.54	50.13	64.70	33.20	55.38	37.87	29.10	47.98
LCRA (ours)	**12.29**	**41.25**	37.96	69.97	**49.33**	83.01	**54.54**	**60.14**	77.40	36.60	60.34	**40.89**	**33.53**	**54.88**

5.2 RULER-STYLE SYNTHETIC IN-CONTEXT RETRIEVAL

LCRA leads all four evaluated 8K synthetic retrieval tasks (Table 2), reaching 95.70% on S-NIAH-1 versus 67.19% for GDN-2 and 38.67% for Mamba-3. At 16K, LCRA retains 76.95% accuracy on S-NIAH-1, compared with 19.53% for GDN-2, 12.11% for Mamba-3, and 0.39% for Liger, although all four fixed-state models score at most 12.11% on each of the remaining three tasks. Among the KV-access references, original NHA, which retains full-attention layers and long-range KV access, scores zero on all four 8K tasks under its original RoPE setting; Transformer++ likewise scores zero on all four tasks at both 8K and 16K under its original RoPE setting. Together, these results show stronger end-to-end length extrapolation for LCRA under the evaluated configurations: it leads all four tasks at 8K and retains substantial S-NIAH-1 accuracy well beyond the 4K training context.

5.3 FIXED-WINDOW MODELING ON PG-19

Figure 2 plots mean target-token NLL and its change relative to 2K on the same 231 PG-19 targets. Appendix Tables 9 and 10 list perplexities and NLL changes.

Because each context length is evaluated on identical target tokens, changes in loss directly measure whether additional history improves prediction. LCRA has the lowest PG-19 loss at every evaluated length and continues to improve beyond its 4K training context, with target NLL decreasing by 0.05786 nats/token from 2K to 16K, the largest reduction among the evaluated models. These gains beyond the 4K training context are obtained without positional rescaling or additional long-context training. In contrast, Transformer++ and original NHA deteriorate sharply from 8K under their original RoPE settings, while NAtS-L, Liger, and NHA† show smaller degradation beyond 4K. GDN-2 and Mamba-3 also benefit from longer context, but remain above LCRA in absolute loss throughout the evaluated range; at 16K, LCRA reaches 32.09 PPL.

Table 2: RULER-style synthetic in-context retrieval accuracy at the approximately 20B-token training endpoint. Scores are percentages; higher is better. NHA uses its original RoPE setting. "–" indicates that the 16K evaluation was not completed due to the high evaluation cost under the current implementation. Additional evaluation details are provided in Appendix B.

Model	S-NIAH-1				S-NIAH-2				S-NIAH-3				MK-NIAH-1			
	2K	4K	8K	16K	2K	4K	8K	16K	2K	4K	8K	16K	2K	4K	8K	16K
Long-Range with KV Access																
Transformer++	99.22	91.02	0.00	**0.00**	91.41	66.80	0.00	**0.00**	**84.77**	33.20	0.00	**0.00**	**41.80**	19.14	0.00	**0.00**
NHA	**100.00**	**92.97**	0.00	–	**97.27**	**89.84**	0.00	–	79.30	**57.03**	0.00	–	34.77	**32.42**	0.00	–
NAtS-L	96.48	78.91	**38.67**	–	46.88	47.66	**5.86**	–	21.88	10.94	**1.56**	–	6.64	14.06	**0.39**	–
Long-Range with Fixed-State Memory																
GDN-2	**100.00**	**99.22**	67.19	19.53	53.91	39.84	6.25	8.20	24.22	20.31	3.91	3.91	14.84	16.80	2.73	2.34
Mamba-3	90.62	64.06	38.67	12.11	42.58	39.06	7.42	8.59	15.62	20.31	3.91	6.25	2.73	3.91	0.78	1.17
Liger	13.28	1.17	0.78	0.39	3.52	4.30	2.73	2.73	3.91	3.91	3.91	3.91	1.95	2.34	1.95	1.95
LCRA (ours)	**100.00**	**99.22**	**95.70**	**76.95**	**77.34**	**68.75**	**20.70**	**12.11**	**67.19**	**62.11**	**10.16**	**11.72**	**19.53**	**18.75**	**7.03**	**6.64**

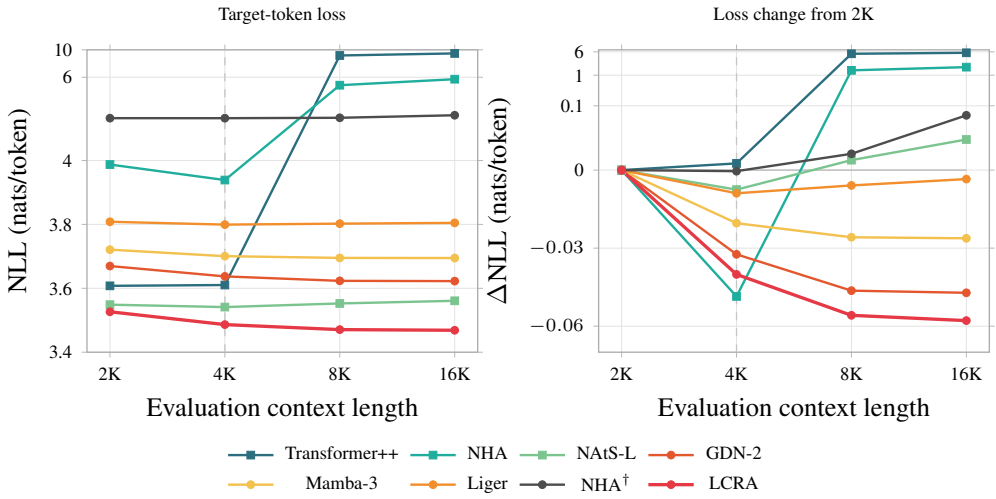

Figure 2: PG-19 target-token NLL and change from 2K on 231 fixed targets. Axes are linear below NLL 4 or ΔNLL 0.01 and logarithmically compressed above; all measured points are shown. The dashed line marks the 4K training length. NHA uses original RoPE; † denotes the converted variant. Full values appear in Appendix C.

