PAY AT THE EVENT: EVENT-CONDITIONED RECURRENT STATE TRACKING AND ASSOCIATIVE RECALL

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce an event-conditioned recurrent block that explicitly separates transition computation from associative storage. FLINT maintains a low-cost diagonal carry at every token while applying expressive Householder transformations only at selected events. When event gates are determined before scanning, the recurrent dependency chain can be reorganized around event boundaries, making expensive transition work proportional to the number of events. EGAM provides fixed-size associative memory through binary addressing, gated additive writes, and normalized slot reads. In a shared-backbone block benchmark, the combined block achieves $1.2\times$ faster forward execution than Gated DeltaNet-2 at known-mask density 0.1 and length 2048. Replacing memory-projection multiplications with additive and bitwise operations reduces mixer multiply-accumulates (MACs) from 0.865M to 0.469M per token and layer, with 1.262M accumulate/bit operations, excluding the MLP. Across three seeds, mean WikiText-103 perplexity changes from 26.705 to 26.709. On FineWeb-Edu, a 122.54M-parameter model trained on 1.007B tokens retains validation perplexity (29.328 to 29.355, one seed) while reducing mixer MACs from 1.838M to 1.047M. The combined task state is independent of context length and, at length 1024, is approximately one-eighth the size of attention's KV cache. Learned-gate and joint tasks evaluate tracking and recall within these budgets. We will release our code upon acceptance.

1 INTRODUCTION

Attention's key–value cache grows with context length (Vaswani et al., 2017); structured and diagonal state-space models retain a fixed state (Gu et al., 2022a;b). Simplified recurrent layers enable parallel processing (Smith et al., 2023; Orvieto et al., 2023), while richer transitions improve tracking at greater per-token cost (Merrill et al., 2024; Siems et al., 2025). However, existing matrix-state models execute non-diagonal, higher-rank transformations uniformly across all sequence indices $t \in [1, L]$. Enforcing dense state updates at timesteps where state dynamics reduce to identity-like or diagonal drift incurs redundant FLOP and MAC overhead without contributing to state capacity. We ask: *can a recurrent block concentrate expressive update work at selected events and support content-addressable recall with a bounded storage budget?*

In *event-conditioned computation*, every token updates a diagonal state; learned gates select richer transformations and associative writes. *Tracking* composes updates in order, while *recall* retrieves an association when its key reappears (Arora et al., 2024a).

Our block combines FLINT, an event-conditioned recurrent mixer, and EGAM (Event-Gated Associative Memory), a binary-addressed associative store, as successive pre-norm residual updates followed by one MLP. Binary memory reads and two-term additive power-of-two (APoT) projections (Li et al., 2020) replace much of the additional multiplication with sparse additions and shift-add arithmetic. The proposed design targets recurrent inference and bounded-state operation; efficient linear-time training of associative retrieval remains an open direction.

We make three contributions:

- **Event-proportional transition computation.** FLINT couples a per-token diagonal carry with gated Householder transformations and an equivalent event-boundary schedule, making expensive transition work proportional to the realized event count.

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.

- **Fixed-size associative recall.** EGAM combines binary addressing, gated additive writes, and normalized slot reads to provide associative retrieval under a context-independent storage budget.
- **Resource-quality characterization.** Controlled timing, learned-gate tasks, and parameter-matched language models characterize the tradeoffs between transition computation, storage budget, arithmetic cost, and language-model quality.

2 RELATED WORK

Efficient sequence models. Linear attention and Performer replace pairwise attention with feature-map accumulation (Katharopoulos et al., 2020; Choromanski et al., 2021). Mamba and its successors develop selective state-space updates and efficient execution (Gu & Dao, 2024; Dao & Gu, 2024; Lahoti et al., 2026); gated linear attention and xLSTM develop gated matrix memories (Yang et al., 2024a; Beck et al., 2024).

Expressive recurrent transitions. Transition structure affects tracking (Merrill et al., 2024) and parallelizability (Merrill et al., 2026). DeltaNet and gated delta-rule models use structured matrix-state updates (Yang et al., 2024b; 2025a; Hatamizadeh et al., 2026). Negative eigenvalues expand state-tracking capabilities (Grazzi et al., 2025); DeltaProduct composes Householder factors (Siems et al., 2025), while PaTH uses accumulated Householder transformations for positional encoding (Yang et al., 2025b). FLINT couples this transition family to vector-state diagonal carry, learned event selection, and an event-boundary schedule.

Adaptive computation. ACT allocates computation per input (Graves, 2016); Skip RNN learns when to update state (Campos et al., 2018). SpikeMamba uses LIF dynamics and spike-triggered accumulations for biomedical sequence modeling (Lee et al., 2026). Here, diagonal carry remains active; a prefix-computed shadow supplies context for selecting non-diagonal events before their traversal.

Associative memory and recall. Sparse distributed memory uses overlapping address regions (Kanerva, 1988); modern Hopfield networks and fast-weight models connect associative retrieval to attention (Ramsauer et al., 2021; Schlag et al., 2021). Memorizing Transformers retrieve from an external key–value store (Wu et al., 2022), and Gated Slot Attention uses recurrent slot memories (Zhang et al., 2024). Test-time training and Titans update learnable memories (Sun et al., 2025; Behrouz et al., 2025); Raven studies sparse memory routing (Afzal et al., 2026). Recall benchmarks characterize the resulting quality–efficiency tradeoffs (Arora et al., 2024a;b). EGAM uses a fixed random binary codebook, gated additive writes, and normalized slot reads to complement event-conditioned tracking.

3 ALLOCATING TRANSITION WORK AND ASSOCIATIVE STORAGE

Block organization. The language-model block applies successive pre-norm residual updates $x \leftarrow x + F(\mathrm{Norm}(x))$ for PDiag, FLINT, EGAM, and an MLP (Figure 1a). Each branch receives the updated stream; x_t denotes its normalized input. Superscripts P, F, E are branch labels. PDiag supplies additive input through a state $p_t \in \mathbb{R}^{d_P}$:

$$p_t = a_t^P \odot p_{t-1} + u_t^P, \qquad y_t^P = W_o^P p_t + b_o^P, \qquad p_0 = 0, \tag{1}$$

where $a_t^P = \mathrm{sigmoid}(W_a^P x_t + b_a^P)$ and $u_t^P = W_u^P x_t + b_u^P$. FLINT separates nonexpansive composition from PDiag's additive input; EGAM provides addressable storage with its own budget.

3.1 FLINT: DIAGONAL CARRY AND GATED COMPOSITION

Let $h_t \in \mathbb{R}^d$, with learned h_0. Input-dependent diagonal carry and a hard event gate define

$$\tilde{h}_t = a_t \odot h_{t-1}, \qquad a_t = \tanh(W_a x_t), \tag{2}$$

$$H_t = H_{t,k} \cdots H_{t,1}, \qquad \beta_{t,i} = 2\,\mathrm{sigmoid}(W_b x_t)_i, \qquad \|v_{t,i}\|_2 = 1, \tag{3}$$

$$h_t = [(1 - g_t)I + g_t H_t]\tilde{h}_t, \qquad g_t \in \{0, 1\}. \tag{4}$$

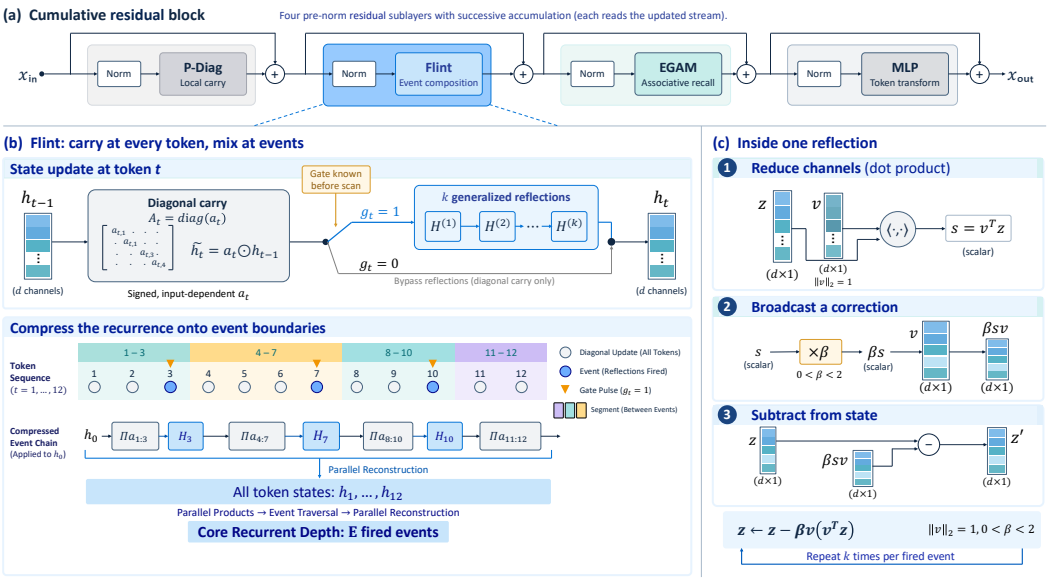

Figure 1: **FLINT: event-proportional composition and segment execution.** (a) Successive pre-norm residual sublayers share one MLP. (b) Per-token diagonal carry connects events with k generalized reflections, followed by parallel token reconstruction. (c) A reflection uses a dot product and vector correction. Depth E excludes projections, gates, and gate-training counterfactuals.

Here $H_{t,i} = I - \beta_{t,i} v_{t,i} v_{t,i}^\top$, applied in order $i = 1, \ldots, k$. Directions $v_{t,i}$ are normalized input projections; affine-head biases are suppressed. The output is $y_t^F = W_o^F h_t + b_o^F$. Input enters the composition state through its transition and gate, with additive injection provided separately by PDiag. A factor interpolates between identity and a full reflection as β varies from 0 to 2. Applying $z \leftarrow z - \beta v(v^\top z)$ costs $O(d)$ without forming a dense matrix, hence $O(kd)$ per fired event, excluding projections.

Expressiveness per event. Let $D_t = \mathrm{diag}(a_t)$ and $T_t = [I + g_t(H_t - I)]D_t$. Each factor contributes at most one direction away from identity, so

$$\mathrm{rank}(H_t - I) \le k, \qquad \mathrm{rank}(T_t - D_t) \le g_t k. \tag{5}$$

Thus k controls the rank of the event correction and its arithmetic cost; $g_t = 0$ leaves only diagonal carry. Appendix B.3 proves the bound and gives group-representation reference budgets.

