MDN: Parallelizing Stepwise Momentum for Delta Linear Attention

Exact chunkwise computation for a two-state memory

Paper by Yulong Huang, Xiang Liu, Hongxiang Huang, Xiaopeng Lin, Zunchang Liu, Xiaowen Chu, Zeke Xie, and Bojun Cheng

The Hong Kong University of Science and Technology (Guangzhou)

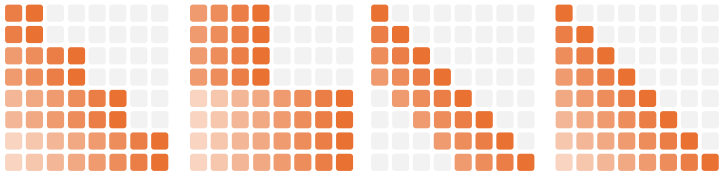
Independent paper reading presentation

arXiv:2605.05838v1 | 20 October 2026

Preserve token-by-token error feedback while training in chunks.

The gap: historical error feedback with efficient training

- Delta attention corrects a compact memory using the current error.
- Momentum keeps earlier error directions active in future updates.
- The challenge is parallelizing the resulting causal dependencies.



Paper, Sec. 1 and Fig. 1, PDF pp. 1–2. The figure compares the authors' selected update schemes.

MDN adds a momentum matrix to the delta update

$$\mathcal{L}(\widetilde{\mathbf{S}}_{t-1}) = \tfrac{1}{2}\|\boldsymbol{v}_t - \widetilde{\mathbf{S}}_{t-1}^{\top}\boldsymbol{k}_t\|_2^2, \qquad \widetilde{\mathbf{S}}_{t-1} = \alpha_t \mathbf{S}_{t-1}$$

One token, one causal memory update

$$\begin{aligned}
\widetilde{\boldsymbol{v}}_t &= \boldsymbol{v}_t - \alpha_t \mathbf{S}_{t-1}^{\top}\boldsymbol{k}_t, \\
\mathbf{M}_t &= \mu_t \mathbf{M}_{t-1} - \eta_t \boldsymbol{k}_t \widetilde{\boldsymbol{v}}_t^{\top}, \\
\mathbf{S}_t &= \alpha_t \mathbf{S}_{t-1} - \beta_t \mathbf{M}_t, \qquad \boldsymbol{o}_t = \mathbf{S}_t^{\top}\boldsymbol{q}_t.
\end{aligned}$$

- $\mathbf{S}_t, \mathbf{M}_t \in \mathbb{R}^{d_k \times d_v}$: memory and momentum.
- α: memory retention; μ: momentum retention.
- β: update step; η: current gradient scale.

Paper, Sec. 3.1, Eqs. (3)–(5), PDF p. 4. Setting $\mu = 0, \eta = 1$ recovers GDN.

One error can affect several future writes

Set $\widehat{\boldsymbol{k}}_i = \eta_i \boldsymbol{k}_i$ and condition on the residual sequence.

Contribution of the first residual to the second state

$$\underbrace{(\alpha_2\beta_1 + \beta_2\mu_2)}_{\text{two propagation paths}} \widehat{\boldsymbol{k}}_1 \widetilde{\boldsymbol{v}}_1^\top$$

Through memory

$\alpha_2\beta_1$ retains the write made at the first step.

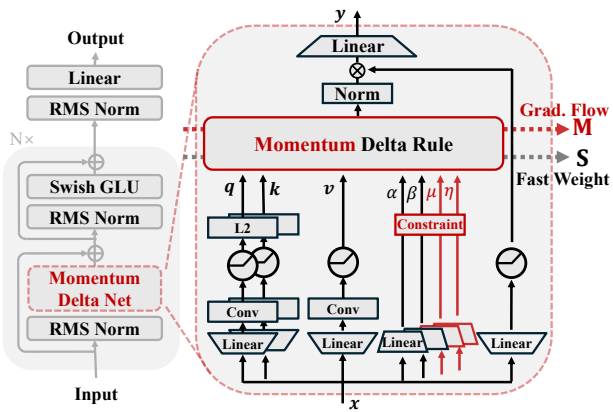
Through momentum

$\beta_2\mu_2$ writes the retained error direction again.

Actual residuals are recomputed from the preceding memory at each step.