5.4 Real-world in-context retrieval

LCRA leads the fixed-state group on four of the six real-world retrieval tasks and achieves the highest six-task mean within that group (32.66%), compared with 28.77% for GDN-2, 25.76% for Mamba-3, and 21.84% for Liger (Table 3). The largest gains over GDN-2 occur on FDA and SWDE, where LCRA improves by 8.71 and 11.07 percentage points, respectively. On NQ, LCRA reaches 20.49%, close to Transformer++ at 21.60%. These results show that LCRA's retrieval gains extend beyond synthetic benchmarks to several real-world tasks. The KV-access references retain higher overall means, ranging from 36.93% to 44.24%.

5.5 Architectural ablations at 3B tokens

Tables 4 and 5 report the separate 3B-token ablation study; Appendix A gives the variant configurations (Table 6). At the 4K training length, the full LCRA variant reaches 62.89% on S-NIAH-1. Removing remote memory raises FineWeb perplexity from 19.76 to 22.14 and reduces the synthetic retrieval score to 1.56%. Removing residual memory while keeping the coarse state yields 21.88%

Table 3: Zero-shot performance on real-world in-context retrieval tasks at the approximately 20B-token training endpoint. Scores are literal answer-containment percentages; higher is better. Avg. is the equal-weight mean of the six unrounded task scores and is separate from the eleven-task mean in Table 1.

Model	SQuAD-Completion	FDA	SWDE	NQ	TriviaQA	DROP	Avg.
Long-Range with KV Access							
Transformer++	**38.91**	**64.52**	**69.49**	**21.60**	**54.09**	16.82	**44.24**
NHA	35.12	57.17	53.11	20.34	51.07	17.35	39.02
NAtS-L	32.74	39.84	57.97	20.27	53.14	**17.63**	36.93
Long-Range with Fixed-State Memory							
GDN-2	31.30	17.33	35.46	17.55	**53.79**	17.20	28.77
Mamba-3	29.02	12.98	25.20	16.25	52.49	**18.64**	25.76
Liger	25.40	12.89	14.67	12.51	48.93	16.63	21.84
LCRA (ours)	**33.65**	**26.04**	**46.53**	**20.49**	51.78	17.44	**32.66**

and raises FineWeb perplexity to 20.52. These gaps support the complete remote-memory configuration for 4K synthetic in-context retrieval. These are component-removal ablations: removing a memory branch also removes its associated recurrent state, while FFN-width adjustments keep trainable parameter counts approximately matched where applicable.

Remote-only variants write each token immediately to memory and omit local–remote fusion; retaining delayed writes would leave recent positions outside both access paths. This is an architecture-level comparison, not an isolated removal of local attention (Appendix A).

Table 4: Language modeling and zero-shot performance in the separate 3B-token ablation study. FW/Wiki are perplexities; task scores are percentages, and acc_n denotes normalized accuracy. Avg. is the unweighted mean of the eight zero-shot tasks.

Variant	FW ppl↓	Wiki ppl↓	LAMB. acc↑	PIQA acc_n↑	Hella. acc_n↑	BLiMP acc↑	Wino. acc↑	ARC-E acc_n↑	SciQ acc_n↑	OBQA acc_n↑	Avg. acc↑
LCRA (full)	19.76	64.49	23.15	63.66	33.62	82.49	49.64	48.44	**69.30**	30.20	50.06
w/o remote	22.14	73.52	22.82	62.24	**33.83**	**82.57**	50.83	**49.83**	67.70	**31.60**	50.18
w/o residual memory	20.52	**58.65**	**23.69**	**64.20**	33.70	82.04	**52.72**	48.36	66.30	31.20	**50.28**
remote-only (same FFN width)	21.58	71.48	20.53	61.59	31.52	80.32	49.96	46.04	64.70	30.00	48.08
remote-only (matched params)	21.82	72.49	20.14	61.97	31.66	79.56	51.78	45.88	63.80	**31.60**	48.30
full-value target	**19.73**	66.54	23.29	63.93	33.57	81.96	52.49	47.73	67.90	30.20	50.13

Table 5: In-context retrieval performance in the separate 3B-token ablation study. Scores are percentages; higher is better.

Variant	S-NIAH-1			Real-world in-context retrieval		
	2K	4K	8K	SQuAD	FDA	SWDE
LCRA (full)	**92.58**	**62.89**	26.17	20.78	**11.25**	**17.37**
w/o remote	1.56	1.56	1.56	11.80	1.63	5.76
w/o residual memory	34.38	21.88	1.56	15.32	5.26	7.02
remote-only (same FFN width)	50.78	14.84	4.30	17.26	4.81	11.61
remote-only (matched params)	23.05	3.13	1.17	17.23	2.99	9.81
full-value target	71.09	50.39	**30.47**	**22.35**	9.44	15.57

The w/o residual memory variant improves WikiText perplexity (58.65 versus 64.49) but weakens synthetic and real-world retrieval. Both remote-only variants have worse FineWeb perplexity and 4K S-NIAH-1 accuracy than the full model. The full-value target improves FineWeb perplexity (19.73 versus 19.76) and 8K S-NIAH-1 (30.47% versus 26.17%), but worsens WikiText perplexity and lowers 4K S-NIAH-1 and two of three real-world scores. These trade-offs depend on task and length: full LCRA ranks first on four of the six retrieval metrics and second on two, while remaining close to the best general-task mean.

Computational efficiency. On one RTX PRO 6000 96GB, LCRA's training throughput declines from 16.53K to 15.75K input tokens/s (4.7%) between 2K and 16K under a fixed-token update protocol, and remains below all six baselines (Figure 3). With CUDA Graph, its 128-token full-request latency is 1.39–1.85 s, the lowest among valid runs at each length, including graph capture. These are measurements of the evaluated implementations and backends; Appendix E gives the protocols, exact speed values, and peak allocated memory.