Norm control. The same factorization bounds state amplification without a dense matrix constraint. For a unit direction v,

$$\|(I - \beta v v^\top) z\|_2^2 = \|z\|_2^2 - \beta(2 - \beta)(v^\top z)^2. \tag{6}$$

Thus $0 \le \beta \le 2$ makes each factor nonexpansive. Together with $|a_{t,j}| \le 1$, this gives $\|h_t\|_2 \le \|\tilde{h}_t\|_2 \le \|h_{t-1}\|_2$ for either gate decision. This is a bound on the composition state; the residual stream also receives the PDiag, memory, and MLP outputs. Retention over longer sequences is evaluated separately from this amplification bound.

3.2 EGAM: BINARY-ADDRESSED ASSOCIATIVE WRITES AND READS

Each of N slots has a fixed random binary address $b_m \in \{0,1\}^{d_k}$ and mutable accumulators $M_{t,m} \in \mathbb{R}^{d_v}$ and $C_{t,m}$, initialized to zero. Thresholding a learned projection gives binary write and query codes. The write path first applies a causal two-tap convolution initialized as a one-token shift; the query uses the current input with the same projection and no convolution. This initially binds a value to its preceding key while allowing the key features to adapt. Writing selects slots by $A_{t,m}^w =$

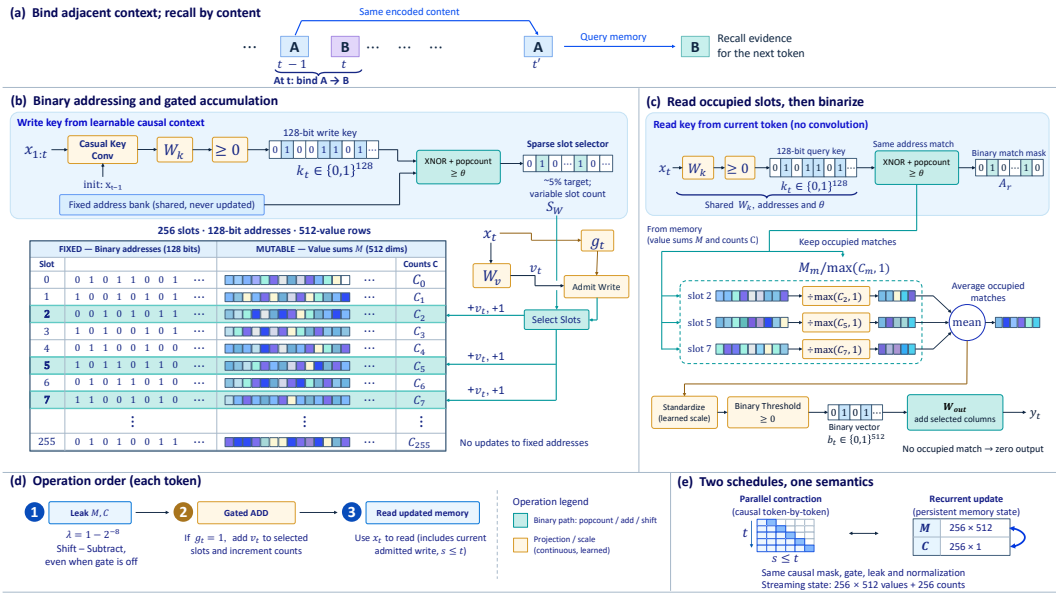

Figure 2: **EGAM: fixed-state recall through binary addressing and additive writes.** (a–b) A fixed address bank routes gated writes to value sums and counts. (c–d) Leak, write, and normalized binary read occur in order. (e) Parallel and recurrent schedules evaluate the same read. WikiText uses 128-bit keys, 256 slots, and 512-dimensional values.

$\mathbf{1}[\mathrm{popcount}(\mathrm{XNOR}(k_t, b_m)) \geq \theta]$; A^r applies the same rule to the query code. With $v_t = W_v x_t$ and a binary write gate g_t^w,

$$M_{t,m} = \lambda M_{t-1,m} + g_t^w A_{t,m}^w v_t, \qquad C_{t,m} = \lambda C_{t-1,m} + g_t^w A_{t,m}^w. \tag{7}$$

Every token leaks both accumulators, adds any admitted write, then reads the updated memory, including the current write. The read averages normalized values over occupied matching slots:

$$r_t = \frac{1}{\max(N_t, 1)} \sum_m A_{t,m}^r \mathbf{1}[C_{t,m} > 0] \frac{M_{t,m}}{\max(C_{t,m}, 1)}, \qquad N_t = \sum_m A_{t,m}^r \mathbf{1}[C_{t,m} > 0]. \tag{8}$$

Standardization across value channels, with learned scalar scale and bias, produces a binary read. Its output is a sum of selected columns:

$$b_t^{\mathrm{out}} = \mathbf{1}[\gamma\, \mathrm{Std}(r_t) + \delta \geq 0], \qquad y_t^E = \mathbf{1}[N_t > 0] \sum_{j : b_{t,j}^{\mathrm{out}} = 1} W_{o,:,j}^E. \tag{9}$$

Here Std denotes channel standardization (Appendix C.6); an empty match returns zero. Address matching uses popcounts, writes use selected additions, and Eq. (9) needs no input–weight multiplication. Both language-model configurations use $N = 256$, $d_k = 128$, and $\theta = 64 + \Phi^{-1}(0.95)\sqrt{32}$, calibrated to a nominal 5% activation fraction (Φ: standard normal CDF). Leak $\lambda = 1 - 2^{-8}$ uses shift–subtract; d_v is 512 on WikiText and 768 on FineWeb-Edu. Synthetic recall also uses $\lambda = 1$.

3.3 LEARNING THE ALLOCATION

Event gate. The language model uses a shadow-state threshold,

$$\hat{h}_t = \left(\prod_{s=1}^{t} a_s \right) \odot h_0, \qquad z_t = w_g^\top x_t + w_{gh}^\top \hat{h}_t + b, \qquad g_t = \mathbf{1}[z_t > 0]. \tag{10}$$

The shared-W_a shadow is prefix-computable. Gates condition on it and the contextual residual x_t, without direct dependence on this layer's reflected state, making them available before traversal. Synthetic controls use input-only, leaky integrate-and-fire (LIF), and true-state gates. LIF adds a scalar side-scan; true-state gates depend on the evolving composition state (Appendix C).

Write gate and surrogates. The LM write gate uses input-local features: $\mu_t = \frac{1}{5}\sum_{j=0}^{4} x_{t-j}^{\odot 2}$ and $\Delta_t = |x_t - x_{t-1}|$, with zero padding and $\Delta_1 = 0$. Its score is $z_t^w = w^\top[\mu_t; \Delta_t] + b$ and $g_t^w = \mathbf{1}[\mathrm{sigmoid}(z_t^w) \geq 0.5]$, available before memory updates. For the LM event/write gates, hard forward decisions use the straight-through form $g_{\mathrm{hard}} + p - \mathrm{sg}(p)$ with $p = \mathrm{sigmoid}(z)$ and stop-gradient sg. Key, address, and binary-read thresholds instead use arctangent surrogates specified in Appendix C.6.

The LM objective couples next-token prediction to event rate and a minimum write rate:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \eta_F(\bar{g}_F - \rho_F)^2 + \frac{\eta_W}{n_E}\sum_{\ell=1}^{n_E}\left[\max\left(0, 1 - \frac{\bar{g}_\ell^w}{\rho_W}\right)\right]^2. \tag{11}$$

Here $\bar{g}_F$ averages event gates over layers, batch, and tokens; $\bar{g}_\ell^w$ averages writes in memory layer ℓ, and n_E counts memory layers. The event target is $\rho_F = 0.1$; $\rho_W = 0.05$ is a write *floor*, not a sparsity target. Coefficients and optimizer settings are in Appendix C.4. Synthetic tracking uses target 1.2ρ without event-position labels; oracle gates receive event indicators only as references.

4 Efficient Execution and Resource Scaling

4.1 Exact Event-Boundary Execution

When hard gates are available before scanning the composition state, let $e_1 < \cdots < e_E$ be the fired positions, $e_0 = 0$, and $s_j = h_{e_j}$. Algorithm 1 folds diagonal segments and traverses event boundaries. With $j(t) = |\{j : e_j \leq t\}|$, it reconstructs every token as

$$h_t = \left(\prod_{s=e_{j(t)}+1}^{t} a_s\right) \odot s_{j(t)}. \tag{12}$$

Empty products are ones, covering $E = 0$ and trailing segments.

Algorithm 1 Event-boundary FLINT forward for precomputed hard gates

Input: h_0, diagonal factors $a_{1:L}$, hard gates $g_{1:L}$, and event directions/strengths $(v_{e_j,i}, \beta_{e_j,i})$.
Extract $e_1 < \cdots < e_E$ where $g_t = 1$; set $e_0 = 0$, $s_0 = h_0$.
Compute $q_j = \prod_{s=e_{j-1}+1}^{e_j} a_s$ and within-segment prefixes in parallel, including the trailing segment.
for $j = 1, \ldots, E$ **do**
  $z \leftarrow q_j \odot s_{j-1}$
  for $i = 1, \ldots, k$ **do** $z \leftarrow z - \beta_{e_j,i} v_{e_j,i}(v_{e_j,i}^\top z)$
  $s_j \leftarrow z$
Reconstruct all h_t by Eq. (12) in parallel; return $y_t^F = W_o^F h_t + b_o^F$.

Exactness and work. Algorithm 1 equals Eq. (4) in exact arithmetic for fixed coefficients and hard gates: $s_{j-1} = h_{e_{j-1}}$ implies $s_j = H_{e_j}(q_j \odot s_{j-1}) = h_{e_j}$; induction and Eq. (12) cover the intervening tokens. With realized rate $\rho_g = E/L$, its state work, excluding projections and gate computation, is

$$W_{\mathrm{state}} = O(Ld) + O(Ekd), \qquad W_{\mathrm{state}}/L = O(d) + O(\rho_g kd). \tag{13}$$

For fixed k, traversal has E dependent stages per sequence, or $\max_b E_b$ in a padded batch. Prefixes, projections, gates, and reconstruction remain token-wise. Gate availability follows Section 3.3; Appendix C.6 gives scan numerics.

