Preconditioned DeltaNet

Curvature-aware Sequence Modeling for Linear Recurrences

Paper: Neehal Tumma, Noel Loo, Daniela Rus

Independent research paper discussion

arXiv:2604.21100v1 | 22 September 2026

A small change to writing improves associative memory

Main idea

Use historical key statistics to scale the *write key*, while keeping the *read key* unchanged.

- Exact inverse-Gram preconditioning connects linear attention and the delta rule.
- A bounded diagonal approximation retains efficient chunkwise computation.
- Paired language-model averages improve; individual tasks can regress.

Secs. 3–4, PDF pp. 3–8. Discussion: approximately 25 minutes.

DeltaNet already corrects errors, but uses one key twice

$$S_t = S_{t-1} + \beta_t \underbrace{(v_t - S_{t-1}k_t)}_{\text{prediction residual}} \underbrace{k_t^\top}_{\text{write direction}}$$

- $S_t \in \mathbb{R}^{d_v \times d_k}$ maps keys to values; $o_t = S_t q_t$.
- The residual reads memory using k_t; the update writes along the same k_t.
- One scalar gain does not adapt the update to coordinate-wise key history.

Secs. 1–2, PDF pp. 1–3.

The regression view identifies the missing curvature

$$\min_S \frac{1}{2}\sum_{i\le t}\|Sk_i - v_i\|_2^2 + \frac{\lambda}{2}\|S\|_F^2$$

$$G_t = \lambda I + \sum_{i\le t} k_i k_i^\top, \quad C_t = \sum_{i\le t} v_i k_i^\top, \quad S_t^\star = C_t G_t^{-1}$$

- G_t: key Gram; C_t: value–key cross-statistic. Take $\lambda > 0$ for invertibility.
- Test-time regression updates an in-sequence memory state, not the outer model parameters.
- DeltaNet is a per-token gradient step; it does not explicitly invert this curvature.

Sec. 2.3, Eqs. (1)–(2), PDF p. 2.

Exact ATQ and ATK produce the same ridge-regression map

Apply to query (ATQ)

$$C_t = C_{t-1} + v_t k_t^\top, \quad P_t = G_t^{-1}, \quad o_t = C_t(P_t q_t).$$

Apply to write key (ATK)

$$\tilde{k}_t = \frac{P_{t-1} k_t}{1 + k_t^\top P_{t-1} k_t}, \qquad S_t = S_{t-1} + (v_t - S_{t-1} k_t)\tilde{k}_t^\top$$

With $S_0 = 0$ and $P_0 = \lambda^{-1} I$, $S_t = C_t P_t$ for every t. Both forms are causal. This exact form has no extra arbitrary gain.

Theorem 3.1, Eqs. (3)–(5), PDF p. 3; proof: App. C.1, p. 18.

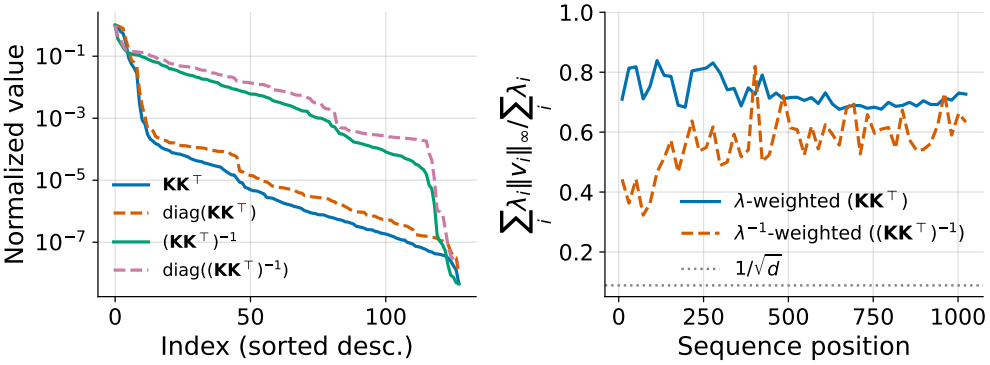
A diagonal approximation trades correlations for efficiency

$$A_t = A_{t-1} + k_t \odot k_t, \qquad P_t \approx \mathrm{Diag}(A_t + \lambda \mathbf{1})^{-1}$$

- Only $O(d_k)$ extra statistics, instead of a full $d_k \times d_k$ Gram.
- Coordinate-wise prefix sums and reciprocals support chunkwise computation.
- Off-diagonal correlations are discarded. Exact ATQ/ATK equivalence is lost.
- In general, $(\mathrm{diag}\, G)^{-1} \neq \mathrm{diag}(G^{-1})$.