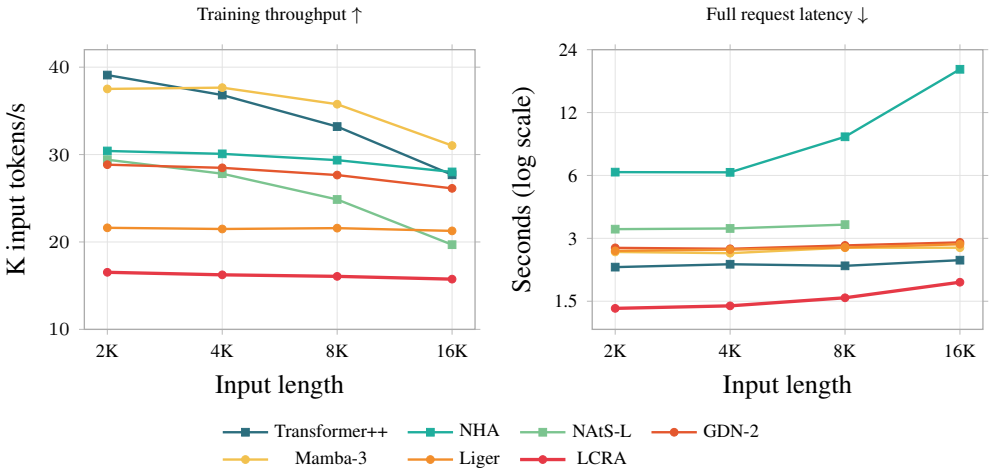

Figure 3: Single-GPU training throughput and request latency. Training uses batches 8/4/2/1 at 2K/4K/8K/16K with GA=1; inference uses B=1 and 128 generated tokens. LCRA uses CUDA Graph (capture included); NHA recomputes full prefixes. NAtS-L 16K inference fails with nonfinite logits.

6 Discussion and Limitations

LCRA's long-context gains are strongest on associative retrieval. At 16K, it retains 76.95% accuracy on S-NIAH-1 and also achieves the highest scores among the evaluated fixed-state models on S-NIAH-2/3 and MK-NIAH-1, although absolute accuracy on these harder tasks remains low across models. On real-world retrieval, LCRA leads the evaluated fixed-state models in mean score but remains below the KV-access references. These results indicate that challenging long-range retrieval remains difficult across architectures, particularly as retrieval complexity increases.

The storage groups are intended to distinguish long-range access regimes rather than define resource-matched comparisons; state size, local-window size, and compute can therefore differ across models. The remote-only comparison is architecture-level: removing the local path also requires immediate remote writes and eliminates local–remote fusion, so it does not isolate the marginal effect of local attention. No experiment separately isolates hierarchical routing. Likewise, NHA† measures a conversion and continuation procedure rather than a fully retrained fixed-state baseline.

The main 20B training runs and 3B ablations each use one training seed; the earlier 200M-token value-configuration study in Appendix D uses two. Model-specific long-context evaluation settings are detailed in Appendix B.

7 Conclusion

LCRA combines fixed-window local attention with hierarchically routed coarse and grouped residual memory of fixed size. It remains competitive on general tasks, leads all four evaluated 8K synthetic retrieval tasks, and achieves the lowest fixed-target PG-19 loss through 16K, improving beyond its 4K training context. It also has the highest real-world retrieval mean among evaluated fixed-state models. The 3B-token ablations support the complete remote-memory configuration while revealing task- and length-dependent trade-offs in the residual write target.

Reproducibility Statement

Sections 3–5 and Appendices A–E specify the architecture, training setups, training endpoints, evaluation protocols, and value-configuration comparison. Anonymous supplementary material provides the source code for the LCRA implementation and the training and evaluation pipelines used in this work.

Statement on the Use of AI

We used generative AI tools to assist with experimental-design feedback, code development and debugging, analysis of experimental results, literature review, and manuscript drafting and editing. All AI-assisted outputs were reviewed and verified by the authors. All reported experimental results were obtained from the described training and evaluation pipelines. The authors take full responsibility for the final content and claims of this work.

References

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall–throughput tradeoff. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1763–1840, 2024a. URL https://proceedings.mlr.press/v235/arora24a.html.

Simran Arora, Aman Timalsina, Aaryan Singhal, Benjamin Spector, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré. Just read twice: Closing the recall gap for recurrent language models. *arXiv preprint arXiv:2407.05483*, 2024b. doi: 10.48550/arXiv.2407.05483. URL https://arxiv.org/abs/2407.05483.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024. URL https://arxiv.org/abs/2405.14782.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2924–2936, 2019. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Difan Deng, Andreas Bentzen Winje, Lukas Fehring, and Marius Lindauer. Neural attention search linear: Towards adaptive token-level hybrid attention models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.03681.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. Hymba: A hybrid-head architecture for small language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=A1ztozypga.

Jusen Du, Jiaxi Hu, Zhang Tao, Weigao Sun, and Yu Cheng. Native hybrid attention for efficient sequence modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 3826–3842, 2026. doi: 10.18653/v1/2026.acl-long.176. URL https://aclanthology.org/2026.acl-long.176/.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs.

In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2368–2378, 2019. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246/.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2312.00752.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026. URL https://arxiv.org/abs/2605.22791.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2006.16236.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

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. *arXiv preprint arXiv:2603.15569*, 2026. URL https://arxiv.org/abs/2603.15569.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1907.05242.

Disen Lan, Weigao Sun, Jiaxi Hu, Jusen Du, and Yu Cheng. Liger: Linearizing large language models to gated recurrent structures. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 32452–32466, 2025. URL https://proceedings.mlr.press/v267/lan25b.html.

Zhixuan Lin, Evgenii Nikishin, Xu Owen He, and Aaron Courville. Forgetting transformer: Softmax attention with a forget gate. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.02130.