Learning a gate. For the relaxed gate, holding other factors fixed gives

$$\frac{\partial[\tilde{h}_t + g_t(H_t - I)\tilde{h}_t]}{\partial g_t} = (H_t - I)\tilde{h}_t. \tag{14}$$

The surrogate therefore uses counterfactual transformations even at unfired positions. Fixed-mask event-only timing and learned-gate training throughput measure different execution paths.

Table 1: **Shared-backbone block latency.** Batch 128, width 128, known-mask $\rho = 0.1$; median ms on an NVIDIA RTX PRO 6000 Blackwell Workstation Edition GPU (96 GB). Synthetic inputs; bf16 autocast. All rows include optimized PDiag, normalization, and residuals; MLP/embedding/head excluded.

	Task state	Forward	Forward + backward	
Block	elements	$L = 2048$	$L = 128$	$L = 2048$
PDiag + FLINT (ours)	**128**	**1.92**	**2.29**	**8.50**
PDiag + FLINT + EGAM (ours)	33,152	4.48	4.35	21.39
PDiag + DeltaNet	8,256	3.55	2.71	12.22
PDiag + Gated DeltaNet	8,256	3.68	3.17	13.47
PDiag + Gated DeltaNet-2	8,256	5.39	3.57	21.54
PDiag + Gated DeltaProduct ($k = 4$)	8,256	7.75	3.47	25.51

4.2 PARALLEL TRAINING AND RECURRENT RECALL

Expanding Eq. (7) and substituting into the normalized read yields a causal contraction over values:

$$r_t = \frac{\sum_{s=1}^{t} \omega_{t,s} v_s}{\max(N_t, 1)}, \qquad \omega_{t,s} = \lambda^{t-s} g_s^w \sum_m \frac{A_{t,m}^r \mathbf{1}[C_{t,m} > 0] A_{s,m}^w}{\max(C_{t,m}, 1)}. \tag{15}$$

With query-time slot counts, this contraction equals the recurrent read (Appendix B.2). For one head and batch size B, the dense schedule uses $O(BL^2(N + d_v))$ time and $O(BL^2 + BL(N + d_v))$ activation storage, excluding projections/addressing. Streaming retains $BN(d_v + 1)$ memory-state elements, independent of L. The gated value-write work is $d_v \sum_t g_t^w \|A_t^w\|_0$ additions, while addressing, leak, and query reads remain token-wise.

4.3 ARITHMETIC REPRESENTATION AND STATE BUDGETS

To reduce input-projection multiplication beyond the selected additions in Eq. (9), we apply post-training two-term APoT (Li et al., 2020):

$$\widetilde{w} = s(\epsilon_1 2^{e_1} + \epsilon_2 2^{e_2}), \qquad \epsilon_i \in \{-1, 0, 1\}. \tag{16}$$

The targets are the key convolution, tied key/query projection, value projection, and write-gate projection. Power-of-two products admit shift-add evaluation; operand representations, shared scales, and normalization are specified separately (Appendix A).

Let P denote their MAC budget, R_{FP}, R_2 the respective residual memory budgets, and A_0 the original accumulate/bit budget:

$$C_{\mathrm{MAC}}^{E,\mathrm{FP}} = P + R_{\mathrm{FP}}, \qquad C_{\mathrm{MAC}}^{E,2\mathrm{term}} = R_2, \qquad C_{\mathrm{AC}}^{E,2\mathrm{term}} = A_0 + 2P. \tag{17}$$

At width 512, $P = 395264$, $R_{\mathrm{FP}} = 8601.6$, and $R_2 = 7577.6$: a 98.1% reduction in the memory branch's MAC budget. The residual covers read variance, gate-energy features, and per-slot normalization. FP32 APoT additions cost 0.9 pJ each; the full operation-type ledger yields 3.24756 μJ/token/layer for WikiText combined APoT (counts and exclusions in Appendix A).

For PDiag width d_P, FLINT width d, and N memory slots, the combined task-state count is

$$S_{\mathrm{task}} = d_P + d + N(d_v + 1), \tag{18}$$

independent of context length L, whereas attention's KV cache has $2d_{\mathrm{model}}L$ elements. Event rate, arithmetic representation, and storage size thus control distinct resource budgets.

5 EXPERIMENTS

Synthetic state-tracking models use $d_{\mathrm{model}} = 128$, $d_{\mathrm{state}} = 64$, and $k = 4$ unless varied. Training length is 128 and batch size is 128. We report per-position and final-position accuracy separately. A *transition* records a training-accuracy crossing, distinct from held-out accuracy; best-cell results include transition counts. Appendix C gives training and evaluation details.

Table 2: **Learned event selection on sparse S_5.** $\rho = 0.1$, $L = 128$. Accuracy is the best cell over the learning-rate and seed grid; transition counts report all cells in each gate battery. Inert fire is the selected cell's firing rate on inactive group tokens. Oracle event positions are a reference; learned gates receive an aggregate rate target but no position labels.

Task	Gate	Position / final accuracy	Transitions	Inert fire
Uniform	Oracle	1.000 / 1.000	8/8	–
	LIF	0.9998 / 0.9985	8/8	–
	Input threshold	0.9983 / 0.9961	2/6	–
	MLP router	0.868 / 0.690	3/4	–
Contextual	Oracle	1.000 / 1.000	2/4	–
	Input threshold	0.425 / 0.092	0/4	1.000
	True-state threshold	0.9999 / 1.0000	3/4	0.016
	Shadow-state threshold	0.9992 / 0.9980	1/4	0.000

Table 3: **Joint tracking and recall.** Query accuracy ranges over four cells per row (two learning rates and two seeds). Training length is 128. Store-once keys isolate composition plus retrieval; rebinding tests repeated writes to the same key in the additive store.

Key protocol	Write gate	Tracking gate	$L = 128$	$L = 512$
Store once	Oracle	Oracle	0.998–1.000	0.938–0.969
	Learned	Oracle	0.995–0.999	0.931–0.994
	Learned	Learned	0.998–1.000	0.969–0.992
Rebinding	Oracle	Oracle	0.835–0.905	0.308–0.385
	Learned	Oracle	0.699–0.860	0.257–0.352

5.1 Measured efficiency of event-based execution

Table 1 includes PDiag, normalization, and residuals for all rows. Separate FLINT sequential/segment checks agree within 3×10^{-8} at $L = 8192$, with identical predictions over 131k positions; fused-backward error is below 7×10^{-6} (Appendix D.3).

At $L = 2048$, FLINT is $1.8\times$ faster forward and $1.4\times$ faster forward–backward than DeltaNet; combined adds 256-slot recall at 4.48 ms forward (Table 1; state sizes differ). A same-recurrence control freezing weights, inputs, and hard gates gives FP32 recurrence speedups of $0.947\times$ at $L = 1024$ and $1.837\times$ at $L = 4096$ (batch 16; Appendix E.3).

5.2 Tracking accuracy under learned event budgets

A group word problem asks for the running product of tokens, each non-identity with probability ρ. Contextual TOGGLE tokens switch whether subsequent group tokens take effect, making eventness undetermined by the current token alone.

Uniform and contextual events. The LIF gate reaches 0.9985 best final accuracy, transitions in all eight uniform-event runs, and recalls at least 0.999 of events (Table 2). Contextual true-state and shadow-state gates reach near-perfect best-cell accuracy; the input-only threshold does not separate active from inert tokens. Shadow context is computed before the scan; transition counts report optimization reliability.

Density, length, and reflection budget. Learned LIF gates reach best final accuracies of at least 0.991 over $\rho = 0.02$–0.2, a $10\times$ range. Appendix D.1 reports the full density/reflection sweep and held-out length degradation; Appendix D.2 analyzes fired updates and gate features.

5.3 Tracking and recall within a bounded state budget

Recall with constant value-state size. On multi-query associative recall, EGAM reaches 0.998–0.999 mean query accuracy with 4–32 pairs over eight seeds. At $L = 256$–2048, accuracy remains 0.998–0.999 with 32768 value-state elements and a fixed 16 pairs (Appendix D.3). Pair count and sequence length test interference and retention, respectively, at the same state allocation.

Table 4: **Language-model quality and mixer efficiency at 236M tokens.** PPL, extraction (%), and Train are means over three seeds. Post-training APoT shares FP Train; xLSTM budgets are layer means.

	Quality–energy		Analytic mixer budget			Extraction (%)		Training
Mixer configuration	PPL ↓	E_{arith} ↓ (μJ)	MAC (M)	AC (M)	Task state (elements)	SWDE	SQuAD	Train (k tok/s)
Diagonal reference	40.739	0.90	0.197	0.000	128.0	1.80	1.17	116.1
Diagonal + FLINT (ours)	36.736	2.12	0.461	0.000	256.0	1.92	1.00	100.6
Diagonal + FLINT + EGAM (ours)	26.705	4.35	0.865	0.472	131,584.0	2.01	5.47	91.6
+ 2-term APoT (ours)	**26.709**	**3.25**	**0.469**	1.262	131,584.0	2.04	5.30	91.6
DeltaNet	30.726	6.64	1.445	0.000	65,664.0	2.25	5.57	107.9
Gated DeltaNet-2	29.325	10.50	2.296	0.000	65,664.0	2.01	6.03	99.4
Gated DeltaProduct ($k = 2$)	29.876	11.13	2.433	0.000	65,664.0	2.07	5.50	104.9
Mamba2	26.765	10.50	2.310	0.000	131,200.0	1.62	6.00	108.9
xLSTM adaptation	26.562	10.69	2.372	0.000	230,659.5	1.71	5.73	55.5
Attention	28.823	8.15	1.771	0.000	1,048,576.0	2.52	4.00	108.6

E_{arith}: estimated mixer arithmetic subtotal (μJ/token/layer; Horowitz, 2014). Cost model: Appendix A. Mixer budgets are per token and layer at $L = 1024$.

Joint keyed state tracking. Group tokens update a running product; STORE_k binds it to key k, and QUERY_k requests that saved product after further updates. The mixer-only reference tracks at 0.9999 but answers queries at 0.211; the store-only reference binds correctly but tracks at 0.090.