Sec. 3.3, PDF pp. 4–5; App. C.2, p. 19. Ridge shown explicitly here.

Observed key geometry motivates the diagonal approximation



Left: sorted spectra. Right: eigenvalue-weighted coordinate alignment along the sequence. Evidence comes from a selected pretrained DeltaNet-340M layer.

Fig. 1, PDF p. 4. This is empirical motivation, not a general approximation guarantee.

The practical preconditioner is learned, bounded, and monotone

$$
\begin{aligned}
A_t &= \alpha_t^P A_{t-1} + \beta_t^P (k_t \odot k_t), \\
r_t &= \log A_t - \mu, \qquad s_t = \frac{r_t}{\mathbf{1} + |r_t|}, \\
B_t &= \exp[-\log(x) s_t], \qquad \tilde{k}_t = B_t \odot k_t.
\end{aligned}
$$

- Elementwise operations; $\mu > 0$ is learned per head; $x = 1.5$.
- $B_{t,j} \in [2/3, 3/2]$: larger accumulated activity means smaller write scaling.
- Uses current statistics and drops the exact-solve normalization.
- This is not an exact least-squares solver.

Eq. (9), Sec. 3.5, PDF pp. 5–6; App. E.1/E.3, pp. 25, 27–30.

One update covers PDN, PGDN, and PKDA

$$\bar{S}_t = S_{t-1} D_t, \qquad S_t = \bar{S}_t + \beta_t (v_t - \bar{S}_t k_t) \tilde{k}_t^\top, \quad o_t = S_t q_t$$

Model	Main decay D_t	Write key
PDN	I	$B_t \odot k_t$
PGDN	$\alpha_t I$	$B_t \odot k_t$
PKDA	$\mathrm{Diag}(\boldsymbol{\alpha}_t)$	$B_t \odot k_t$

The statistics have independent gates α_t^P, β_t^P. Do not replace the key inside the residual with $\tilde{k}_t$.

Secs. 3.4–3.5, PDF pp. 5–6. Diagonal decay expanded using the stated state dimensions.

Two keys still admit the compact delta-rule algorithm

For one chunk, $Q, K, \tilde{K} \in \mathbb{R}^{C \times d_k}$ and $V \in \mathbb{R}^{C \times d_v}$.

$$
\begin{aligned}
T &= [I + \mathrm{tril}(\mathrm{Diag}(\beta) K \tilde{K}^\top, -1)]^{-1} \mathrm{Diag}(\beta), \\
W &= TK, \qquad U = TV, \qquad R = U - W S_{\mathrm{in}}^\top, \\
S_{\mathrm{out}} &= S_{\mathrm{in}} + R^\top \tilde{K}, \\
O &= Q S_{\mathrm{in}}^\top + [(Q \tilde{K}^\top) \odot M] R.
\end{aligned}
$$

M: causal mask. The triangular solve uses read–write interactions. State writing and local output use the write key.

Eq. (6), PDF p. 4; App. F.2.1, pp. 35–36. Shown without decay.

Preserving two interaction matrices matters for throughput

- Original delta rule: interactions $KK^\top$ and $QK^\top$.
- Preconditioned delta rule: $K\tilde{K}^\top$ and $Q\tilde{K}^\top$.
- General DPLR needs four distinct interaction paths in the cited implementation.
- Compute diagonal statistics first; then use the specialized two-key kernel.
- Backward separates read/write gradients and propagates through the preconditioner.

App. D, PDF pp. 21–24, Figs. 8–9 and Table 3; App. F, pp. 33–37.

The eigenvalue bound is useful, but local

For PGDN with $\|k_t\|_2 = 1$, $\alpha_t, \beta_t \in [0, 1]$:

$$T_t = \alpha_t(I - \beta_t k_t \tilde{k}_t^\top), \qquad \lambda(T_t) = \begin{cases} \alpha_t & (d_k - 1 \text{ copies}), \\ \alpha_t(1 - \beta_t k_t^\top \tilde{k}_t) & (1 \text{ copy}). \end{cases}$$

Since $k_t^\top \tilde{k}_t \in [1/x, x]$, $\beta_t x \leq 2$ ensures $|\lambda(T_t)| \leq 1$.