Lucas D. Lingle. Transformer-VQ: Linear-time transformers via vector quantization. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16354.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. URL https://arxiv.org/abs/1609.07843.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient infinite context transformers with Infini-attention. *arXiv preprint arXiv:2404.07143*, 2024. doi: 10.48550/arXiv.2404.07143. URL https://arxiv.org/abs/2404.07143.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024. URL https://arxiv.org/abs/2406.17557.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020. URL https://iclr.cc/virtual_2020/poster_SylKikSYDH.html.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4463–4473, 2019. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2102.11174.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1711.00937.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.03762.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.06464.

A LCRA IMPLEMENTATION DETAILS AND CONFIGURATION

A.1 RECURRENT FORWARD SPECIFICATION

Algorithm 2 expands Algorithm 1 for one head of one layer. State variables A, Z, M omit the time index and are updated in place; F, E, B, D abbreviate the corresponding writer/group quantities in Eqs. 6–7. At the start of iteration t, remote memory contains writes from positions $1, \dots, \max(t - W - 1, 0)$. The residual target uses this old coarse state; the current query reads the updated states, including position $t-W$ when it exists. For $t \le W$, both remote readouts are zero. The loop specifies causal dependencies; training can evaluate them using coarse scans and chunked residual operators. Streaming continuation also carries the bounded local and pending-writer buffers, token-shift histories, and convolution caches, in addition to (A, Z, M).

Algorithm 2 LCRA single-head recurrence with delayed remote writes

Input: $x_{1:T}$, window W, prototypes $p_g, c_{g,j}$, and learned parameters.
Output: $o_{1:T}$ and terminal remote states (A, Z, M).

Construct local and remote features as defined below.
$A \leftarrow 0_{G\times J\times d_v}$; $Z \leftarrow 0_{G\times J}$; $M \leftarrow 0_{G\times d_v\times d_k}$; $\gamma \leftarrow \sigma(\theta)$.
for $t = 1, \dots, T$ **do**
  $o_t^l \leftarrow \mathrm{LocalFoX}(t)$ using Eq. 9; $n \leftarrow t - W$.
  if $n \ge 1$ **then**
    $(\rho_n^k, \pi_n^k, w_n^k) \leftarrow \mathrm{Route}(k_n^c)$ using Eq. 1.
    for $g = 1, \dots, G$ **do**
      $\bar{v}_{n,g}^- \leftarrow \mathrm{SafeDiv}(\sum_j \pi_{n,g,j}^k A_{g,j}, \sum_j \pi_{n,g,j}^k Z_{g,j})$.
      $e_{n,g} \leftarrow v_n^R - \mathrm{sg}(\bar{v}_{n,g}^-)$. *Pre-write target*
      $d_{n,g} \leftarrow k_n^r - p_g$; $u_{n,g}^k \leftarrow \mathrm{L2Norm}(d_{n,g})$.
      $(\ell_{n,g}, \eta_{n,g}, \beta_{n,g}) \leftarrow \mathrm{Controller}(d_{n,g}, v_n^R; g)$.
      $F \leftarrow \exp(\rho_{n,g}^k \ell_{n,g})$; $E \leftarrow \rho_{n,g}^k \eta_{n,g}$; $B \leftarrow \rho_{n,g}^k \beta_{n,g}$.
      $D \leftarrow M_g \,\mathrm{diag}(F)$.
      $M_g \leftarrow D + [B \odot e_{n,g} - D(E \odot u_{n,g}^k)](u_{n,g}^k)^\top$.
    end for
    $A_{g,j} \leftarrow A_{g,j} + w_{n,g,j}^k v_n^R$; $Z_{g,j} \leftarrow Z_{g,j} + w_{n,g,j}^k$ for all g, j.
  end if
  $(\rho_t^q, \pi_t^q, w_t^q) \leftarrow \mathrm{Route}(q_t^c)$ using Eq. 1.
  $o_t^c \leftarrow \mathrm{SafeDiv}(\sum_{g,j} w_{t,g,j}^q A_{g,j}, \sum_{g,j} w_{t,g,j}^q Z_{g,j})$.
  $u_{t,g}^q \leftarrow \mathrm{L2Norm}(q_t^r - p_g)$ for all g; $o_t^r \leftarrow \sum_g \rho_{t,g}^q M_g u_{t,g}^q$.
  $\widetilde{r}_t \leftarrow \mathrm{RMSNorm}_0(W_{\mathrm{lift}}(o_t^c + o_t^r))$.
  $o_t \leftarrow (1 - \gamma) \odot o_t^l + \gamma \odot \widetilde{r}_t$.
end for; **return** $o_{1:T}, (A, Z, M)$.

A.2 FEATURE AND PROTOTYPE CONSTRUCTION

For one head, let $\widehat{q}_t^l = W_q^l x_t$, $\widehat{k}_t^l = W_k^l x_t$, and $\widehat{v}_t^l = W_v^l x_t$. Scalar shift gates $s_t^k = \sigma(a_k^\top x_t)$ and $s_t^v = \sigma(a_v^\top x_t)$ mix the current and previous features, with zero initial histories:

$$q_t^l = \mathrm{RMSNorm}_q(\widehat{q}_t^l), \qquad k_t^l = \mathrm{RMSNorm}_k(s_t^k \widehat{k}_{t-1}^l + (1 - s_t^k)\widehat{k}_t^l), \tag{12}$$

$$v_t^l = s_t^v \widehat{v}_{t-1}^l + (1 - s_t^v)\widehat{v}_t^l, \qquad v_t^R = W_{\mathrm{adapt}} v_t^l, \tag{13}$$

$$f_t^l = \sigma(a_f^\top x_t + b_f), \qquad g_t^l = \sigma(W_g^l x_t). \tag{14}$$