With store-once keys, learning both gates gives 0.998–1.000 query accuracy at training length and 0.969–0.992 at $L = 512$ (Table 3). The tracking gate recalls all group events while firing on 2.8% of STORE/QUERY tokens, without position labels. Rebinding separately tests the additive mixture of repeated writes.

5.4 LANGUAGE-MODEL MEMORY, ARITHMETIC, AND QUALITY

Eight-layer WikiText-103 models (Merity et al., 2017) use width 512 and 50.71–50.72M parameters, matched by adjusting GELU MLP width. Every arm sees approximately 236M tokens in the same order at $L = 1024$. FLINT uses $d_{\text{state}} = 128$, $k = 2$; the final EGAM configuration follows Section 3.2.

Recall at nearly the tracking-only MAC budget. Two-term memory projections reduce combined MACs from 0.865M to 0.469M, only 7.6k above Diagonal + FLINT (Table 4). At this budget, the parameter-matched combined APoT model lowers PPL from 36.736 for Diagonal + FLINT to 26.709. Equation (17) gives the corresponding 1.262M AC/bit operations. The 131584 task-state elements per layer are independent of L (Eq. (18)), approximately $8\times$ fewer than attention's KV cache at $L = 1024$.

Each combined checkpoint is evaluated before and after approximation on identical batches. Mean SWDE/SQuAD extraction accuracy changes from 2.01/5.47% to 2.04/5.30% (Table 4). Appendix C.4 specifies the 40-batch PPL evaluation and extraction protocol adapted from Arora et al. (2024b). Error bars in Figure 3a show three-seed sample SD for every model. Reference mixers are FP.

Quality and arithmetic energy. At lower PPL, combined APoT uses 51% less estimated mixer arithmetic energy than float DeltaNet. Its FP-to-APoT shift reduces this subtotal by 25% (Figure 3a), using the three-seed combined ledger in Appendix A.

Quality across seeds. Combined PPL is 26.705 ± 0.081 before APoT and 26.709 ± 0.077 after it across three seeds (mean $\pm$ sample SD); the largest paired change is below 0.0077.

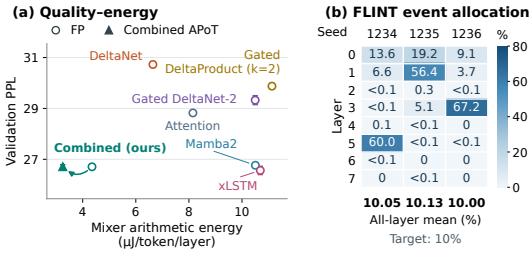

Figure 3: **WikiText-103: quality–energy trade-offs and learned event allocation.**

Table 5: **FineWeb-Edu quality and mixer efficiency at 1.007B tokens.** Matched 122.54M models; one seed. Post-training APoT shares FP Train.

	Quality–energy		Analytic mixer budget			Training
Mixer configuration	PPL ↓	E_{arith} ↓ (μJ)	MAC (M)	AC (M)	Task state (elements)	Train (k tok/s)
Diagonal + FLINT + EGAM (ours)	29.328	9.17	1.838	0.872	197,248.0	51.1
+ 2-term APoT (ours)	**29.355**	**6.96**	**1.047**	2.451	197,248.0	51.1
Gated DeltaNet-2	30.431	21.89	4.780	0.000	98,496.0	53.5
Mamba2	30.698	21.89	4.798	0.000	196,800.0	61.1
Attention	31.213	16.52	3.591	0.000	1,572,864.0	60.1

E_{arith}: estimated mixer arithmetic subtotal (μJ/token/layer; Horowitz, 2014). Cost model: Appendix A. Mixer budgets are per token and layer at $L = 1024$.

Learned allocation across layers. Over the same 40 validation batches per seed, combined FP global FLINT event rates are 10.05%, 10.13%, and 10.00% (Figure 3b): a shared aggregate budget with seed-dependent layer allocation, as Eq. (11) constrains the global mean. Across seeds, the most active layer reaches an event rate of 56.4–67.2%, while the all-layer mean remains near 10%. The aggregate constraint therefore permits concentrated transition work within the shared event budget. The prediction objective can redistribute events across layers while Eq. (13)'s transition work scales with their total count.

Measured training throughput. Table 4's floating-point training throughput includes the MLP, gate learning, and normalization. APoT quality uses approximated weights evaluated in floating point; its operation counts are analytic budgets.

Recurrent comparisons at similar perplexity. Shared-backbone Mamba2 (Dao & Gu, 2024) and xLSTM (Beck et al., 2024) reach 26.765 and 26.562 mean WikiText PPL. At 26.709 PPL, combined APoT uses 69.1% and 69.6% less estimated mixer arithmetic energy, respectively (Table 4). Combined APoT and FP Mamba2 use nearly identical task-state budgets (131584 versus 131200 elements per layer). At comparable perplexity, combined APoT requires 79.7% fewer mixer MACs within this storage budget. Appendix C.4 gives configurations and seed variability; Appendix E.1 reports paired approximation controls. Separate supervised SQuAD EM/F1 results are in Appendix E.2.

5.5 SCALING ARITHMETIC CONVERSION TO FINEWEB-EDU

We train twelve-layer, width-768 models (approximately 122.54M parameters) for 1.007B tokens on a fixed FineWeb-Edu subset (Penedo et al., 2024). Table 5 compares one seed per model on identical final-budget validation batches; Appendix C.5 specifies data and configurations. Combined FP-to-APoT PPL changes from 29.328 to 29.355 while mixer MACs decrease by 43.0% and estimated arithmetic energy from 9.17 to 6.96 μJ/token/layer. This extends paired quality retention to a second corpus and larger model, with 197248 task-state elements per layer independent of length.

6 CONCLUSION

FLINT and EGAM separately allocate transition work and associative storage. Per-token diagonal carry and gated Householder products admit an exact event-boundary schedule for precomputed gates; binary-addressed recall adds fixed-size storage with shift-add projections. Known-mask execution and joint tasks evaluate the associated timing and quality. Across three WikiText seeds, arithmetic conversion retains perplexity while reducing mixer MACs from 0.865M to 0.469M per token and layer, with 51% lower estimated mixer arithmetic energy than float DeltaNet at lower perplexity. At 122.54M parameters on FineWeb-Edu, paired quality retention accompanies a 43.0% MAC reduction. These results support complementary control of transition work, storage, and arithmetic across the two evaluated model scales and corpora.

REPRODUCIBILITY STATEMENT

Sections 3–4 define the branch updates, gate-training objective, execution algorithm, and resource budgets. Appendix B derives the segment and memory schedules and reflection budget. Appendix C supplies gate settings, training and evaluation details, and reference configurations. Appendix D provides supporting design analyses. Appendix A details the projection approximation, operation accounting, and arithmetic-energy model.

AI USE STATEMENT

In this work, we used generative AI tools solely to assist with translation and improve the readability of the manuscript. Generative AI tools were not used to conduct the research itself.

REFERENCES

Arshia Afzal, Aviv Bick, Eric P. Xing, Volkan Cevher, and Albert Gu. Raven: High-recall sequence modeling with sparse memory routing. *arXiv preprint arXiv:2607.25357*, 2026.

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 *International Conference on Learning Representations (ICLR)*, 2024a.

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 *International Conference on Machine Learning (ICML)*, pp. 1763–1840, 2024b.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Maximilian Beck, Korbinian Pöppel, Phillip Lippe, and Sepp Hochreiter. Tiled flash linear attention: More efficient linear RNN and xLSTM kernels. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Víctor Campos, Brendan Jou, Xavier Giró-i-Nieto, Jordi Torres, and Shih-Fu Chang. Skip RNN: Learning to skip state updates in recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations (ICLR)*, 2021.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, pp. 10041–10071, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

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 *International Conference on Learning Representations (ICLR)*, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR)*, 2022a.

Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, volume 35, 2022b.

Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. From one tree to a forest: A unified solution for structured web data extraction. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 775–784, 2011.

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

Mark Horowitz. Computing's energy problem (and what we can do about it). In *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 10–14, 2014. doi: 10.1109/ISSCC.2014.6757323.

Pentti Kanerva. *Sparse Distributed Memory*. MIT Press, 1988.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, pp. 5156–5165, 2020.

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 *International Conference on Learning Representations (ICLR)*, 2026.

Si Yong Lee, Ryangjin Lee, Hawon Park, Yoora Kim, Byungkon Kang, and Yoon Seok Yang. Spike-Mamba: Spike-driven state space models for energy-efficient biomedical sequence modeling. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2026*, pp. 565–575, 2026. doi: 10.1007/978-3-032-38260-3_54.

Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 6, 2024.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations (ICLR)*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *International Conference on Machine Learning (ICML)*, pp. 35492–35506, 2024.

William Merrill, Hongjian Jiang, Yanhong Li, Anthony Lin, and Ashish Sabharwal. Why are linear RNNs more parallelizable? In *International Conference on Machine Learning (ICML)*, 2026.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning (ICML)*, pp. 26670–26698, 2023.

Gunho Park, Baeseong Park, Minsub Kim, Sungjae Lee, Jeonghoon Kim, Beomseok Kwon, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, and Dongsoo Lee. LUT-GEMM: Quantized matrix multiplication based on LUTs for efficient inference in large-scale generative language models. In *International Conference on Learning Representations (ICLR)*, 2024.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

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

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2021.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning (ICML)*, pp. 9355–9366, 2021.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. DeltaProduct: Improving state-tracking in linear RNNs via Householder products. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. In *International Conference on Machine Learning (ICML)*, pp. 57503–57522, 2025.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with Hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning (ICML)*, pp. 48630–48656, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, pp. 38087–38099, 2023.

Songlin Yang and Yu Zhang. FLA: A Triton-based library for hardware-efficient implementations of linear attention mechanism. GitHub repository, fla-org/flash-linear-attention, version 0.5.2, 2024.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning (ICML)*, pp. 56501–56523, 2024a.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations (ICLR)*, 2025a.