Reading boundary

A one-step spectral bound does not by itself establish stability of every time-varying matrix product or training gradient. The scalar-decay formula is not directly a PKDA proof.

App. E.3, PDF pp. 29–30. Boundary statement: our mathematical interpretation.

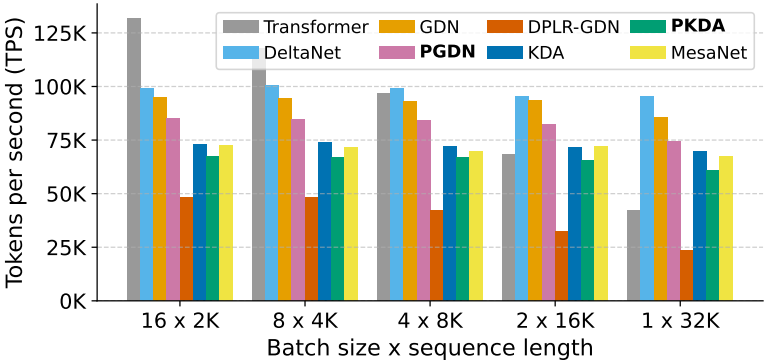
Evaluate within matched architecture pairs

	Smaller models	Larger models
Training tokens	15B	50B
Parameter scale	340M / 355M	1B
Context / batch	2048 / 0.5M tokens	2048 / 0.5M tokens
Pairs	DN, GDN, KDA families	GDN, KDA families

- SlimPajama; AdamW; peak LR 4×10^{-4}; 8 H100 GPUs.
- KDA/PKDA use an output gate; the smaller pair is actually 355M.
- Compare each recurrence with its preconditioned version. Cross-family tuning differs.
- Main accuracy tables do not report multi-seed variability.

Sec. 4, PDF pp. 6–8; App. E.1, p. 25.

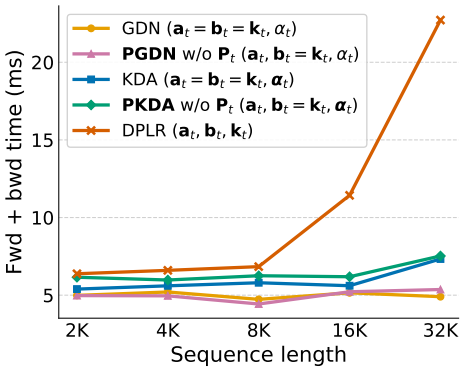
End-to-end throughput pays for the extra statistics



Higher is better. Author summary: about 10% overhead for PGDN/PKDA; PGDN more than 20% faster than MesaNet in this profiling setup.

Fig. 3, Sec. 4.1, PDF p. 6. 340M-scale models, 8 H100s, DDP; batch size × length on x-axis.

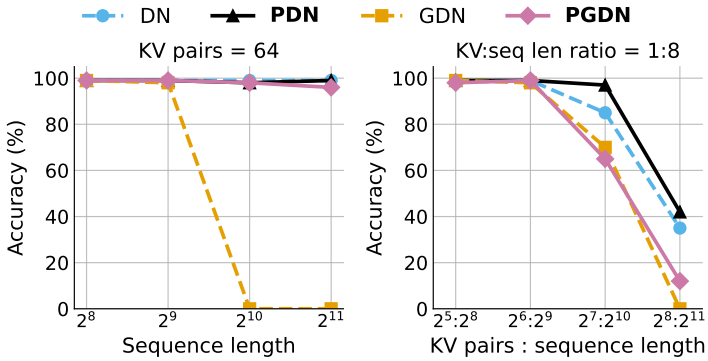
Two-key kernels add little cost when statistics are excluded



Lower is better. Near-baseline two-key kernel timing does not remove the end-to-end cost of computing A_t and B_t.

Fig. 4, PDF p. 6. Forward + backward, batch 1, 8 heads, head dimension 128.

MQAR supports better resistance to associative interference



Accuracy versus sequence length. Left: 64 KV pairs. Right: KV count / length = 1:8. Preconditioning generally maintains or improves accuracy in these cuts.

Fig. 5, PDF p. 7; fixed-length KV sweep: Fig. 10, p. 26. No seed error bars shown.