Let d_{model} denote the layer hidden width. All projections and gate parameters above are learned per head: $W_q^l, W_k^l, W_v^l, W_g^l \in \mathbb{R}^{d_l \times d_{\mathrm{model}}}$, $a_k, a_v, a_f \in \mathbb{R}^{d_{\mathrm{model}}}$, $b_f \in \mathbb{R}$, and $W_{\mathrm{adapt}} \in \mathbb{R}^{d_v \times d_l}$. The local query/key normalizations have learned gains. For $z \in \{q, k\}$, remote features are

$$[\widehat{z}_t^c, \Delta z_t] = \mathrm{SiLU}(\mathrm{CausalDWConv}_z(W_z^R x_{1:T})_t), \qquad z_t^c = \mathrm{L2Norm}(\widehat{z}_t^c), \quad z_t^r = z_t^c + s_z \Delta z_t. \tag{15}$$

Here $W_z^R \in \mathbb{R}^{2d_k \times d_{\mathrm{model}}}$ is learned, s_z is a learned scalar per head, and the brackets denote splitting into two d_k-dimensional features. $\mathrm{SiLU}(a) = a \odot \sigma(a)$; CausalDWConv is a causal depthwise convolution along token positions. Remote queries and keys have separate projections

and convolutions. The remote value adapts the already shifted local value, without a second shift or value convolution. Local FoX uses Eq. 9; its output already includes the local output gate and is not normalized again during fusion.

For learned raw parent vectors $\widehat{p}_g \in \mathbb{R}^{d_k}$ and child offsets $\delta_{g,j} \in \mathbb{R}^{d_k}$, define

$$p_g = \mathrm{L2Norm}(\widehat{p}_g), \qquad \bar{\delta}_{g,j} = \delta_{g,j} - J^{-1} \sum_i \delta_{g,i}, \tag{16}$$

$$\delta^{\perp}_{g,j} = \bar{\delta}_{g,j} - p_g \frac{p_g^\top \bar{\delta}_{g,j}}{\max(p_g^\top p_g, \epsilon_2)}, \qquad c_{g,j} = \mathrm{L2Norm}(p_g + \delta^{\perp}_{g,j}). \tag{17}$$

Equation 1 uses this shared codebook for query and key routing, with dense parent and conditional child softmax probabilities.

A.3 Residual controller and numerical conventions

The controller takes the unnormalized relative key $d = k_n^r - p_g$ and writer value $v = v_n^R$:

$$\ell_{n,g} = -\exp(\chi_g)\,\mathrm{softplus}(W_f d + b_{f,g}), \qquad \eta_{n,g} = \sigma(W_e d), \qquad \beta_{n,g} = \sigma(W_{\beta v} v + W_{\beta d} d). \tag{18}$$

Here $\ell, \eta \in \mathbb{R}^{d_k}$ and $\beta \in \mathbb{R}^{d_v}$. The learned matrices $W_f, W_e \in \mathbb{R}^{d_k \times d_k}$, $W_{\beta v} \in \mathbb{R}^{d_v \times d_v}$, and $W_{\beta d} \in \mathbb{R}^{d_v \times d_k}$ are shared across groups within a head; χ_g is a group-specific scalar and $b_{f,g}$ a group-specific vector. Erase and write projections have no additive biases. The write gate depends on the writer value, while the residual target supplies the value to the memory update.

The numerical operators use the positive configuration tolerances ϵ_2, ϵ_r, and ϵ_m for L2 normalization, RMS normalization, and the mass threshold, respectively; mean averages over vector components:

$$\mathrm{L2Norm}(a) = \frac{a}{\sqrt{\|a\|_2^2 + \epsilon_2}}, \qquad \mathrm{RMSNorm}_0(a) = \frac{a}{\sqrt{\mathrm{mean}(a \odot a) + \epsilon_r}}. \tag{19}$$

$\mathrm{SafeDiv}(a, b)$ returns a/b when $b > \epsilon_m$ and zero otherwise. $\mathrm{RMSNorm}_0$ is non-affine and sg stops gradients through its argument. After all heads are evaluated, Eq. 11 completes the decoder block.

A.4 Configurations and ablation state ordering

Model dimensions. Transformer++ has 23 layers, hidden width 1536, FFN width 4016, 16 attention heads of dimension 96, and 719,953,920 parameters. It uses full causal attention and RoPE with base 10000. The 20B LCRA model has 19 layers, hidden width 1536, 12 heads, FFN width 4184, and 719,976,396 parameters. The GDN-2 and Mamba-3 models each have 19 layers and width 1536; their FFN widths are 4096 and 4176, with 720,180,836 and 720,357,536 parameters, respectively. Liger has 22 layers, width 1536, FFN width 4296, and 720,393,216 parameters. The 3B LCRA variants have 15 layers, width 1024, and eight heads (Table 6). In the main 20B comparison and 3B ablations, $W = 64$, $G = J = 8$, $d_k = 32$, $d_v = 64$, and $d_l = 128$; the other baseline dimensions follow their recorded evaluation configurations.

The converted NHA† has 24 NHA layers, 721,394,688 parameters, 64 memory slots, and 32 visible local tokens. Its eleven-task evaluation uses batch size 64 except for SciQ and BoolQ at batch size 32. Its perplexities use all 12,207 FineWeb-Edu test windows and 139 WikiText-103 test windows, with batch size one, BF16 forward computation, and FP32 cross-entropy. The evaluated inputs and scoring windows match those of the original NHA evaluation. Its PG-19 evaluation scores the same 231 targets from 77 books at 2K/4K/8K/16K, using batches 64/64/32/16, FP32 weights and cross-entropy, BF16 forward computation, and no padding or generation cache. This endpoint has no complete results for either synthetic or real-world in-context retrieval.

Coarse value/mass and grouped residual states require $GJ(d_v + 1) + Gd_v d_k = 20{,}544$ recurrent scalars per head: 246,528 per 20B layer or 164,352 per 3B layer. These counts exclude the local buffer, convolution caches, parameters, and temporary activations; they are not measured peak memory.

Ablation variants. The six 3B variants use 15 layers and width 1024 (Table 6). The full model has 254.66M parameters. The w/o remote variant retains only local attention, while w/o residual memory

retains local attention and coarse remote memory. Both widen their FFNs to keep parameter counts near the full model. The two remote-only variants share the same memory architecture and differ only in FFN width. The full-value target variant keeps the complete architecture but writes the full remote value instead of its deviation from the pre-write coarse estimate.

