Lattice: Learning to Efficiently Compress the Memory

Orthogonal writes into a fixed recurrent state

Paper authors: Mahdi Karami, Razvan Pascanu, Vahab Mirrokni

arXiv:2504.05646v2 (30 December 2025)
Paper discussion prepared 14 September 2026

arxiv.org/abs/2504.05646v2 | Independent reading presentation

The question is how to write into limited memory

Softmax attention	Fixed recurrent state
Keep historical keys and values.	Keep $S_t \in \mathbb{R}^{d \times m}$.
Cache grows with sequence length. Sequence attention work: $O(T^2 d)$.	$S_t = S_{t-1} + v_t k_t^\top$. Repeated writes can interfere.

Can each slot inspect its contents before accepting a write?

- Low-rank structure motivates a compact learned dictionary.
- Fixed state implies lossy compression; no lossless-cache guarantee.

Paper, Secs. 2–3, PDF pp. 2–4. Costs refer to sequence mixing, excluding projections/MLPs.

The recurrent state is a decoder learned online

$$k_t = W_k x_t \in \mathbb{R}^m, \quad v_t \in \mathbb{R}^d, \quad S_t = [s_{1,t}, \ldots, s_{m,t}] \in \mathbb{R}^{d \times m}.$$

Inner loop: adapt memory to this sequence

$$S_t = S_{t-1} - \gamma_t \nabla_S \mathcal{L}(S_{t-1}, v_t, k_t), \qquad y_t = S_t q_t, \quad q_t \in \mathbb{R}^m.$$

- Outer training learns projections and the rule's parameters.
- Inner adaptation updates S, including during inference.
- The target comes from the current token; no future label is needed.

Paper, Sec. 3.1, Eqs. (3)–(4), PDF pp. 4–5; readout, p. 8.

Three objectives share a normalized dictionary

$$\Phi = \phi(S), \qquad \phi_i = \frac{s_i}{\|s_i\|}.$$

Variant	Compression objective	Interpretation
DEC	$\frac{1}{2}\|\Phi k_t - v_t\|^2$	Decode a key/code
DOT / SIM	$-\langle \Phi k_t, v_t \rangle$	Align with the value
ENC	$\frac{1}{2}\left\|\Phi^\top v_t - k_t\right\|^2$	Encode the value

- DEC error: $e_t = \Phi k_t - v_t \in \mathbb{R}^d$.
- ENC error: $e_t = \Phi^\top v_t - k_t \in \mathbb{R}^m$.

Paper, Eqs. (5), (7), (24), PDF pp. 5, 19. Half-squared losses here make the omitted factor 2 explicit; it can be absorbed into the step size.

Normalizing a slot makes its gradient orthogonal

$$J_\phi(s_i) = \frac{1}{\|s_i\|} \underbrace{\left(I - \frac{s_i s_i^\top}{\|s_i\|^2}\right)}_{P(s_i)\text{: orthogonal projector}}$$

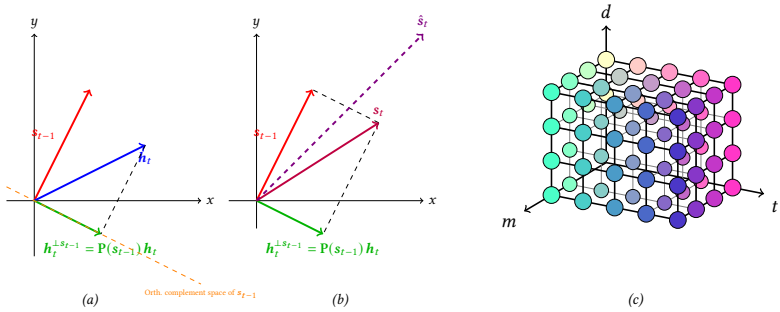
DEC: the chain rule gives a slot-specific direction

$$\nabla_{s_i} \mathcal{L}_t = \frac{k_{i,t}}{\|s_i\|} P(s_i) e_t, \qquad s_i^\top \nabla_{s_i} \mathcal{L}_t = 0.$$

- $k_{i,t}$ controls the basic write coefficient.
- $P(s_i)$ removes the component parallel to the current slot.
- State dependence comes from the projection and the error.