Teaching derivation from Eqs. (4)–(5), PDF p. 4. This is inner memory optimization; outer training uses AdamW.

The memory sits inside a familiar language-model block



- Project, short-convolve, and activate q, k, v.
- Normalize queries and keys.
- Generate four scalar gates per head.
- Update memory and momentum.
- RMSNorm, output gate, projection.

Comba-style query correction is also retained.

Paper, Sec. 3.3 and Fig. 3, PDF pp. 6–7. Red components identify the momentum-related additions.

Reorder the triangular sum to isolate scalar coefficients

Within one chunk, write $\widehat{\boldsymbol{k}}_j = \eta_j \boldsymbol{k}_j$.

$$\mathbf{S}_i = \bar{\alpha}_i \mathbf{S}_0 - b_i \mathbf{M}_0 + \sum_{j \le i} \gamma_{i,j} \widehat{\boldsymbol{k}}_j \widetilde{\boldsymbol{v}}_j^\top$$

$$\bar{\alpha}_i = \prod_{r \le i} \alpha_r, \quad \bar{\mu}_i = \prod_{r \le i} \mu_r, \quad c_i = \sum_{r \le i} \frac{\beta_r \bar{\mu}_r}{\bar{\alpha}_r},$$

$$b_i = \bar{\alpha}_i c_i, \qquad \gamma_{i,j} = \frac{\bar{\alpha}_i}{\bar{\mu}_j}(c_i - c_{j-1}).$$

Why the reordering helps

Fix the error's origin j and aggregate all later writes into $\gamma_{i,j}$. The traversal order changes; the causal domain $j \le i$ is preserved.

Paper, Sec. 3.1, Eqs. (7)–(9), p. 4; Appendix C, Eqs. (30)–(31), p. 17. We use the consistent Appendix C expression.

Solve the unknown residuals with a lower triangular system

Define $\boldsymbol{p}_i = \alpha_i \boldsymbol{k}_i$ and stack token vectors as matrix rows.

$$\Gamma^-_{i,j} = \begin{cases} \gamma_{i-1,j}, & j < i, \\ 0, & j \geq i. \end{cases} \qquad \mathbf{T} = \left[\mathbf{I} + (\mathbf{P}\widehat{\mathbf{K}}^\top) \odot \mathbf{\Gamma}^-\right]^{-1}$$

$$\begin{aligned} \mathbf{U} &= \mathbf{T}\mathbf{V}, \\ \mathbf{Y} &= \mathbf{T}\,\mathrm{Diag}(\bar{\boldsymbol{\alpha}}^-)\mathbf{P}, \qquad \mathbf{Z} = \mathbf{T}\,\mathrm{Diag}(\boldsymbol{b}^-)\mathbf{P}, \\ \widetilde{\mathbf{V}} &= \mathbf{U} - \mathbf{Y}\mathbf{S}_0 + \mathbf{Z}\mathbf{M}_0. \end{aligned}$$

- $\mathbf{U}, \mathbf{Y}, \mathbf{Z}$ depend only on chunk-local inputs.
- Γ^- is Γ shifted down one row: the error reads $\mathbf{S}_{i-1}$.
- $\bar{\boldsymbol{\alpha}}^- = (1, \bar{\alpha}_1, \ldots)$; $\boldsymbol{b}^- = (0, b_1, \ldots)$.

Paper, Eqs. (15)–(18), p. 5; Appendix C, p. 18; row shift confirmed by Appendix E, p. 22.

Chunk outputs combine incoming states and local writes

$$\mathbf{O} = \underbrace{\mathrm{Diag}(\bar{\boldsymbol{\alpha}})\mathbf{Q}\mathbf{S}_0 - \mathrm{Diag}(\boldsymbol{b})\mathbf{Q}\mathbf{M}_0}_{\text{history entering the chunk}} + \underbrace{[(\mathbf{Q}\widehat{\mathbf{K}}^\top) \odot \boldsymbol{\Gamma}]\widetilde{\mathbf{V}}}_{\text{writes inside the chunk}} .$$

- Matrix operations process the chunk; two final states pass to the next chunk.
- Decoding uses the same token recurrence with fixed-size $\mathbf{S}, \mathbf{M}$.
- Chunking is an exact algebraic reformulation of the recurrence.
- At fixed head dimensions, the recurrent matrix state has twice as many elements as a single-state rule.