All five paired task averages increase

Pair	Commonsense avg.	Real-world ICR avg.
DN → PDN (340M)	47.41 → 47.96	22.59 → 23.84
GDN → PGDN (340M)	48.86 → 49.96	24.77 → 26.17
KDA → PKDA (355M)	49.12 → 49.89	25.88 → 26.76
GDN → PGDN (1B)	55.44 → 56.38	33.87 → 34.99
KDA → PKDA (1B)	55.43 → 56.64	34.55 → 34.88

Accuracy in percent, higher is better. Averages cover 8 commonsense and 5 ICR tasks; perplexities are excluded. Individual tasks can regress: 1B PKDA FDA 23.23 → 21.69, SWDE 50.14 → 48.79.

Table 1, PDF p. 7. Matched-pair interpretation; not a cross-family ranking.

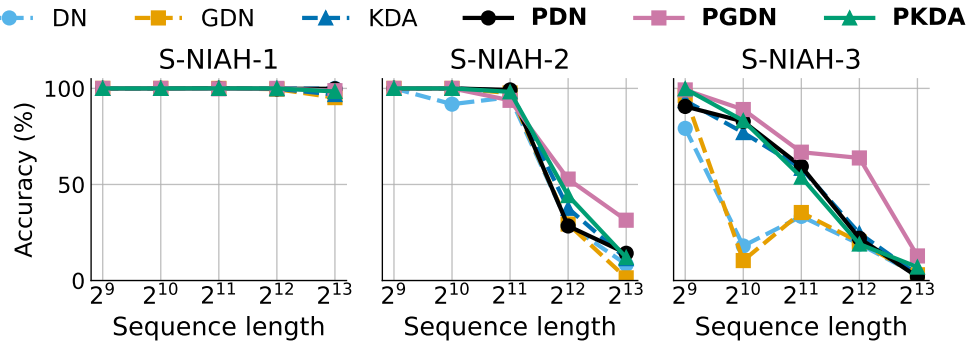
Perplexity gains are larger on LAMBADA than WikiText

Pair	LAMBADA ppl ↓	WikiText ppl ↓
DN → PDN (340M)	45.76 → 43.05	29.04 → 29.05
GDN → PGDN (340M)	31.39 → 28.43	27.08 → 26.86
KDA → PKDA (355M)	31.37 → 25.33	26.18 → 25.98
GDN → PGDN (1B)	13.46 → 13.09	18.05 → 18.01
KDA → PKDA (1B)	14.52 → 11.86	17.95 → 17.82

1B KDA's LAMBADA perplexity drops by 2.66, or about 18.32% relative. This is not an accuracy gain. The small DN WikiText regression is retained in the comparison.

Table 1, PDF p. 7. Relative percentage calculated from reported values.

Long-context retrieval improves unevenly



Trained at 2K; evaluated through 8K. Easy S-NIAH-1 largely saturates. Difficult retrieval still deteriorates with length.

Fig. 6, PDF p. 8; exact values: Table 5, p. 27. Smaller models only in this plot.

Strong PGDN gains coexist with explicit counterexamples

S-NIAH-3 pair	Length	Base	Preconditioned
GDN / PGDN (340M)	1K	10.4	89.0
GDN / PGDN (340M)	4K	20.0	63.8
GDN / PGDN (340M)	8K	3.0	12.8
KDA / PKDA (355M)	4K	24.6	19.0
GDN / PGDN (1B)	4K	64.2	63.8
KDA / PKDA (1B)	1K	98.0	92.4

Accuracy in percent. The 340M GDN 4K gain is 43.8 percentage points. Do not translate the paper's broad prose into universal task-by-length dominance.

Table 5, PDF p. 27. No multi-seed uncertainty is reported for these rows.

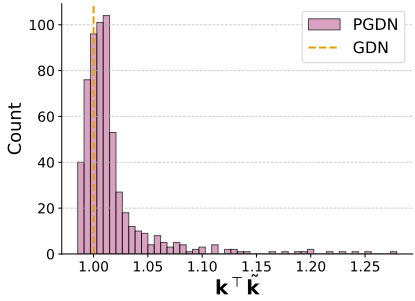
Trainability choices matter as much as adding curvature

340M / 15B-token configuration	Training loss ↓
PGDN, bounded final form	2.511
PGDN, tied statistic/main gates	2.518
GDN	2.528
PGDN, unstable ATK	2.536
P-Mamba-2, ATQ	2.543

Extra normalizations of the final preconditioner worsen loss: 2.511 without normalization vs. 2.517, 2.521, 2.524. The ATQ comparison changes the base recurrence as well; it is not a universal placement theorem.