Paper, Sec. 3.1.1, Eqs. (5)–(9), PDF pp. 5–6. Assumes nonzero slot norms.

Orthogonality is local to each current slot



- (a) Decompose the incoming vector; (b) write its orthogonal part.
- (c) The state has m slots of width d, evolving over time t.
- This does not ensure mutually orthogonal slots or lossless recall.

Paper, Fig. 1, PDF p. 2; exact crop. The diagram uses an abstract positive write coefficient.

Renormalization stabilizes the state and changes retention

For $\|s_i\| = 1$ and $s_i^\top \Delta s_i = 0$:

$$\|s_i + \Delta s_i\|^2 = 1 + \|\Delta s_i\|^2, \qquad s_{i,t} = \underbrace{(1 + \|\Delta s_i\|^2)^{-1/2}}_{\beta_{i,t}}(s_i + \Delta s_i).$$

- Normalization *inside the loss* determines the write direction.
- Normalization *after the update* controls the state norm.
- Since $\beta \leq 1$, the old direction loses relative weight.

Geometric interpretation

A tangent-space gradient step followed by a sphere retraction.
This is not a theorem about optimal historical compression.

Paper, Sec. 3.2, Eq. (10), Proposition 3.1, PDF p. 6; proof p. 21. Formula shown without explicit decay ($\mu = 1$).

The orthogonal write costs $O(dm)$ per token

No $d \times d$ projector needs to be materialized:

$$P(s_i)h_t = h_t - s_i \frac{s_i^\top h_t}{\|s_i\|^2}.$$

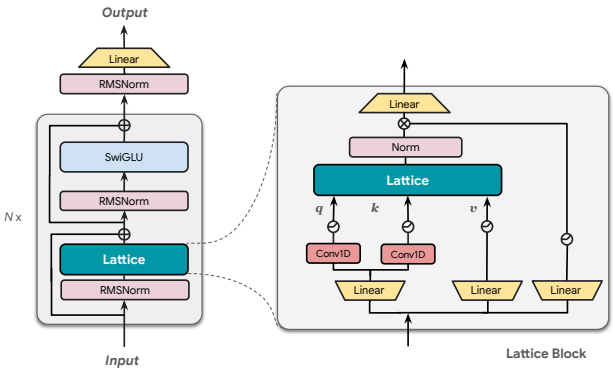
For unit slots, define $\hat{h}_t = S_{t-1}^\top h_t$ and $\hat{k}_t = \gamma_t \beta_t \odot k_t$:

$$\begin{aligned} S_t = (\mu_t \mathbf{1} \beta_t^\top) \odot S_{t-1} \\ + [\mathbf{1}(\hat{h}_t \odot \hat{k}_t)^\top] \odot S_{t-1} - h_t \hat{k}_t^\top. \end{aligned}$$

- State: $O(dm)$ per head; sequence mixing work: $O(Tdm)$.
- DEC: $h_t = \Phi k_t - v_t$; DOT: $h_t = -v_t$.
- Explicit L2 state decay gives $\mu_t = 1 - \gamma_t \lambda_t$.

Paper, Eqs. (11), (14), PDF pp. 7–8. Fixed d, m; excludes the rest of the network and training activations.

Lattice is embedded in a full language model



- Multi-head recurrent mixing.
- Short q/k convolutions, width 4.
- Output normalization and GeLU post-gate.
- Learned inner step size: $\gamma_t = \text{sigmoid}(W_\gamma x_t)$.

Paper, Fig. 4 and Appendix C, PDF p. 22; exact crop. The outer block also contains RMSNorm, SwiGLU and residual connections.