Paper, Sec. 3.1, PDF p. 5; Appendix C, Eqs. (40)–(42), PDF p. 18. Exactness refers to real arithmetic.

The implementation trades saved residuals for throughput

Coefficient computation

- Prefix sums of $\log \alpha, \log \mu$.
- Stable prefix log-sum-exp for c_i.
- Local $C \times C$ coefficient matrices.

Backward computation

- Save corrected values $\widetilde{\mathbf{V}}$.
- Reconstruct states for gradients.
- Avoid storing every full state pair.

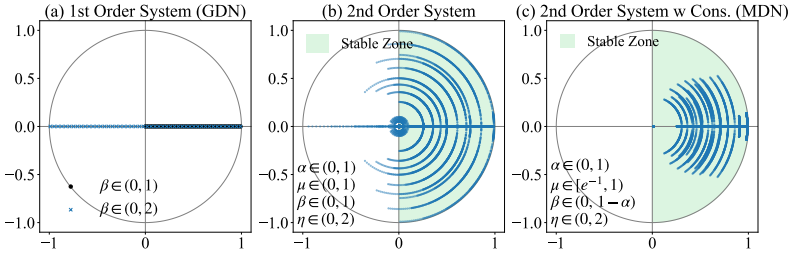
Pseudocode uses $C = 64$ by default.

Additional activations buy training speed.

Fixed chunk size gives linear scaling in sequence length. The triangular broadcast still performs $O(C^2)$ coefficient work.

Paper, Sec. 3.1, p. 5; Algorithm 1, p. 19; Appendix E, pp. 20–22. The paper's constant-time claim treats C as fixed.

Momentum permits damped oscillatory modes



$$r_t = \alpha_t + \mu_t - \alpha_t\beta_t\eta_t, \quad \lambda_\pm = \frac{r_t \pm \sqrt{r_t^2 - 4\alpha_t\mu_t}}{2}$$

- For normalized keys, a negative discriminant gives a complex pair.
- Its modulus is $\sqrt{\alpha_t\mu_t}$: decay with phase evolution.
- These are frozen-step spectra, not a bound on all time-varying products.

Paper, Fig. 2, p. 6; Appendix F, Eq. (60), p. 23. r_t denotes the trace to avoid overloading temperature τ.

Coupled gates restrict the spectral region

$$\eta_t = 1 + \tanh(\mathbf{W}_\eta \boldsymbol{x}_t/\tau), \qquad \theta_t = \arctan(s\eta_t)$$

$$\alpha_{\max} = \cos^2\theta_t, \qquad \beta_{\max} = \sin^2\theta_t \quad \Longrightarrow \quad \alpha_t + \beta_t \le 1$$

$$\log\alpha_t = f(\mathbf{W}_\alpha \boldsymbol{x}_t) + \log\alpha_{\max},$$

$$\beta_t = \sigma(\mathbf{W}_\beta \boldsymbol{x}_t)\beta_{\max},$$

$$\log\mu_t = \max\{f(\mathbf{W}_\mu \boldsymbol{x}_t), \mu_{\min}^{\log}\}.$$

- $f(z) = -a\,\mathrm{softplus}(z+b)$; $\eta_t \in (0,2)$.
- Text/illustration use a -1 momentum log floor.
- Reported experiments use -2, as specified in Tables 4 and 6.

Paper, Sec. 3.3, pp. 6–7; Tables 4 and 6, pp. 9 and 25.

Evaluation separates model quality, recall, and cost

	400M	1.3B
Training tokens	15B	100B
Tokens per batch	0.5M	1M
Training context	4K	4K
Warmup tokens	0.5B	1B

- SlimPajama subset, GPT-2 tokenizer, AdamW; matched data/tokens.
- Transformer, Mamba2, GDN, Comba, and pure-linear KDA baselines.
- PPL; eight-task reasoning mean; six-task retrieval mean at 2K.
- MQAR, LongBench at 16K, RULER NIAH, and H100 efficiency.

Paper, Sec. 4, pp. 7–9; Appendix G, pp. 23–24. Matched parameter scale does not imply matched recurrent state capacity.