Table 6: Model configurations for the separate 3B-token ablation study. Grouped denotes grouped residual remote memory; the remote-only variants use the state ordering described in Appendix A.

Variant	Local	Coarse	Grouped	FFN width	Parameters
LCRA (full)	✓	✓	✓	2280	254,662,952
w/o remote	✓	×	×	2696	254,769,272
w/o residual memory	✓	✓	×	2472	254,600,312
remote-only (same FFN width)	×	✓	✓	2280	207,185,072
remote-only (matched params)	×	✓	✓	3312	254,739,632
full-value target	✓	✓	✓	2280	254,662,952

Hybrid and remote-only state ordering. Let δ denote the delay from a token position to its remote-memory write. At step t, the writer processes $n_t = t - \delta$ whenever $n_t \geq 1$, with

$$\delta = \begin{cases} W, & \text{hybrid LCRA}, \\ 0, & \text{remote-only}. \end{cases}$$

Let $\mathcal{L}_t$ denote positions directly accessible through local attention and $\mathcal{R}_t$ positions already incorporated into the remote state at step t. For hybrid LCRA,

$$(\mathcal{L}_t, \mathcal{R}_t) = \begin{cases} (\{1, \ldots, t\}, \varnothing), & t \leq W, \\ (\{t - W + 1, \ldots, t\}, \{1, \ldots, t - W\}), & t > W. \end{cases}$$

Thus $\mathcal{L}_t \cap \mathcal{R}_t = \varnothing$ and $\mathcal{L}_t \cup \mathcal{R}_t = \{1, \ldots, t\}$. These sets describe the assignment of position-indexed representations; remote memory still compresses its assigned history into fixed-size state. Removing Local while retaining $\delta = W$ would leave $\{t - W + 1, \ldots, t\}$ outside both the local-access and remote-memory paths. The remote-only variants instead use $\delta = 0$, giving

$$\mathcal{L}_t = \varnothing, \qquad \mathcal{R}_t = \{1, \ldots, t\}.$$

Equations (2)–(8) apply with $n = n_t$: the residual target is formed from the pre-write state, the writer is committed, and the current query reads the updated coarse and residual states. Since remote-only has no local readout, Eq. (10) reduces before the shared mixer output projection to

$$o_t = \widetilde{r}_t.$$

B Training and evaluation protocols

Converted NHA variant. NHA† replaces the original NHA's full-attention layers with NHA layers, inherits its 20B-token checkpoint, undergoes 50M-token adaptation and 2B-token continuation, and is evaluated at continuation step 15641. It is therefore not a from-scratch baseline under the main training budget. Original NHA retains full-attention layers and long-range KV access, whereas the converted variant uses fixed-state long-range storage. We therefore report NHA† as an additional converted reference rather than as a directly matched main baseline. Its state capacity, local-window size, and compute are not matched to the main fixed-state baselines.

Training settings. The 20B training runs use the same unique FineWeb-Edu prefix, GPT-2 tokenizer, seed 42, and BF16 computation. Each update processes $4096 \times 4 \times 2 \times 4 = 131{,}072$ tokens (sequence length, per-GPU micro-batch, accumulation, GPUs). Transformer++ switched from micro-batch four with two accumulation steps to micro-batch two with four accumulation steps after step 102000, preserving the update token budget and data order. Fused AdamW uses learning rate 3×10^{-4}, $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1, and norm clip 1.0; 7,620 warmup updates precede cosine decay to 3×10^{-5}. The 152,590 updates consume $20{,}000{,}276{,}480$ tokens per original run. The separate 3B study uses sample-10BT, one GPU, micro-batch eight, accumulation four, seed 42, BF16, and the same tokens/update. Its 5B cosine schedule has 1,908 warmup and 38,147 planned updates, but stops at step 22,889 ($3{,}000{,}107{,}008$ tokens). The main comparison and the 3B ablations use different training budgets.

Language modeling and general tasks. PPL is $\exp(\text{total token NLL}/\text{scored tokens})$. For the main approximately 20B-token comparison, FineWeb-Edu is evaluated in non-overlapping 4K windows and WikiText-103 in non-overlapping 2K windows. In the separate 3B-token ablation study, both datasets use 4K windows: held-out FineWeb-Edu comprises 1,220 complete windows (4,995,900 scored tokens), and WikiText-103 comprises 69 complete windows (282,555 scored tokens). All six variants are evaluated on identical windows and token counts. The eleven main likelihood tasks are LAMBADA, PIQA, HellaSwag, BLiMP, WinoGrande, ARC-Easy, SciQ, OpenBookQA, BoolQ, SocialIQA, and ARC-Challenge (Clark et al., 2019; Sap et al., 2019; Clark et al., 2018). General tasks use zero-shot likelihood accuracy; PIQA, HellaSwag, ARC-Easy, SciQ, and OpenBookQA use normalized accuracy in both tables. The eleven-task mean at the approximately 20B-token endpoint and the eight-task mean in the separate 3B-token study correspond to distinct training endpoints. At 3B, the first eight general tasks above, FineWeb-Edu and WikiText-103 perplexity, S-NIAH-1, and SQuAD-Completion/FDA/SWDE are evaluated on matched inputs across variants. Transformer++ uses batch size 64 for likelihood tasks and variable-length left-padded cached generation.