Chunk parallelism requires an approximation

Let S_0^b be the state before a chunk of C tokens.

1. Compute state-dependent quantities using S_0^b.
2. Evaluate the resulting recurrence in parallel within the chunk.
3. Pass the final state to the next chunk.

Simplified approximation: Eq. (17)

$$S_t = \hat{G}_t \odot S_{t-1} - h_t \hat{k}_t^\top, \qquad \tilde{h}_t = (S_0^b)^\top h_t,$$
$$\hat{G}_t = \mathbf{1}(\mu_t \beta_t + \tilde{h}_t \odot \hat{k}_t)^\top.$$

Larger C improves parallelism, but changes the nonlinear update.

Paper, Eqs. (15)–(17), PDF p. 8; Appendix B.1, pp. 18–19. DEC errors also use the chunk-start state. Inter-chunk dependencies remain.

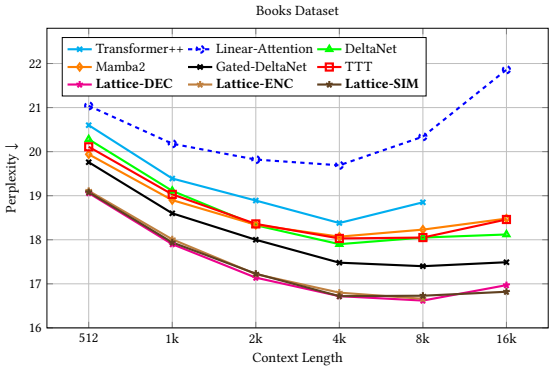
The evaluation uses different operating points

Question	Setup	Key condition
Context scaling	Books3, 110M	512–16k; $C = 1$
Model scaling	FineWeb-Edu	110M / 340M / 760M
Zero-shot tasks	340M / 760M	4k training; usually $C = 4$
Component ablation	FineWeb-Edu, 110M	Usually $C = 1$
MQAR	Two blocks, one head	$m = d$; synthetic recall
Training throughput	110M; 2×2 TPU	$C = 16$ and $C = 64$

- LM batch: 0.5M tokens. Training budgets: 5B / 15B / 30B tokens.
- Lower perplexity is better; higher accuracy/throughput is better.

Paper, Sec. 4 and Appendix C, PDF pp. 9–10, 22–26. MQAR uses its own batch sizes.

Long-context gains are not monotonic improvements



110M, Books, $C = 1$

- 8k: DEC 16.62 vs. Gated-DeltaNet 17.40.
- 16k: DOT 16.82 vs. Gated-DeltaNet 17.49.
- DEC worsens from 16.62 at 8k to 16.97 at 16k.

Each context length is a separate training setting. Transformer++ has no 16k result here.

Paper, Fig. 2 left / Fig. 5 left, PDF pp. 9, 23; exact crop. Values from the authors' plotted data in books_plot.tex.

Perplexity improves at the three tested model scales

Model	110M / 5B	340M / 15B	760M / 30B
Transformer++	16.09	11.92	9.75
Gated-DeltaNet	15.378	11.438	9.5378
TTT	15.596	11.81	9.7468
Lattice-DEC	**15.021**	**11.339**	**9.47**
Lattice-DOT	15.033	11.377	**9.47**

- FineWeb-Edu; perplexity ↓; parameters and training tokens both grow.
- The original horizontal axis is FLOPs, not parameter count.
- Largest-scale DEC gain over Gated-DeltaNet: 0.0678 ppl.

Paper, Fig. 2 right, PDF p. 9; values from Scalling_pattern.tex. FLOPs differ slightly; three points do not establish a general scaling law.

The best variant improves average zero-shot accuracy

Model	340M / 15B	760M / 30B
Transformer++	46.93	51.94
GLA	48.68	52.57
Gated-DeltaNet	48.60	52.64
TTT	47.88	52.42
Lattice-DEC, $C = 4$	**49.14**	53.54
Lattice-DOT, $C = 4$	48.66	**54.03**

- Over the best baseline: **+0.46 pp** at 340M; **+1.39 pp** at 760M.
- Gated-DeltaNet wins ARC-e; Transformer++ wins BoolQ at 760M.