Songlin Yang, Yikang Shen, Kaiyue Wen, Shawn Tan, Mayank Mishra, Liliang Ren, Rameswar Panda, and Yoon Kim. PaTH attention: Position encoding via accumulating Householder transformations. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Haoran You, Xiaohan Chen, Yongan Zhang, Chaojian Li, Sicheng Li, Zihao Liu, Zhangyang Wang, and Yingyan Lin. ShiftAddNet: A hardware-inspired deep network. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Yu Zhang, Songlin Yang, Ruijie Zhu, Yue Zhang, Leyang Cui, Yiqiao Wang, Bolun Wang, Freda Shi, Bailin Wang, Wei Bi, Peng Zhou, and Guohong Fu. Gated slot attention for efficient linear-time sequence modeling. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

APPENDIX

A ARITHMETIC BUDGETS AND ENERGY MODEL

Arithmetic-energy subtotal. For Tables 4 and 5, we use the 45 nm reference arithmetic costs of Horowitz (2014) (Fig. 1.1.9): FP32 multiplication, 3.7 pJ; FP32 addition, 0.9 pJ; and INT32 addition, 0.1 pJ. The estimated subtotal in μJ is

$$E_{\text{arith}} = 10^{-6}(4.6N_{\text{MAC}} + 3.7N_{\text{mul}} + 0.9N_{\text{FPadd}} + 0.1N_{\text{INTadd}})\,. \tag{19}$$

Here N_{MAC} counts multiply–accumulates, while N_{mul} and N_{FPadd} count standalone operations. The tables report analytic MAC budgets that include selected scalar-arithmetic terms, whereas N_{MAC} in the energy model counts multiply–accumulates only. The energy calculation uses this operation-level decomposition; it does not assign a uniform cost to the aggregate AC budget. Memory, control, nonlinear/normalization, and unpriced shift/bit costs are outside this subtotal. The estimate is an arithmetic cost model, not a measurement of total device energy. WikiText combined energy averages the three checkpoints' observed gate and slot statistics; reference mixers retain representative-checkpoint ledgers. For FineWeb-Edu, slot terms use the nominal $0.05 \times 256 = 12.8$ active slots per address and full write admission; these are budget assumptions, not measured slot occupancy. Its FLINT energy uses the selected checkpoint's mean event rate (0.09975).

WikiText combined APoT decomposition. The following operation-type ledger averages the three adopted checkpoints and reconstructs the 3.25 μJ entry in Table 4, per token and layer. Counts use observed gate/slot statistics; the table's nominal MAC/AC budgets are reported separately.

Operation	Mean count	Cost (pJ)	Contribution (μJ)
FP32 MAC	460, 467.491	4.6	2.118150
Standalone FP32 multiply	12, 917.965	3.7	0.047796
Standalone FP32 add	1, 201, 677.489	0.9	1.081510
INT32 add	1, 042.978	0.1	0.000104
Total			3.247561

The four approximated input projections contribute 790528 FP32 additions within the FP-add row, priced at 0.9 pJ each. Shared-scale multiplications remain in the standalone-multiply row. The exclusions stated above also apply to this decomposition.

Two-term projection approximation. The post-training approximation in Eq. (16) uses a per-tensor scale s for the four paths in Section 4.3. Greedy residual decomposition quantizes w/s with exponents in $[-8, 0]$. The shift-add budget assumes suitable operand representations and scale folding into thresholds and normalization.

Relation to efficient arithmetic. ShiftAddNet replaces multiplications with shifts and additions (You et al., 2020), while LUT-GEMM uses lookup tables for quantized matrix multiplication (Park et al., 2024). Post-training quantization methods address complementary sources of error: GPTQ uses approximate second-order information (Frantar et al., 2023), AWQ protects salient weights using activation statistics (Lin et al., 2024), and SmoothQuant redistributes activation outliers into weights (Xiao et al., 2023). QuIP# uses incoherence processing and lattice codebooks (Tseng et al., 2024); SpinQuant learns rotations to improve quantization (Liu et al., 2025). Our evaluated approximation targets the memory's four input projection paths; the rest of the block retains its trained representation.

Memory projection and residual arithmetic. At width 512, the four paths respectively contribute $1024 + 131072 + 262144 + 1024 = 395264$ MACs per token. Two-term evaluation adds at most $2(395264)$ shift-add operations to the 471859.2 AC budget, totaling 1262387.2. Read variance, the gate's energy feature, and per-slot scaling retain $512 + 512 + 0.05 \times 256 \times 512 = 7577.6$ MAC budget units, using the configured 5% address activation. The FP convention additionally counts $2d_v$ channelwise read-scaling terms; the shift-add budget folds these into the binary threshold. Thus $R_{\text{FP}} = R_2 + 2d_v$ in Eq. (17). AC groups binary-selected additions, XNOR/popcount, and shift-add/subtract.

WikiText Mamba2 and xLSTM ledgers. Including the common PDiag, Mamba2 contributes 2176000 MACs, 133264 standalone multiplications, and 1152 additions per token and layer. For xLSTM, these counts are $(2335744, 268416, 2192)$ in each mLSTM layer and $(722944, 2688, 4736)$ in the sLSTM layer. Table 4 averages the seven mLSTM layers and one sLSTM layer. The mLSTM ledger reuses the gated key $u_t = i_t k_t$ in both the matrix and normalizer updates, avoiding a redundant matrix-wide multiply. Mamba2 task state is $128 + 1024(128) = 131200$ elements; xLSTM includes matrix, normalizer, and stabilizer states, giving $[7(263300) + 2176]/8 = 230659.5$ elements per layer on average. The nonlinear/normalization exclusion above includes Mamba2's gated RMSNorm.

Scaling the ledger. At width 768, the four memory projections contribute $1536 + 196608 + 589824 + 1536 = 789504$ MACs per token. The nominal residual is $768 + 768 + 12.8 \times 768 = 11366.4$; the two-term AC budget is $871628.8 + 2(789504) = 2450636.8$. All-token projections are retained in the FLINT budget. The FineWeb Gated DeltaNet-2 ledger uses the checkpoint's six 128×128 matrix states, 3932160 projection MACs, and 9216 convolution MACs. Its recurrence contributes $3(98304)$ MACs, $98304 + 2(768)$ scalar multiplications, and 768 additions. Mamba2 uses 3753984 projection and 7168 convolution MACs; its 1536×128 SSM state contributes two contractions (393216 MACs), decay/input/skip multiplications, and skip additions. The common PDiag projections contribute $3(768)(192)$ MACs. Attention uses the causal mean prefix $(1024 + 1)/2$ in both tables.

Storage and whole-model scope. Task state counts the recurrent vector, matrix, or value/count bank; the attention entry is its KV cache. Auxiliary shadow/convolution/gate histories and the immutable address bank are separate from these entries. For the large combined model, $192 + 192 + 256(768 + 1) = 197248$ task-state elements occupy 770.5 KiB at FP32. Large Gated DeltaNet-2, Mamba2, and attention have 384.75, 768.75, and 6144 KiB under the same task-storage convention. These are logical payloads, not peak training allocations. Including all twelve MLPs and the tied vocabulary projection changes the principal arithmetic subtotal to $518.87\,\mu$J/token for combined APoT, versus 583.98, 593.73, and 606.69 for FP Gated DeltaNet-2, Mamba2, and attention. The same unpriced operations remain outside these whole-model subtotals.

Operand-level implementation check. A CPU reference validates dyadic exponent adjustment and FP32 accumulation, packed 64-bit XNOR/popcount addresses, shift-subtract leak, and binary-input output accumulation. Operands and accumulators are FP32; address counts are integer. Per-projection scales and floating-point normalization are retained, with exceptional dyadic operands handled by a fallback. Across eight layers and eight fixed sequences ($L = 128/1024$), all 64 branch cells have identical binary decisions, with no exceptional-operand fallbacks. The largest connected absolute NLL difference is 5.79×10^{-5}. This validates an arithmetic path, separately from GPU latency or device-energy measurement.

B Derivations and execution semantics

B.1 Segment recurrence

Fix the input features, transition parameters, and hard gate sequence. Using the notation of Section 4.1, assume $s_{j-1} = h_{e_{j-1}}$. Multiplying the intervening diagonal factors in Eq. (4) gives $\tilde{h}_{e_j} = q_j \odot s_{j-1}$; applying H_{e_j} therefore gives $s_j = h_{e_j}$. Induction from $s_0 = h_0$ proves equality at every event. The same diagonal-product identity proves Eq. (12) between events, including the event-free case $E = 0$.

The equivalence is algebraic. Signed factors can be tracked through sign parity and magnitude products; implementations using logarithms must handle zeros and finite-precision underflow consistently with the reference. For fixed gate decisions, the schedules also define the same derivatives with respect to transition parameters in exact arithmetic. Gate-surrogate derivatives additionally require Eq. (14), including when the hard $g_t = 0$; a fixed-mask recurrence alone does not provide those counterfactual transformations.

B.2 Parallel and recurrent memory schedules

Take $M_{0,m} = 0$ and $C_{0,m} = 0$. Expanding Eq. (7) gives

$$M_{t,m} = \sum_{s=1}^{t} \lambda^{t-s} g_s^w A_{s,m}^w v_s, \tag{20}$$

$$C_{t,m} = \sum_{s=1}^{t} \lambda^{t-s} g_s^w A_{s,m}^w. \tag{21}$$

For an unnormalized sum read, substitution yields

$$r_t = \sum_{s=1}^{t} \lambda^{t-s} g_s^w \langle A_t^r, A_s^w \rangle v_s.$$

In matrix notation this is $R = SV$ with $S = (A_r A_w^\top) \odot B$ and $B_{t,s} = \mathbf{1}[s \le t]\lambda^{t-s} g_s^w$.

For the per-slot read, substitute the expanded $M_{t,m}$ into Eq. (8) and interchange the finite sums over m and s. The coefficient of v_s is exactly $\omega_{t,s}/\max(N_t, 1)$ from Eq. (15). The count denominator and occupied-slot indicator retain their query-time index t in this substitution; replacing them with write-time quantities would change the read.

B.3 Reflection rank and group reference values

Why each factor contributes at most one direction. For matrices A and B of the same size, $AB - I = A(B - I) + (A - I)$, hence

$$\operatorname{rank}(AB - I) \le \operatorname{rank}(B - I) + \operatorname{rank}(A - I).$$

Each generalized Householder factor differs from identity by a matrix of rank at most one. Induction proves $\operatorname{rank}(H_t - I) \le k$, including degenerate factors and singular products. Since $T_t - D_t = g_t(H_t - I)D_t$, multiplication by D_t cannot increase rank, and $g_t = 0$ makes the correction zero. This proves both bounds in Eq. (5).