At 400M, recall improves more than reasoning

Model	Lamb. PPL ↓	Wiki. PPL ↓	Reasoning ↑	Retrieval ↑
Transformer	54.36	32.80	48.06	30.42
Mamba2	60.42	33.45	47.85	21.12
GDN	45.63	32.10	48.51	22.93
Comba	46.19	31.73	48.91	24.01
KDA	43.44	31.96	49.06	24.47
MDN	**41.62**	**31.51**	**49.42**	**26.76**

- Versus the strongest other linear model: +0.36 reasoning points, +2.29 retrieval points.
- Transformer retains the best retrieval mean: 30.42.

Paper, Table 2, PDF p. 8. 400M parameters, 15B training tokens; retrieval uses 2K inputs. Reasoning averages mixed acc/acc_n.

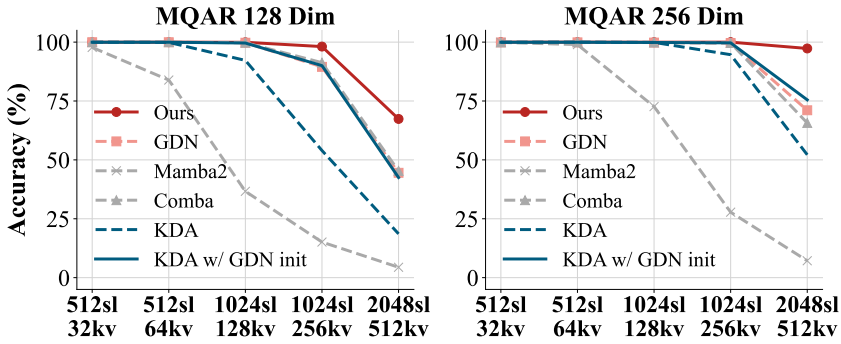
The pattern persists at 1.3B

Model	Lamb. PPL ↓	Wiki. PPL ↓	Reasoning ↑	Retrieval ↑
Transformer	17.90	18.99	57.95	40.60
Mamba2	18.20	19.14	57.10	32.45
GDN	16.12	18.51	57.44	32.80
Comba	15.17	18.37	58.49	35.25
KDA	16.83	19.24	58.56	32.62
MDN	**14.87**	**18.03**	**58.82**	**36.14**

- Best PPL and reasoning mean in this comparison.
- Retrieval: $+0.89$ points versus Comba, -4.46 versus Transformer.
- Gains vary by task: MDN's DROP score is below GDN and Comba.

Paper, Table 2, PDF p. 8. 1.3B parameters, 100B training tokens. Comparisons use this paper's matched evaluation protocol.

MQAR gains emerge at high association density



- Train on 64–256 tokens with 4–64 key–value pairs.
- Test up to 2,048 tokens and 512 pairs; MDN degrades later.
- Both length and pair count change; KDA initialization also matters.

Paper, Fig. 4, p. 7; Table 5, p. 24. Panels: embedding dimension 128/256; y-axis: accuracy.

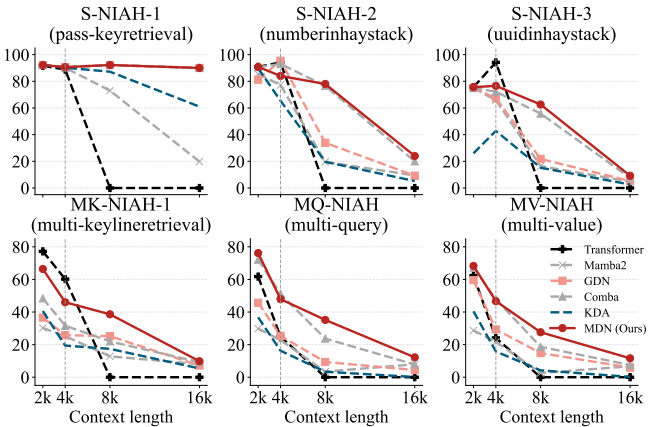
LongBench gains are strongest in code and summarization

Score ↑	GDN	Comba	KDA	MDN
Overall mean	19.28	18.11	18.62	**20.18**
LCC (code)	42.94	44.74	37.99	**50.50**
RepoBench-P	30.70	36.60	33.39	**39.13**
GovReport	7.81	9.60	10.02	**10.24**
QMSum	15.98	16.52	15.59	**17.20**
MultiNews	16.82	16.74	16.95	**18.85**

- 1.3B models: train at 4K, evaluate at 16K.
- Overall: +0.90 score points over GDN.
- Task trade-off: SSM is 15.65 for MDN versus 26.27 for GDN.