Long-context modeling and in-context retrieval scoring. The four selected RULER-style synthetic in-context retrieval tasks use 256 matched instances per length and at most 128 generated tokens. All seven reported checkpoints have 2K/4K/8K scores; Transformer++, LCRA, GDN-2, Mamba-3, and Liger also have 16K scores. Here, *oracle* denotes a short control prompt containing the needle sentences and query without the long distractor context. Each task has 64 oracle examples and 256 examples at each evaluated length; long-context scores appear in Table 2, and oracle scores in Table 7. The 3B study evaluates S-NIAH-1 at oracle/2K/4K/8K, with 4K as its training length; its oracle scores appear in Table 8. PG-19 uses three common anchors in each of 77 books, fresh state per window, and identical final 256-token targets. Thus $\Delta\text{NLL}(L) = \text{NLL}(L) - \text{NLL}(2\text{K})$ reports the observed change on fixed targets, not stateful book-level PPL. The six real-world in-context retrieval tasks are SQuAD-Completion, FDA, and SWDE (Rajpurkar et al., 2016; Arora et al., 2024a), plus NQ, TriviaQA, and DROP with Just Read Twice adapters (Kwiatkowski et al., 2019; Joshi et al., 2017; Dua et al., 2019; Arora et al., 2024b). For real-world in-context retrieval, we use at most 48 greedy generated tokens, newline stopping, and literal answer containment. The six-task real-world in-context retrieval mean is separate from the eleven-task likelihood mean. Transformer++ and original NHA retain their original RoPE settings.

Model-specific evaluation settings. NAtS-L's 8K synthetic retrieval evaluation uses cache recovery; its PG-19 evaluation does not use generation cache. Liger's generated-answer evaluation uses an evaluation-only cache-container correction. Its 16K NIAH evaluation uses batch size one on the same fixed evaluation inputs.

For GDN-2 and Mamba-3, batch sizes are adjusted by sequence length as follows. Their NIAH oracle/2K/4K/8K/16K batches are 64/64/64/32/16 and 64/32/16/8/4, respectively; their PG-19 2K/4K/8K/16K batches are 64/64/32/16 and 64/32/16/8. Transformer++ uses its original RoPE setting; its NIAH batches for oracle/2K/4K/8K/16K are 64/32/16/8/4, and its PG-19 batches are 64/64/32/16. For 16K PG-19, original NHA uses batch size 16 and unchanged RoPE, while NAtS-L uses batch size one without generation cache; both evaluate the same 231 fixed targets.

C Additional evaluation results

Short-prompt oracle controls. Tables 7 and 8 report the short-prompt controls for the main comparison and the separate 3B-token ablations. Liger solves all 64 oracle examples on S-NIAH-1 and S-NIAH-2, but only 6 of 64 on MK-NIAH-1. Thus its low long-context scores on the first two tasks cannot be explained by a failure on the corresponding short prompts, whereas the multikey task is already difficult for this checkpoint without distractors. Oracle prompts also differ in length and distractor content, so these controls do not isolate a single cause of long-context errors.

Table 7: Short-prompt oracle accuracy for the approximately 20B-token checkpoints (%, higher is better). Each task has 64 examples scored by the `string_match_all` criterion.

Model	S-NIAH-1	S-NIAH-2	S-NIAH-3	MK-NIAH-1
Long-Range with KV Access				
Transformer++	98.44	100.00	89.06	89.06
NHA	100.00	100.00	98.44	42.19
NAtS-L	96.88	96.88	84.38	25.00
Long-Range with Fixed-State Memory				
GDN-2	100.00	100.00	95.31	75.00
Mamba-3	100.00	100.00	96.88	46.88
Liger	100.00	100.00	78.12	9.38
LCRA (ours)	100.00	100.00	100.00	78.12

Table 8: S-NIAH-1 short-prompt oracle accuracy for the separate 3B-token ablation study (%, higher is better). All six variants use the same 64 examples.

Variant	Oracle acc↑
LCRA (full)	95.31
w/o remote	96.88
w/o residual memory	85.94
remote-only (same FFN width)	95.31
remote-only (matched params)	95.31
full-value target	93.75

Table 9: PG-19 fixed-target perplexity across context lengths. Lower is better. Original NHA retains its original RoPE setting. † denotes the converted NHA variant.

Model	2K ppl	4K ppl	8K ppl	16K ppl
Long-Range with KV Access				
Transformer++	36.88	36.98	6595.02	9931.98
NHA	53.92	51.36	231.31	346.26
NAtS-L	**34.77**	**34.51**	**34.90**	**35.19**
Long-Range with Fixed-State Memory				
GDN-2	39.24	37.99	37.46	37.43
Mamba-3	41.31	40.47	40.25	40.24
Liger	45.07	44.67	44.81	44.92
NHA†	76.67	76.64	77.15	80.65
LCRA (ours)	**34.01**	**32.67**	**32.16**	**32.09**

Table 10: Change in PG-19 fixed-target NLL relative to 2K context. Values are in nats/token; lower is better, and negative values indicate improved prediction with additional context. † denotes the converted NHA variant.

Model	4K−2K	8K−2K	16K−2K
Long-Range with KV Access			
Transformer++	+0.00257	+5.18635	+5.59579
NHA	**−0.04856**	+1.45635	+1.85976
NAtS-L	−0.00751	**+0.00383**	**+0.01211**
Long-Range with Fixed-State Memory			
GDN-2	−0.03238	−0.04635	−0.04717
Mamba-3	−0.02041	−0.02582	−0.02622
Liger	−0.00891	−0.00590	−0.00346
NHA†	−0.00041	+0.00626	+0.05062
LCRA (ours)	**−0.04010**	**−0.05583**	**−0.05786**

D SHARED VALUE CONFIGURATION STUDY

We compare two early LCRA value configurations after 200,015,872 training tokens. Both use 18 layers, hidden width 1024, eight heads, a 64-token local window, local value dimension 128, and remote key/value dimensions 32/64. They train on the same 2K FineWeb-Edu sequence order at 65,536 tokens per update for 3,052 updates under the same cosine schedule, with separate model seeds 42 and 43. Evaluation uses matched 2K windows within each seed: 2,441 FineWeb-Edu windows and 139 WikiText-103 windows, each with 2,047 scored tokens.

The independent-values configuration projects and convolves remote values separately, followed by SiLU. The shared shifted-values configuration adapts the shifted local value to the remote value dimension and omits the remote value convolution and SiLU. Both retain separate local and remote query/key projections.