For Eq. (6), write $z = (v^\top z)v + z_\perp$. The factor maps this to $(1 - \beta)(v^\top z)v + z_\perp$. Orthogonality gives the stated squared-norm identity, and composing nonexpansive factors with $\|D_t\|_2 \le 1$ proves the state bound.

When the group calculation applies. Assume the event operator for every $g \in G$ is itself a product of at most k factors, and these event operators realize a faithful real representation π of G. Applying Eq. (5) to every element yields

$$k \ge b(G), \qquad b(G) = \min_{\pi \text{ faithful}} \max_{g \in G} \operatorname{rank}(\pi(g) - I). \tag{22}$$

A finite-group representation can be made orthogonal by averaging an inner product over the group. For an orthogonal matrix, the minimum number of full reflections is its fixed-space codimension. Full reflections ($\beta = 2$) and identity factors ($\beta = 0$) are allowed in this idealized reference; the sigmoid parameterization approaches these endpoints.

S_3 and A_5. The standard two-dimensional representation of S_3 is faithful, and a three-cycle has fixed-space codimension two. The faithful icosahedral representation of A_5 lies in $\mathrm{SO}(3)$, where every non-identity rotation has fixed-space codimension two. For the lower bound in both cases, a real finite-order matrix with $\operatorname{rank}(M - I) \le 1$ can have only one non-unit eigenvalue, which must be real. It therefore cannot represent a nontrivial element of odd order. The presence of a three-cycle forces a maximum codimension of at least two. Thus $b(S_3) = b(A_5) = 2$.

S_5. The real irreducible representations have dimensions $1, 1, 4, 4, 5, 5, 6$. A five-cycle acts trivially on the two one-dimensional representations. On the standard four-dimensional representation, its eigenvalues are the four nontrivial fifth roots of unity; the sign twist has the same action. For either five-dimensional irreducible, the character is zero at all four non-identity powers of the cycle,

giving each fifth root multiplicity one. For the six-dimensional irreducible the character is one at those powers, giving multiplicity two for eigenvalue one and one for each other fifth root. Every nontrivial such summand therefore contributes rank four to $\pi(g) - I$. A faithful representation must contain a summand on which the five-cycle acts nontrivially. The standard representation attains the resulting lower bound, so $b(S_5) = 4$.

Relation to the trained network. The diagonal factor in $H\,\mathrm{diag}(a)$ may itself differ from identity in many coordinates, and a learned classifier need not identify its state with a faithful linear group representation.

C Training and evaluation details

C.1 Synthetic training

For the state-tracking tasks in Section 5.2, the learned-gate experiments use learning rates 0.001 and 0.003 with multiple seeds. The default budget is 15k steps, extended to 30k for the $\rho = 0.02$ and 0.05 density cells. The locked schedule maintains the learning rate until training accuracy exceeds 0.9 in two consecutive readings, then reduces it by a factor of ten. Transition flags and schedule triggers are distinct recorded quantities; the published-reference battery uses a 0.6 training accuracy crossing for its transition flag.

C.2 Gate parameterization

For the LIF gate, the pre-reset accumulator and its update are

$$c_t^- = \alpha c_{t-1} + \mathrm{softplus}(z_t), \qquad g_t = \mathbf{1}[c_t^- > \theta], \qquad c_t = c_t^- - \theta g_t.$$

The straight-through expression is evaluated as $g_{\mathrm{hard}} + (g_{\mathrm{soft}} - \mathrm{stopgrad}(g_{\mathrm{soft}}))$, so the cancelling soft terms are evaluated first.

The synthetic rate penalty is $\eta(\bar{g} - 1.2\rho)^2$. The reflection-strength head is initialized near full reflections.

C.3 Reference implementations and measurement environment

Synthetic published references use the authors' implementations in flash-linear-attention version $0.5.2$ (Yang & Zhang, 2024): DeltaNet, Gated DeltaNet, Gated DeltaNet-2, and Gated DeltaProduct. They use two heads of width 64, negative-eigenvalue support, and no short convolution. The initial budget is 3k steps with cosine decay; extended budgets are 15k and 60k with transition-triggered decay. Sparse $\rho = 0.1$ and dense $\rho = 1$ are evaluated.

In the oracle experiment, FLINT reaches $0.999/0.994$ position/final accuracy at 3k steps. Gated DeltaProduct with $k = 4$ reaches $0.9996/0.997$ in one of two extended-budget runs, first crossing at approximately 17k steps within a 60k budget.

The always-dense vector Householder architectural reference reaches $0.160/0.010$ position/final accuracy at 3k steps and $0.171/0.012$ at 15k on sparse S_5. Its transition differs from the combined diagonal-plus-reflection recurrence, so this comparison does not isolate a change to the gate alone. The three published rank-one delta-rule models record no transitions in their 48 sparse/dense and short/extended-budget cells. At $L = 512$, the successful Gated DeltaProduct $k = 4$ run reaches $0.819/0.405$ position/final accuracy.

Hardware and software. Experiments use a single NVIDIA RTX PRO 6000 Blackwell Workstation Edition GPU with 96 GB memory. The recorded environments are:

Component	Configuration
GPU driver / CUDA	575.51.02 / 12.8
Python / PyTorch	3.12.12 / 2.11.0.dev20251223+cu128
Triton / host compiler	3.6.0+git9844da95 / GCC 12.3.0
Reference kernels	flash-linear-attention 0.5.2 (Yang & Zhang, 2024)
Transformers	4.57.3
NumPy	2.4.1 in the custom-model environment; 2.3.5 in the FLA environment

Both environments use the same PyTorch and Triton versions. Language-model training uses bf16 autocast; the fused FLINT state recurrence accumulates in FP32. The fast FLINT path combines a fused event loop with compiled counterfactual and glue operations. All EGAM forward functions use torch.compile(dynamic=False): eight in the WikiText model and twelve in the FineWeb-Edu model.

Timing protocols. Table 1 measures randomly initialized blocks on synthetic inputs, with a shared state-64 PDiag before each mixer and pre-norm residual updates around every branch. All rows use the same FP32 affine PDiag scan, compiled normalization and residual updates, and an outer torch.compile(dynamic=False) wrapper for inference. The same known $\rho = 0.1$ masks drive FLINT events and EGAM writes. FLINT uses fused token-axis kernels; combined additionally uses a chunked EGAM slot recurrence with 256 slots and key/value width 128. Its task state is $64 + 64 + 256(128 + 1) = 33152$ elements; the Delta blocks use $64 + 2 \cdot 64^2 = 8256$. Combined inference uses packed XOR/popcount addressing, 32-token slot-count scans, and 128-token chunks for the memory-value recurrence. Gradient-enabled execution uses the differentiable chunked implementation. Two warmup calls precede twelve synchronized groups of ten calls; cyclic and reversed model orders balance each model's position. We report the median group mean, with no new compilation during timing. Forward–backward includes the output-squared-mean loss and backpropagation, without an optimizer step or learned-gate counterfactuals. These block timings use optimized scan/chunk kernels; language-model results in Tables 4 and 5 use the original PDiag implementation and dense parallel EGAM read. The sequential/segment comparison uses two warmup calls followed by five timed calls, or three at $L = 8192$; CUDA is synchronized before and after each timed group, and elapsed time is divided by the number of calls. For the Diagonal + FLINT and combined language models, Table 4 reports median throughput over 50-step float-training windows, with CUDA synchronization at the logging boundary. Periodic validation and checkpointing are followed by a timer reset. These full-model windows include forward, backward, and optimizer work. The combined median excludes the first 1200 optimizer steps. For repeated configurations, the table averages the per-run medians. For Table 5, all models use the median of logged, synchronized 50-step windows after step 1200 through step 61400. The first window after each 1200-step boundary is excluded to remove checkpoint/restart effects. Timer resets exclude periodic validation. The isolated schedule control in Appendix E.3 instead uses ten warmups and nine alternating, synchronized groups of ten calls, repeated in three processes.

Optimized-execution validation. Applying the optimized kernels to the three WikiText checkpoints gives mean FP/APoT perplexity $26.703/26.713$ on the same 40 validation batches. Each seed and representation changes by less than 0.05% relative to the chunked implementation before the packed-address/count and outer-compilation changes. The learned shadow-gate computation retains its original compilation boundary in this check. On synthetic block inputs, the compiled bf16 outputs have relative L_2 difference 0.0848 from that implementation; this validation concerns perplexity retention, distinct from the FLINT parity test.

Arithmetic counts and runtime capture different aspects of efficiency. FlashAttention and FlashAttention-2 optimize memory traffic and work partitioning for exact attention (Dao et al., 2022; Dao, 2024); Tiled Flash Linear Attention improves linear-RNN kernels through tiling (Beck et al., 2025). These results motivate reporting measured throughput alongside the analytic mixer budgets, rather than inferring runtime from operation counts alone.

C.4 LANGUAGE-MODEL CONFIGURATION, TRAINING AND EVALUATION

Configuration. The Diagonal + FLINT and combined configurations in Table 4 use the following settings. Shared embeddings use the GPT-2 vocabulary (Radford et al., 2019); parameter matching adjusts the GELU MLP width.

Setting	Value
Layers / residual width / vocabulary	8 / 512 / 50,257
Sequence length / batch size	1,024 / 16
Optimizer steps / training tokens	14,400 / 235,929,600
Gradient accumulation	One batch per optimizer step
MLP hidden width	2,594 (Diagonal + FLINT); 2,016 (combined)
Trainable parameters	50,716,968 (Diagonal + FLINT); 50,717,488 (combined)
PDiag state / FLINT state	128 / 128 per layer, where present
FLINT reflection budget / gate	$k = 2$ / shadow-state threshold
EGAM slots / key bits / value width	256 / 128 / 512; one head
Address-threshold calibration	Nominal activation fraction 0.05
Memory leak	$255/256$
Optimizer	AdamW; weight decay 0.1
Schedule / gradient clipping	300-step warmup, cosine decay / global norm 1

WikiText Delta-family references use four heads of width 128, negative-eigenvalue support, and no short convolution. Mamba2 uses expansion two, state width 128, head dimension 64, and a four-tap convolution. The xLSTM adaptation uses seven mLSTM layers and one sLSTM layer at the second block, preserving the published internal mixers in the shared backbone. Their GELU MLP widths are 1175 and 1408, with $50{,}719{,}288$ and $50{,}720{,}824$ trainable parameters, respectively. MLP projection costs per token and layer are approximately 2.7M MACs for Diagonal + FLINT, 2.1M for the combined block, and 1.9M for attention; they are outside the mixer budgets.