Paper, Table 3, PDF p. 8. Overall mean covers all 14 tasks, not only the rows shown. Task scores use different metrics.

NIAH improves beyond the 4K training length



At 8K context

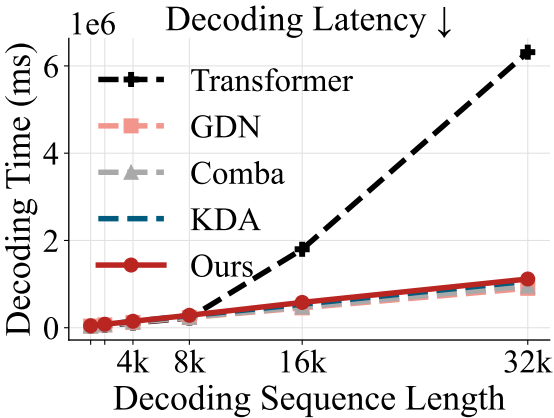
	MDN	Gain
MK	38.60	13.40
MQ	35.15	11.45
MV	27.60	8.95

Gain: points over the strongest other baseline per task.

Multi-needle accuracy still falls at 16K.

Paper, Fig. 5 and accompanying text, PDF p. 8. Dashed vertical line: 4K training context; red line: MDN.

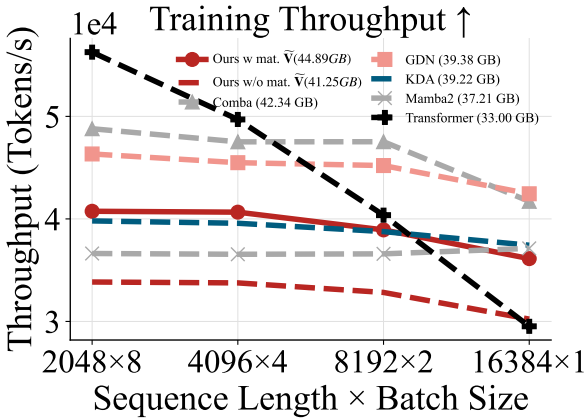
Decoding remains close to recurrent baselines



- MDN tracks GDN, Comba, and KDA as generated length grows.
- Transformer time grows faster at long lengths.
- Both recurrent matrices remain fixed in size.

Paper, Fig. 6 (left panel), PDF p. 8. 1.3B models, one H100. The plotted time is not per-token latency.

Training speed improves when residuals are saved



- Comparable to KDA and Mamba2.
- Below GDN and Comba.
- Red solid line saves $\widetilde{\mathbf{V}}$; red dashed line does not.

Reported memory:

$$41.25 \rightarrow 44.89 \text{ GB}.$$

Paper, Fig. 6 (right panel), PDF p. 8. 1.3B, one H100; sequence length $\times$ batch $= 16{,}384$ tokens.

The momentum ablation loses 6.64 retrieval points

400M variant	Lamb. PPL ↓	Reasoning ↑	Retrieval ↑
MDN	41.62	49.42	26.76
Without output correction	42.31	49.19	25.52
Without momentum*	47.01	49.26	20.12
Without $\beta_{\max}$	42.72	48.93	26.40
$\eta = 2\sigma(\cdot)$	49.10	49.41	25.54

- *Runnable proxy: GDN with $\alpha_{\max}, \beta_{\max}$ constraints.
- Directly setting $\mu = 0$ breaks this parallel kernel's divisions.
- No momentum floor: NaN at step 70; no $\alpha_{\max}$: NaN at step 1.

Paper, Table 4, p. 9; Table 6 and footnotes, p. 25. The mathematical recurrence itself permits $\mu = 0$.

The momentum floor changes the quality trade-off

Minimum $\log \mu$	Lamb. PPL ↓	Wiki. PPL ↓	Reasoning ↑	Retrieval ↑
-2 (reported)	**41.62**	31.51	49.42	**26.76**
-1.5	42.65	31.50	48.94	26.03
-1.357	44.90	31.44	49.50	25.31
-1	43.53	**31.39**	**49.88**	25.85

- A larger floor forces more momentum retention.
- -2 gives the strongest retrieval and LAMBADA PPL.
- -1 improves the reasoning mean and WikiText PPL.
- There is no single monotonic improvement across metrics.