Paper, Table 2, PDF p. 10. Mean accuracy (%) over seven tasks; HellaSwag/ARC-c use normalized accuracy. No confidence intervals reported.

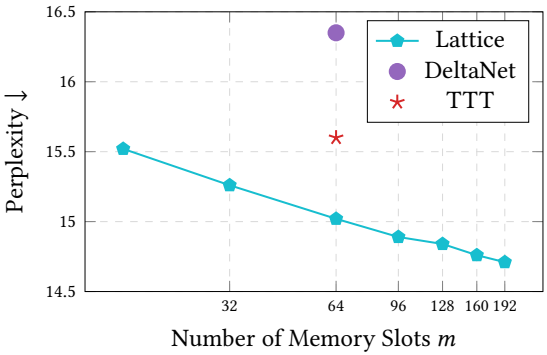
Orthogonal writes and normalization drive the ablation

Configuration	Perplexity ↓	Increment
DeltaNet	16.35	—
+ Orthogonal recurrence	15.38	−0.97
+ State projection to the sphere	15.07	−0.31
+ Explicit forget gate	**15.02**	−0.05
TTT, $C = 1$ (separate baseline)	15.60	—
Lattice, $C = 4$, Eq. (15)	15.15	—
Lattice, $C = 4$, Eq. (17)	15.12	—

The additions are cumulative; this is not a full factorial experiment.

Paper, Table 3, PDF p. 10. FineWeb-Edu, 110M, $m = d = 64$; cumulative rows use $C = 1$.

A quarter of the state can match a larger TTT memory



110M, FineWeb-Edu, $C = 1$

Lattice, $m = 16$: **15.52 ppl**.
TTT, $m = 64$: **15.60 ppl**.

For $d = 64$, state elements per head:

$$64 \times 16 = 1024, \quad 64 \times 64 = 4096.$$

- The 4× claim concerns recurrent matrix state in this setting.
- Changing m also changes key/query projection parameters.

Paper, Table 3 / Fig. 6, PDF pp. 10, 23; exact crop. Not a measurement of total training or inference memory.

Larger chunks expose normalization and quality costs

Eq. (15) approximation	Per-token norm	Per-chunk norm
$C = 4$	15.15	15.17
$C = 16$	15.32	15.36

- Both normalization placement and chunk size affect perplexity.
- Separately, Eq. (17) with $C = 16$ and *per-chunk* normalization was reported unstable.
- That instability is distinct from the finite values above.

Paper, Table 5 and accompanying text, PDF p. 24. FineWeb-Edu, 110M; perplexity ↓.

Dense associative recall is stronger at limited width

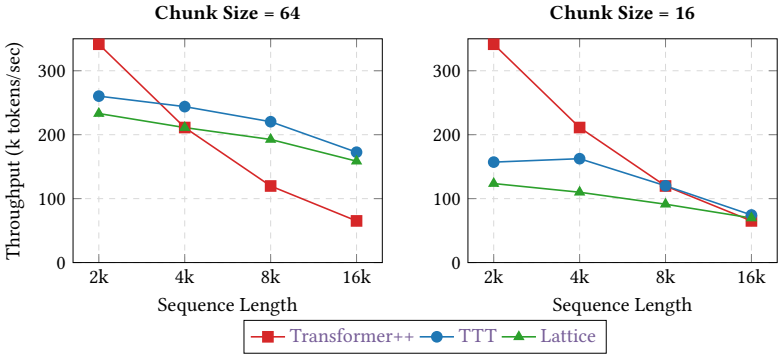
MQAR: 256 key–value pairs, $d = m = 64$, two blocks.