Training objective. Equation (11) uses $\eta_F = 5$ and $\eta_W = 1$, with event target $\rho_F = 0.1$ and write floor $\rho_W = 0.05$. Terms are omitted when their branch is absent. The write penalty is normalized by its target and applied separately to each memory layer. These configurations use AdamW (Loshchilov & Hutter, 2019) and the same training-token order.

Evaluation. The combined recipe was selected on validation perplexity and then held fixed for the repeated runs. Each seed is evaluated after the full training budget. Perplexity is the exponential of mean next-token cross-entropy over 40 fixed validation batches of shape 16×1024, using data-offset seeds $777000 + i$ for $i = 0, \ldots, 39$. Recurrent states are initialized afresh for each sequence. Paired float and two-term evaluations use the same batches and compiled execution configuration; the projection approximation is specified in Appendix A.

Extraction uses the BASED-adapted validation sets (Arora et al., 2024b): all 1,111 SWDE examples (Hao et al., 2011) and the first 1,000 SQuAD examples (Rajpurkar et al., 2016). The document and prompt are left-truncated or left-padded with EOS token 50,256 to 976 tokens. Evaluation uses batches of 32 and generates 24 greedy tokens, with no early stopping at EOS. After lowercasing, stripping, and collapsing whitespace, a prediction is correct when the normalized gold answer occurs within the generated text. This is the extraction-accuracy protocol used in Table 4. Float and approximated models use identical examples.

Repeated runs. The combined model uses seeds 1234, 1235, and 1236, with the same training-token order and paired float/APoT evaluation. For every model, Table 4 averages PPL, extraction accuracy, and per-run training throughput across these three seeds under the same evaluation protocol. Mamba2 and xLSTM select the configuration with lowest final validation NLL on seed 1234 and repeat it for seeds 1235 and 1236.

Seed	Combined float PPL	Combined two-term APoT PPL
1234	26.781	26.779
1235	26.716	26.721
1236	26.619	26.627
Mean	26.705	26.709
Sample SD	0.081	0.077

DeltaNet's float PPL is 30.726 ± 0.110 and Gated DeltaNet-2's is 29.325 ± 0.173 over the three seeds. FLINT and attention give 36.736 ± 0.079 and 28.823 ± 0.005. Mamba2 and xLSTM give 26.765 ± 0.098 and 26.562 ± 0.160. Diagonal and Gated DeltaProduct ($k = 2$) give 40.739 ± 0.075 and 29.876 ± 0.108, respectively.

C.5 FineWeb-Edu configuration and data

Table 5 uses a fixed sample from the FineWeb-Edu 10BT release (Penedo et al., 2024). Shard order is shuffled once with seed 20260919. Documents are assigned by the first eight hexadecimal digits of SHA256(text), modulo 1000: residues 0–4 to validation, 5–9 to test, and the rest to training. The GPT-2 tokenizer and an EOS separator produce 1050106876 training tokens and 2001697 validation tokens. The test stream is reserved; reported PPL and configuration selection use validation. Dataset revision, shard hashes, and stream hashes accompany the configurations.

Setting	FineWeb-Edu configuration
Layers / residual width / vocabulary	12 / 768 / 50,257
Sequence / batch / accumulation	1,024 / 16 / 1
Optimizer steps / training tokens	61,440 / 1,006,632,960
Training seed	1234, all four models
PDiag / FLINT state, reflection budget	192 / 192, $k = 2$
EGAM slots / key bits / value width	256 / 128 / 768; leak $255/256$
MLP	GELU; width adjusted for parameter matching
Optimization	AdamW; 300-step warmup, cosine decay, gradient norm clip 1
Evaluation	40 fixed batches; batch 16, length 1024, offsets $777000 + i$

Mixer	Parameters	MLP width	Peak training (GiB)
Combined	122,536,392	3,040	38.11
Gated DeltaNet-2	122,536,380	1,695	35.02
Mamba2	122,533,956	1,811	29.16
Attention	122,528,676	2,723	30.05

Gated DeltaNet-2 uses six width-128 heads and four-tap short convolutions. Mamba2 uses expansion two, SSM state width 128, head dimension 64, and a four-tap convolution. These published mixers replace the mixer sublayer in the common PDiag/pre-norm/GELU backbone; the comparison is not to their canonical full model stacks. The FLINT kernel pads its logical width-192 state to 256 internally, without changing the logical state budget. Final-budget checkpoints are evaluated before and after approximation. Configurations are model-specific, and the complete selected-run configurations are included with the reproducibility artifacts; equal search effort is not assumed. FineWeb-Edu and WikiText have different corpora and budgets and are reported separately.

C.6 Implementation specification

This subsection specifies implementation details for the branches in Section 3. It uses the normalized branch input x_t and the states and address selectors defined there.

Normalization, initialization, and readout. Pre-norm and final normalization use $\mathrm{RMSNorm}(x) = \gamma \odot x/\sqrt{\mathrm{mean}(x^2) + 10^{-6}}$. For the final residual stream x, logits are $30\tanh(\mathrm{head}(\mathrm{RMSNorm}(x))/30)$, with head weights tied to the input embedding. Embeddings are initialized with standard deviation 0.02; residual-branch output weights use $0.02/\sqrt{2n_{\mathrm{layers}}}$, with zero output biases where present.

Branch execution and gate surrogates. The parallel PDiag implementation of Eq. (1) uses 32-token chunks. The FLINT readout in Section 3.1 acts directly on h_t; RMSNorm is applied at the residual sublayer input. With $p_t^F = \sigma(z_t)$, the FLINT straight-through expression is $\mathbf{1}[z_t > 0] + (p_t^F - \mathrm{sg}(p_t^F))$, where sg denotes stop-gradient.

Write-gate features and address thresholds. For the write-gate features in Section 3.3, let $p_t^w = \sigma(z_t^w)$. Training uses $g_t^w + p_t^w - \mathrm{sg}(p_t^w)$. The gate projection starts with zero weights and bias one. Key bits use threshold zero. A fixed query's overlap with a random 128-bit Bernoulli$(1/2)$ address has standard deviation $\sigma = \frac{\sqrt{128}}{2} = \sqrt{32}$. The threshold in Section 3.2 is approximately 73.3047, so integer overlap counts of at least 74 activate a slot. The hard key, address-overlap, and binary-read thresholds use an arctangent surrogate derivative

$$\varphi_s(z) = \frac{1}{s\,[1 + (\pi z/s)^2]}, \qquad s = \begin{cases} 1, & \text{key/read thresholds,} \\ \frac{\sqrt{128}}{2}, & \text{overlap thresholds.} \end{cases}$$

Here z is the pre-threshold value after subtracting the threshold.

Read normalization and gradients. The standardization in Eq. (9) acts across the d_v value channels using $(r_t - \mathrm{mean}(r_t))/(\mathrm{std}_{\mathrm{sample}}(r_t) + 10^{-5})$, where the sample standard deviation uses correction one. During parallel training, gradients pass through $\max(C_{t,m}, 1)$ where differentiable. The outer denominator $\max(N_t, 1)$ in Eq. (8) is stop-gradient. Occupancy and nonempty-match indicators use hard comparisons, and the final binary read uses φ_1 above.

Scan numerics. Segment products and the diagonal shadow are evaluated through prefix sums of $\log(\max(|a_t|, 10^{-30}))$ and integer sign parity. Hard gate indices select event positions; the fused event loop converts its coefficients and state to FP32.

D DESIGN ANALYSES

D.1 DENSITY, LENGTH, AND REFLECTION-BUDGET SWEEPS

At density 0.5, learned gate recall ranges from 0.503 to 0.752. At density 1, all four learned-LIF runs have realized firing rate 1.000, applying the transition at every token.

Table 6: **Learned-LIF density sweep.** Final-position accuracy ranges over all four learning-rate/seed cells per row; train length is 128. The 512 column evaluates the same checkpoints at four times that length.

ρ	Steps	Transitions	Final, $L = 128$	Final, $L = 512$
0.02	30k	4/4	0.960–1.000	0.558–0.886
0.05	30k	4/4	0.980–1.000	0.466–0.687
0.1	15k	4/4	0.968–0.999	0.053–0.439
0.2	15k	4/4	0.890–0.991	0.009–0.024
0.5	15k	0/4	0.053–0.373	0.006–0.011
1	15k	2/4	0.141–0.994	0.010–0.014

Training directly at length 512 produces 0.987–1.000 final accuracy at that same length for the learned-LIF cells. Across the breadth study at $\rho = 0.1$ and $k = 4$, the oracle and LIF configurations for Z_5, S_3, S_4, A_5 all transition (32/32 cells), with best position/final accuracy $1.000/1.000$ in each group.

At $k = 2$, A_5 reaches $0.991/0.962$ best position/final accuracy while S_5 reaches $0.848/0.602$; S_3 is solved at $k = 1$.

D.2 GATE STATE-FEATURE ABLATIONS

What happens when the gate fires? In successful uniform-event LIF cells, the relative displacement $\|H_t\tilde{h}_t - \tilde{h}_t\|/\|\tilde{h}_t\|$ is approximately 1 at fired true events and 0.11–0.24 at fired non-events.

Table 7: **Reflection-budget sweep.** Oracle gates, $\rho = 0.1$, training length 128. Ranges include all four cells per setting.

Group	k	Transitions	Final, $L = 128$	Final, $L = 512$
S_3	1	4/4	1.000–1.000	0.981–1.000
S_3	2	4/4	1.000–1.000	0.935–1.000
A_5	1	0/4	0.031–0.110	0.018–0.019
A_5	2	4/4	0.762–0.962	0.071–0.231
S_5	1	0/4	0.021–0.033	0.007–0.019
S_5	2	0/4	0.058–0.602	0.006–0.032

The single-factor correction $|\beta v^\top z|$ is small for near-orthogonal directions, but each firing incurs Eq. (13)'s cost.