Paper, Table 4, PDF p. 9. All rows are 400M models under the same training setup.

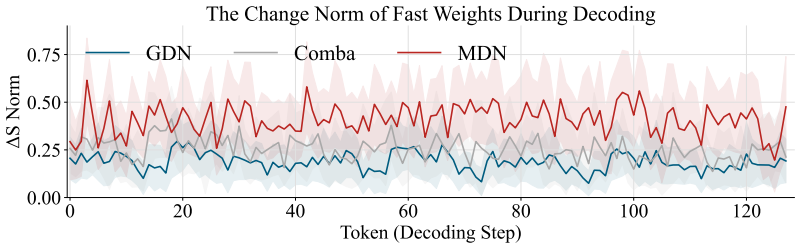
Sparse full-attention layers are promising, with trade-offs

400M hybrid	Lamb. PPL ↓	Wiki. PPL ↓	Reasoning ↑	Retrieval ↑
Comba-H 3:1	**40.88**	29.92	49.35	**34.72**
KDA-H 3:1	41.78	**29.49**	48.93	34.46
MDN-H 3:1	46.71	30.09	48.61	33.84
MDN-H 7:1	42.95	30.32	**49.68**	34.37

- Ratios count linear layers : full-attention layers.
- MDN's 7:1 improves reasoning and retrieval over its 3:1 variant.
- At 3:1, MDN does not dominate Comba or KDA.
- This table reports quality; it does not measure the speed gain of 7:1.

Paper, Table 4, PDF p. 9. Selected hybrid rows; all use the 400M training budget.

Memory updates are more active during decoding



$$\Delta S_t = \frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{S}_t^{(i)} - \mathbf{S}_{t-1}^{(i)} \right\|_F$$

- MDN has larger average step-to-step state changes.
- The authors connect this richer dynamics to retrieval improvements.
- Change magnitude is a descriptive probe, not a measure of information quality.

Paper, Fig. 7 and Sec. 4.3, PDF p. 9. The paper does not define the shaded-band statistic.

The contribution is an exact, practical momentum memory

1. **Algorithm:** geometric reordering and triangular residual solving preserve token-level updates.
2. **Quality:** recall and long-context gains over strong linear baselines; full attention still leads 2K retrieval.
3. **Cost:** an extra recurrent state and saved residuals; training trails GDN/Comba.

What remains open

Larger than 1.3B; faster dual-state kernels; broader hybrid configurations; stability of input-dependent matrix products.

Paper, Sec. 5, p. 9; Appendix I, p. 24. Matrix-product stability is a reading inference beyond the frozen-step analysis.

Reference and source conventions

Yulong Huang, Xiang Liu, Hongxiang Huang, Xiaopeng Lin, Zunchang Liu, Xiaowen Chu, Zeke Xie, and Bojun Cheng.

MDN: Parallelizing Stepwise Momentum for Delta Linear Attention.

arXiv:2605.05838v1, submitted 7 May 2026.

Author repository: github.com/HuuYuLong/MomentumDeltaNet

- All experimental numbers and figures are from this paper version.
- PDF page numbers are one-based and match the printed page numbers.
- Teaching derivations and interpretive limits are labeled explicitly.
- Backup slides clarify formula inconsistencies that affect implementation.

Backup: a one-row shift is essential for causality

$$\boldsymbol{\Gamma} = \begin{pmatrix} \gamma_{1,1} & 0 & 0 \\ \gamma_{2,1} & \gamma_{2,2} & 0 \\ \gamma_{3,1} & \gamma_{3,2} & \gamma_{3,3} \end{pmatrix}, \qquad \boldsymbol{\Gamma}^{-} = \begin{pmatrix} 0 & 0 & 0 \\ \gamma_{1,1} & 0 & 0 \\ \gamma_{2,1} & \gamma_{2,2} & 0 \end{pmatrix}.$$

The residual at position i reads $\mathbf{S}_{i-1}$

$$\widetilde{\boldsymbol{v}}_i^{\top} = \boldsymbol{v}_i^{\top} - \boldsymbol{p}_i^{\top} \mathbf{S}_{i-1}.$$

Its earlier-write coefficients are $\gamma_{i-1,j}$, not $\gamma_{i,j}$.