Table 11: Perplexity of two early LCRA value configurations after 200M training tokens. FW-Edu and Wiki denote FineWeb-Edu and WikiText-103; lower is better.

Configuration	**Params (M)**	**Seed 42**		**Seed 43**	
		FW-Edu ppl↓	**Wiki** ppl↓	**FW-Edu** ppl↓	**Wiki** ppl↓
Independent values	303.60	<u>41.21</u>	<u>147.96</u>	<u>41.03</u>	<u>141.85</u>
Shared shifted values	295.30	**40.31**	**143.61**	**39.94**	**134.95**

The shared shifted-values configuration has lower perplexity on both datasets under both seeds. Because remote value preprocessing and parameter count also differ, this comparison supports the complete configuration rather than isolating the effect of sharing the value source.

E COMPUTATIONAL SCOPE AND REPRODUCIBILITY

Figure 3 and Tables 12 and 13 report measurements on one NVIDIA RTX PRO 6000 Blackwell Server Edition 96GB, using the approximately 720M-parameter, 20B-token checkpoints. NHA is the original model with full-attention layers. Each point runs in a fresh process with two warmups and three timed measurements; reported values are medians. CUDA synchronization bounds timing, which excludes model loading, first compilation, tokenizer processing, and archival I/O. Parameters are FP32 with BF16 autocast. These short measurements do not establish variability across independent processes.

Training. The 2K/4K/8K/16K input lengths use batches 8/4/2/1, GA=1, and 16,384 input tokens per update on the same FineWeb-Edu input stream. Timing includes forward and backward passes, gradient clipping, and a newly initialized fused AdamW optimizer (learning rate 3×10^{-5}, $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1, norm clip 1). Updated weights are discarded. LCRA uses `torch.compile` and a coarse-forward TF32 optimization; the other models use their native operators without whole-model compilation. Global TF32 is disabled. All 28 training points complete with finite loss and gradients and no OOM. This is a short performance test, not a new long-training quality evaluation.

Inference. Batch-one requests use the same fixed prompt at each length and greedily generate exactly 128 tokens without early EOS. Each request starts with fresh state. Prefill includes the first generated token; decode covers the next 127. LCRA uses CUDA Graph, NHA uses full-prefix recomputation, and the other models use cached generation. LCRA's four initial Graph requests match the corresponding eager outputs token for token. Stable decode throughput excludes each request's graph capture; inclusive decode and full-request latency retain it. Full-request timing includes prefill, capture, decoding, sampling, and per-step finite-logit checks. Graphs are not reused across requests. Thus the comparison includes execution-backend differences.

At 4K, LCRA's prefill throughput is 30.0K input tokens/s versus 97.3K for Mamba-3, while its decode throughput is 98.5 versus 50.7 generated tokens/s when LCRA's Graph capture is included. These phase timings explain how slower prefill and lower full-request latency coexist in the measured implementations.

NAtS-L at 16K produces all-NaN logits on its fourth generated token in the first warmup. This point has no valid timing or memory result and is marked NF; no cache recovery, NaN replacement, or full-prefix fallback is allowed. The remaining 27 points complete all requests with finite logits and identical output hashes within each point. Peak allocated memory is measured by PyTorch over the node, including warmup (LCRA resets peaks after the initial Graph/eager comparison). These peaks are not persistent recurrent-state or KV-cache sizes.

Runtime. Measurements use PyTorch 2.9.0+cu128, Triton 3.5.0, and driver 580.82.09. Liger and NAtS-L use FLA 0.4.0; the others use FLA 0.5.2. Liger uses Transformers 4.45.2 and the others 4.56.2. The different compilation, precision, and cache paths preclude interpreting these numbers as equally optimized architectural limits. Anonymous supplementary material provides implementation and evaluation code.

Table 12: Training throughput (K input tokens/s, ↑) and full-request latency (s, ↓). NF denotes nonfinite failure.

Model	Training throughput				Request latency			
	2K	4K	8K	16K	2K	4K	8K	16K
Long-Range with KV Access								
Transformer++	**39.096**	**36.811**	**33.202**	27.688	**2.184**	**2.255**	**2.216**	**2.359**
NHA	30.410	30.082	29.362	**28.012**	6.224	6.210	9.190	19.340
NAtS-L	29.410	27.812	24.861	19.688	3.320	3.345	3.490	NF
Long-Range with Fixed-State Memory								
GDN-2	28.842	28.482	27.655	26.134	2.701	2.674	2.775	2.870
Mamba-3	**37.511**	**37.657**	**35.761**	**31.032**	2.582	2.547	2.707	2.705
Liger	21.619	21.485	21.583	21.267	2.614	2.649	2.712	2.810
LCRA	16.529	16.240	16.064	15.746	**1.387**	**1.425**	**1.558**	**1.850**

Table 13: Peak allocated GPU memory (GiB, ↓) during training and inference. NF denotes nonfinite failure.

Model	Training				Inference			
	2K	4K	8K	16K	2K	4K	8K	16K
Long-Range with KV Access								
Transformer++	**29.350**	**29.348**	**29.348**	**29.348**	4.469	4.750	5.468	6.910
NHA	43.092	43.115	43.155	43.239	4.210	4.421	4.783	**5.502**
NAtS-L	43.859	43.855	43.857	43.852	**4.076**	**4.254**	**4.698**	NF
Long-Range with Fixed-State Memory								
GDN-2	43.997	43.998	43.998	43.998	**4.069**	**4.156**	**4.450**	**5.036**
Mamba-3	**35.036**	**35.036**	**35.036**	**35.036**	4.303	4.615	5.352	6.830
Liger	64.609	64.611	64.617	64.530	4.429	4.819	5.707	7.499
LCRA	71.023	71.708	71.899	71.661	4.509	5.034	6.087	8.209

LCRA Graph capture takes 0.242/0.238/0.233/0.236 seconds at 2K/4K/8K/16K, respectively.