Model	$L = 1024$	$L = 2048$
Transformer++ (external result)	100	100
xLSTM[1:1] (external result)	73	66
TTT	27	42
DeltaNet	61	45
Lattice	**82**	**84**

- At $L = 2048$: +42 pp over TTT; +39 pp over DeltaNet.
- At $d = 128$, all three reach 100%; small-width failures still exist.

Paper, Fig. 7 and Appendix C.1, PDF pp. 24–25. Accuracy (%); plotted fractions converted to percent. Some baselines are borrowed from the xLSTM paper.

Training scales better than attention at long lengths



- 16k, $C = 64$: 158.582k tokens/s, **2.43**× Transformer++.
- Still **8.12% slower** than TTT there; attention wins at 2k.

Paper, Fig. 8, PDF p. 26; exact crop. 110M, 2×2 TPU Trillium, training throughput. TTT/Lattice use pure JAX; Transformer++ uses FlashAttention-2.

The best quality and speed are different settings

Quality evidence

Books long-context results: $C = 1$.

Most zero-shot results: $C = 4$.

Perplexity worsens as C grows in the normalization ablation.

Speed evidence

Throughput curves: $C = 16/64$.

16k speedup vs. attention:
$1.07\times$ at $C = 16$;
$2.43\times$ at $C = 64$.

A matched quality–throughput frontier is still needed.

- No measured decoding latency or peak-memory comparison here.
- Nonlinearity prevents standard inter-chunk associative scans.

Paper, Appendix C, Table 5, Fig. 8, and limitations, PDF pp. 23–26, 28. Interpretation of the reported configurations.

Reproduction needs mathematical clarifications

- **Gradient convention:** squared losses omit a factor 2 in some main-text updates; the appendix retains it.
- **Projection norm:** Eq. (23) adds squared components, but projection geometry requires a subtraction; Listing 1 subtracts.
- **Chunk normalization:** Listing 1 line 93 uses η^2 where norm preservation suggests μ^2 for the old-state term.
- **Implementation completeness:** `GLA_intra_chunk` is not defined in the listing; initialization is not fully specified.

These are reading checks, not evidence that the reported training results are invalid.

Paper, Eqs. (5)–(6), (23), (27), Listing 1, PDF pp. 5, 19, 21, 27–28. Independent derivation; downloaded author code was not executed.

The contribution is a better rule for limited-state writes

1. Slot normalization yields an interpretable, state-dependent orthogonal update.
2. Fixed-state memory is used more effectively in several reported LM and recall settings.
3. Chunk approximation makes computation practical, with a quality–parallelism trade-off.

Discussion question

At matched quality, parameters and memory, what chunk size makes Lattice preferable to a linear recurrence?

Primary source: Mahdi Karami, Razvan Pascanu, Vahab Mirrokni. *Lattice: Learning to Efficiently Compress the Memory*. arXiv:2504.05646v2, 2025. PDF and matching author TeX, including Appendices A–D, were read for this presentation.

Backup: DEC, ENC and TTT normalize different objects

Model	Predicted quantity	Orthogonal to
Lattice-DEC	$\phi(S)k$	Each slot s_i
Lattice-ENC	$\phi(S)^\top v$	Each slot s_i
TTT (paper view)	$\phi(Sk)$	Decoder output Sk

$$\text{ENC:} \quad \nabla_{s_i} \frac{1}{2} \left\| \Phi^\top v - k \right\|^2 = \frac{\phi_i^\top v - k_i}{\|s_i\|} P(s_i) v.$$

- Removing normalization from DEC recovers the delta rule.
- The TTT comparison uses the paper's L2-normalization abstraction.

Paper, Table 1; Eqs. (12)–(13), (18)–(19), (24)–(27), PDF pp. 7, 18–21.

Backup: the two chunk approximations are distinct

Let $\tilde{h}_t = (S_0^b)^\top h_t$ and $G_t = \mu_t \mathbf{1}\beta_t^\top$.