Simply deleting the diagonal from Γ produces the wrong coefficients.

Paper, Appendix C, PDF p. 18; Appendix E, PDF p. 22, code lines 130–133. Main-text p. 5 wording is imprecise.

Backup: use the consistent log-domain coefficients

$$A_i = \sum_{r \le i} \log \alpha_r, \quad U_i = \sum_{r \le i} \log \mu_r, \quad \ell_i = \mathrm{logcumsumexp}_{r \le i}(\log \beta_r + U_r - A_r).$$

$$b_i = \exp(A_i + \ell_i), \qquad \gamma_{i,j} = \exp(A_i + \ell_i - U_j)[1 - \exp(\ell_{j-1} - \ell_i)].$$

- $\ell_0 = -\infty$; coefficients are defined only on the causal region.
- Eq. (8): use $\mathrm{Diag}(\Gamma_{t,:})$, as in Appendix C.
- Eq. (10): $b_i = \bar{\alpha}_i c_i$ requires cumulative $\log \alpha$.
- Appendix E's `b_t` implements this consistent definition.

Paper, Eqs. (8)–(10), pp. 4–5; Eq. (30), p. 17; Appendix D, p. 19; Appendix E, p. 21.

Backup: the full spectrum includes orthogonal directions

$$\mathbf{A}_t = \begin{pmatrix} \alpha_t \mathbf{I} - \alpha_t \beta_t \eta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top & -\beta_t \mu_t \mathbf{I} \\ \alpha_t \eta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top & \mu_t \mathbf{I} \end{pmatrix}$$

$$\mathrm{Spec}(\mathbf{A}_t) = \{ \underbrace{\alpha_t}_{d_k - 1}, \underbrace{\mu_t}_{d_k - 1}, \lambda_+, \lambda_- \}.$$

- Key-aligned dynamics have trace r_t and determinant $\alpha_t \mu_t$.
- GDN also has orthogonal eigenvalues α_t; $\alpha_t(1 - \beta_t)$ is only its key-aligned eigenvalue.
- For $d_k > 1$ and standard positive gates, GDN's spectral radius is α_t.

Paper, Appendix F, Eqs. (58)–(60), PDF p. 23. Last two bullets correct the spectral-radius statement on p. 5.

Backup: the reported floor also permits right-half-plane roots

For unit keys, $0 < \alpha < 1$, $\beta \le 1 - \alpha$, and $0 < \eta < 2$:

$$\begin{aligned} r &= \alpha + \mu - \alpha\beta\eta \\ &\ge \mu + \alpha - 2\alpha(1-\alpha) \\ &= \mu + 2\left(\alpha - \tfrac{1}{4}\right)^2 - \tfrac{1}{8} \ge \mu - \tfrac{1}{8}. \end{aligned}$$

A sufficient condition derived for this reading

$\mu > 1/8$ gives positive trace and determinant, hence positive real parts. The reported floor satisfies $e^{-2} \approx 0.1353 > 1/8$.

This reconciles the -2 experimental floor with the spectral intent. It is a frozen-step result, not a proof of end-to-end training stability.

Reading derivation from Appendix F, Eq. (60), p. 23; constraints in Sec. 3.3, pp. 6–7; reported setting in Table 4, p. 9.

Backup: eliminating momentum reveals the second-order rule

For $\beta_{t-1} \neq 0$, substitute $\mathbf{M}_{t-1} = (\alpha_{t-1}\mathbf{S}_{t-2} - \mathbf{S}_{t-1})/\beta_{t-1}$:

$$\mathbf{S}_t = \left[\left(\alpha_t + \frac{\beta_t \mu_t}{\beta_{t-1}}\right)\mathbf{I} - \alpha_t \beta_t \eta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top\right]\mathbf{S}_{t-1} - \frac{\beta_t \mu_t \alpha_{t-1}}{\beta_{t-1}}\mathbf{S}_{t-2} + \beta_t \eta_t \boldsymbol{k}_t \boldsymbol{v}_t^\top.$$

- The additional $\mathbf{S}_{t-2}$ term explains the second-order interpretation.
- Table 1's summary omits α_t in the key-dependent transition term.
- The two-state recurrence avoids dividing by a previous step size.

Reading derivation from Eqs. (4)–(5), p. 4; compare Table 1, p. 3, with the consistent block matrix in Appendix F, p. 23.