In two uniform-event threshold checkpoints, $\|w_{gh}\|/\|w_g\|$ is 0.98 and 1.18. Removing the state-reading term leaves evaluation accuracy unchanged. Contextual replay is different: a diagonal shadow reproduces one checkpoint's gate decisions exactly, but another checkpoint depends on features altered by reflections.

D.3 Additional recall and timing results

Parameter sharing across the two functions. On the individual module tasks, the combined block stays within 0.0007 recall accuracy and one percentage point of tracking accuracy of the single-module references. A 12-cell probe on the module-specific tasks compared the paired block with alternating single-module blocks. The paired organization matched the alternating reference's reported accuracy with 27% fewer parameters in this probe. It places the two functions in successive residual branches followed by one MLP (Figure 1a). Joint experiments retain single-module controls.

Table 8: **Synthetic associative recall.** Mean query accuracy over eight seeds. State counts refer to the configured recurrent value state; the continuous reference uses the same task harness.

Configuration	State	$p = 4$	$p = 8$	$p = 16$	$p = 32$
EGAM	32,768	0.999	0.999	0.999	0.998
ShiftSSM + EGAM	33,920	1.000	1.000	0.999	0.998
Continuous reference	–	1.000	1.000	1.000	1.000

The binary recall runs have a worst recorded seed accuracy of 0.992 over the 96 evaluated runs. In the fixed-pair length study, the two-layer attention reference transitions in $3/8$ to $6/8$ seeds depending on length, while the store configurations transition in all eight seeds. These results concern this synthetic task and training configuration.

Extended mixer timings. At $\rho = 0.1$ and $k = 4$, the expected number of selected Householder factors is 0.4 per token. This counts transition applications, not interchangeable units of work across vector and matrix states. Gate decisions match exactly in the reported sequential/segment comparison. The gradient difference reported in Section 5.1 is at the recorded fused-backward reproducibility floor. Table 9 reports the original isolated-mixer timings across lengths and projection precisions. These exclude the PDiag, normalization, and residual work included in Table 1.

D.4 Checkpoint retrieval as a controlled intervention

An additional intervention studies whether associative storage can preserve a continuous composition state across long inert gaps. At known segment boundaries, an external schedule writes the trained mixer's state under a fresh key and restores it before the next active segment. The shadow state is included when needed by the gate. This uses non-leaky continuous storage and oracle checkpoint timing, separately from the learned language-model configuration. An exact-state-copy control isolates retrieval error from the subsequent recurrent computation.

Table 9: **Isolated-mixer timing by length.** Batch 128, width 128, known-mask density 0.1; PDiag, normalization, and residual updates are excluded. Each triplet corresponds to $L = 128/512/2048$; times are ms.

Mixer	Forward	Forward + backward
FLINT, fp32	0.35 / 0.37 / 0.77	1.2 / 1.3 / 4.5
FLINT, bf16 projections	0.44 / 0.45 / 0.76	1.3 / 1.3 / 3.2
DeltaNet	0.59 / 0.56 / 2.33	1.7 / 1.9 / 6.8
Gated DeltaNet	0.68 / 0.67 / 2.39	2.0 / 2.3 / 7.3
Gated DeltaNet-2	0.82 / 0.84 / 4.73	2.4 / 3.3 / 15.1
Gated DeltaProduct ($k = 4$)	0.81 / 1.48 / 6.62	2.2 / 4.4 / 19.3

Table 10: **Checkpoint intervention.** Final-segment accuracy for shadow-state / true-state gate models. Random keys use activation fraction 0.05. The selected code has zero overlap among activated slot sets. All conditions use the same streams as their matched controls.

Gap length	Gaps	No store	Random keys	Zero overlap	Exact copy
1024	1	.862 / .916	.995 / 1.000	.995 / 1.000	.995 / 1.000
1024	2	.536 / .288	.938 / .979	.982 / .997	.982 / .997
1024	4	.094 / .026	.516 / .623	.867 / .934	.867 / .934
1024	8	.004 / .006	.009 / .026	.365 / .444	.365 / .444
4096	1	.095 / .184	1.000 / 1.000	1.000 / 1.000	1.000 / 1.000
4096	4	.013 / .006	.436 / .654	.860 / .893	.860 / .893

The single-gap retrieval matches exact copying, whereas increasing checkpoint count exposes overlap among random address sets (Table 10). A zero-overlap code selected from 2048 random candidates matches the copy control across the evaluated conditions. At eight gaps the copy control itself reaches $0.365/0.444$, separating residual composition errors from cross-key interference.

E ADDITIONAL VALIDATION

E.1 PAIRED PROJECTION-APPROXIMATION CONTROLS

Combined approximation targets the four memory projection paths. For the reference mixers, the same two-term exponent range is applied to their linear projections (including headwise linear projections for xLSTM), leaving recurrence and convolution arithmetic unchanged. Approximation is paired within each checkpoint and uses no additional training. Applying it to the WikiText Mamba2 and xLSTM checkpoints in Table 4 gives PPL 26.807 ± 0.106 and 26.613 ± 0.170, respectively (mean $\pm$ sample SD over three seeds). The selected FineWeb-Edu checkpoint comparisons are:

	Validation PPL		E_{arith} (μJ/token/layer)	
Configuration	FP	Two-term	FP	Two-term
Combined	29.328	29.355	9.17	6.96
Gated DeltaNet-2	30.431	30.456	21.89	10.91
Mamba2	30.698	30.735	21.89	11.39
Attention	31.213	31.223	16.52	9.93

Energy uses the identical operation-type pricing and exclusions of Appendix A, with output scales retained. These controls separate approximation tolerance from the different recurrent arithmetic that remains after the projection conversion.

E.2 SUPERVISED SQUAD EVALUATION

Each pretrained WikiText model is fine-tuned for 5000 steps with batch 16 and context 1024, using answer-token supervision. A hash-based split holds out 5% of training contexts for internal diagnostics; the final fixed-budget checkpoint is evaluated on 1000 official-validation questions selected by

question-ID SHA256 order. Prompts use context windows of at most 768 tokens with half-window stride and up to 32 greedy answer tokens. Among nonempty window predictions, the answer with highest mean generated-token log probability is selected. EM and token F1 use official SQuAD normalization and maximum over gold answers. Table 11 is a separate supervised evaluation, distinct from the zero-shot substring-containment metric in Table 4.

Table 11: **Supervised SQuAD on a fixed 1000-question subset.** Mean $\pm$ sample SD over three training seeds; scores are percentages. FP and two-term evaluation use the same fine-tuned checkpoint.

	FP		Two-term	
Configuration	EM	F1	EM	F1
Diagonal + FLINT + EGAM	13.80 ± 0.62	21.61 ± 0.24	12.53 ± 0.38	20.32 ± 0.95
Diagonal + FLINT	1.47 ± 0.21	5.30 ± 0.53	–	–
DeltaNet	19.33 ± 1.03	27.54 ± 1.42	19.33 ± 0.96	27.38 ± 1.41
Attention	9.57 ± 3.70	15.15 ± 5.27	9.30 ± 3.75	14.97 ± 5.41
Mamba2	15.37 ± 6.07	22.75 ± 6.92	15.00 ± 6.13	22.61 ± 7.27
xLSTM adaptation	19.23 ± 1.21	27.57 ± 1.00	19.30 ± 1.01	27.61 ± 0.67

There is no memory projection to approximate in the FLINT-only arm. For combined, the paired means are $13.80/21.61$ EM/F1 before conversion and $12.53/20.32$ afterwards; other architectures' results are reported under the same protocol.

E.3 IDENTICAL-RECURRENCE SCHEDULE CONTROL

We replay a trained WikiText checkpoint with the same projected coefficients, initial state, and frozen hard gates under token and event schedules. Both schedules project at every token; the event timer includes count/max synchronization, indexing, gathering, and reconstruction. The recurrence comparison therefore isolates scheduling, including its overhead. Each condition uses four fixed batches and three process repetitions, with the synchronized timing protocol in Appendix C.3.

Table 12: **Schedule control across batch size and length.** Median token/event speed ratio over 12 cells. Greater than one favors event scheduling. FP32 errors are maxima over layers and cells.

Batch	Length	FP32 recurrence speedup	State relative L_2	Output relative L_2
1	512	0.738	3.40×10^{-6}	3.02×10^{-6}
1	1024	1.218	4.20×10^{-6}	4.14×10^{-6}
1	2048	1.861	9.48×10^{-6}	8.90×10^{-6}
1	4096	2.730	9.54×10^{-6}	9.85×10^{-6}
16	512	0.683	3.03×10^{-6}	2.65×10^{-6}
16	1024	0.947	5.16×10^{-6}	4.79×10^{-6}
16	2048	1.397	7.80×10^{-6}	7.20×10^{-6}
16	4096	1.837	9.35×10^{-6}	8.48×10^{-6}

The corresponding bf16-projection/FP32-state recurrence speedups at batch 16 are 0.683, 0.972, 1.402, and 1.888 for $L = 512, 1024, 2048, 4096$. Short-sequence overhead and larger-context savings coexist. Whole-model forward replay gives median speed ratios 0.941, 1.010, 1.044, and 1.049 at these lengths. In that connected replay downstream gates are recomputed, so numerical changes can alter gate decisions; the frozen-gate numerical comparison is the isolated recurrence table, not an assertion of identical full-model outputs.

Batch event utilization. The most active layer of each combined seed has mean/max event counts $614.1/692.0$, $577.3/663.8$, and $687.9/753.5$ over 40 batches at $B = 16, L = 1024$ (layers 6, 2, and 4, counting from one). The mean per-batch utilization $\mathrm{mean}_b\, E_b / \max_b E_b$ is 0.888, 0.870, and 0.913, respectively. A global event-rate target constrains total events but does not determine padded batch work.

State and normalization diagnostics. At a same-prefix extension to $L = 8192$, the final FLINT layer's tail-state norm can become small (4.98×10^{-5}, 3.99×10^{-4}, 1.33×10^{-8} across the three combined seeds). This concerns the composition state, whose output is the affine readout in Section 3.1, rather than the full residual stream. Nonexpansiveness controls amplification but does not guarantee information retention. Separately, for an occupied-match count $N > 0$, standardizing u/N with denominator $\mathrm{std}(u/N) + \epsilon$ equals $(u - \mathrm{mean}(u))/(\mathrm{std}(u) + N\epsilon)$. This identifies the role of the outer average after standardization; the per-slot count normalization remains a separate operation.