Eq. (15): freeze the state inside the write correction

$$S_t = G_t \odot S_{t-1} + [\mathbf{1}(\tilde{h}_t \odot \hat{k}_t)^\top] \odot S_0^b - h_t \hat{k}_t^\top.$$

Eq. (17): absorb the correction into a gate

$$S_t = [\mathbf{1}(\mu_t \beta_t + \tilde{h}_t \odot \hat{k}_t)^\top] \odot S_{t-1} - h_t \hat{k}_t^\top.$$

- Causal products $\Omega_{j,i,:} = a_j / a_i \, \mathbb{I}_{i \le j}$ unroll Eq. (15).
- For $C > 1$, instantaneous orthogonality and norm preservation need separate checking.

Paper, Eqs. (15)–(17), (20)–(23), PDF pp. 8, 18–19. Use tokens $bC + 1, \ldots, (b+1)C$ and state S_{bC} to avoid the printed indexing ambiguity.

Backup: the projection norm has a negative cross term

For $\|s\| = 1$, let $a = s^\top h$:

$$\|h - as\|^2 = \|h\|^2 - 2a(s^\top h) + a^2 \|s\|^2 = \|h\|^2 - a^2.$$

Independent check, not an author-reported experiment

$s = (0.6, 0.8)$, $h = (2, -1)$ gives $a = 0.4$.

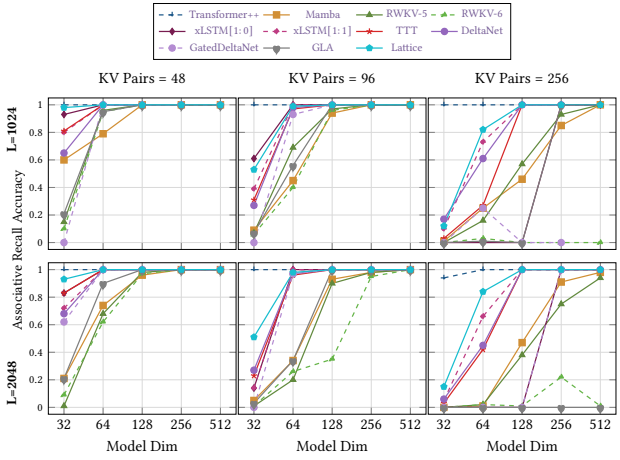
Correct projected norm squared: $5 - 0.16 = 4.84$.
The positive-sum expression gives $5 + 0.16 = 5.16$.

With explicit decay and an orthogonal increment:

$$\frac{\mu s + \Delta s}{\sqrt{\mu^2 + \|\Delta s\|^2}} \quad \text{has unit norm.}$$

Compare Paper Eq. (23), PDF p. 19, with Listing 1 lines 89–95, p. 28. Listing subtraction is consistent with geometry; its old-state denominator is a separate question.

Backup: MQAR gains depend on width and difficulty



Paper, Fig. 7, PDF p. 25; exact crop, all panels and legend. Rows: $L = 1024/2048$. Columns: 48/96/256 KV pairs. Axis accuracy is a fraction despite the percent sign in the caption.

Backup: training details and coverage

Params / tokens	Blocks	Width	Head dim	Peak LR
110M / 5B	12	768	64	10^{-2}
340M / 15B	24	1024	64	$1.5 \cdot 10^{-3}$
760M / 30B	24	1536	64	$1.25 \cdot 10^{-3}$

- LM: AdamW, 0.5M tokens/batch, 1k-step warm-up, cosine decay; final LR $3 \cdot 10^{-5}$, weight decay 0.1, gradient clip 1.0.
- MQAR: 100k training / 3k validation samples; 64 epochs; vocabulary 8192; separate learning-rate sweeps.

Paper, Table 4 and Appendix C, PDF pp. 22–25. Tokenizer, full evaluation commands and multi-seed uncertainty are not sufficiently reported for an independent reproduction.