Tables 6–7, PDF pp. 27, 30. The anomalous 5.520 entry is discussed in the notes and backup.

The mechanism evidence is suggestive, not a causal isolation



- $k_t^\top \tilde{k}_t$ differs from 1, altering the write eigenvalue.
- Observed PGDN does not learn negative eigenvalues (App. E.4).
- Negative-spectrum baseline: GDN ICR avg. 26.22 vs. PGDN 26.17 at 340M.

Fig. 7, PDF p. 8; Table 4, p. 26; App. E.4/Fig. 16, pp. 31, 33.

DGPS separates four practical design decisions

Axis	Decision
Decay	What state information is retained?
Gain	How strongly is the residual written?
Preconditioner	How are key/query directions transformed?
Solve	Accumulate statistics or update the fitted map?

- TTR supplies the regression interpretation; structured recurrences supply efficient kernels.
- A weighted proximal norm also derives key-preconditioned updates.
- Proposed but unevaluated variants should remain future directions.

Sec. 5/Table 2, PDF pp. 8–9; App. B and G, pp. 15–17, 38–40.

The contribution is useful within clear evidence boundaries

- A distinct, history-dependent write key is a practical lever for linear recurrent models.
- Exact theory motivates the design; the trained model uses a bounded approximation.
- Matched task averages improve with about 10% reported throughput overhead.
- No universal long-context gain, equal-compute quality frontier, or multi-seed significance claim.
- Open questions: larger-scale training, stronger stability guarantees, and richer cheap preconditioners.

Paper conclusions, Sec. 6, PDF p. 9; evidence boundaries synthesized from Sec. 4 and App. C/E.

Reference and reading provenance

Neehal Tumma, Noel Loo, Daniela Rus.
Preconditioned DeltaNet: Curvature-aware Sequence Modeling for Linear Recurrences.

arXiv:2604.21100v1, submitted 22 April 2026.

Read on 22 September 2026 using the matching author TeX and 40-page PDF, including appendices A–G. Original figures are reproduced from the author's source bundle.

This deck is an independent reading presentation. Experiments were not rerun. Source locators use 1-based PDF pages.

Backup: exact equivalence fails after diagonal approximation

Paper counterexample: $\lambda = 1$, $k_1 = v_1 = (1,1)^\top$, $S_0 = 0$.

$$\widehat{G}_1 = 2I, \qquad S_1^{\mathrm{ATQ}} = \tfrac{1}{2}\mathbf{1}\mathbf{1}^\top.$$

Using the previous diagonal inverse and its normalized ATK gain:

$$\tilde{k}_1 = \frac{I\mathbf{1}}{1 + \mathbf{1}^\top I\mathbf{1}} = \tfrac{1}{3}\mathbf{1}, \qquad S_1^{\mathrm{ATK}} = \tfrac{1}{3}\mathbf{1}\mathbf{1}^\top.$$

Hence the two constructions are not generally equivalent. Special data can still make their outputs agree.

App. C.2, PDF p. 19.

Backup: the DN–LA theorem needs the proof assumptions

The proof additionally assumes $\lambda = 0$, $v_t = S^\star k_t$ (noiseless fixed map), and $\beta_t \in [0, 1]$. Keys and independent queries are uniform on the unit sphere.

$$\mathbb{E}\|o_t^{\mathrm{LA}} - o_t^\star\|^2 = \frac{\|S^\star\|_F^2}{d}\left(1 - \frac{t}{d} + \frac{t(t-1)}{d^2}\right),$$

$$\mathbb{E}\|o_t^{\mathrm{DN}} - o_t^\star\|^2 \le \frac{\|S^\star\|_F^2}{d}.$$

For $t \ge d + 1$, the LA factor is at least 1. This does not prove superiority of the final practical ATK model over every ATQ model.

Theorem 3.2, PDF p. 5; assumptions and proof: App. C.5, pp. 20–21.

Backup: provenance issues should not be silently repaired

- Table 6 lists GDN$[-1, 1]$ training loss as **5.520** in both TeX and PDF. It is anomalous; we do not guess a correction.
- Table 2 caption calls G_t the inverse Gram; Secs. 2–3 define G_t as Gram and $P_t = G_t^{-1}$. Use the equations.
- App. E.3 sometimes describes bounding A_t; the actual bounded quantity is B_t.
- MesaNet training failed in the authors' setup, so speed results do not establish a completed matched-quality comparison.

Table 2, PDF p. 9; Table 6, p. 27; App. E.3/E.5, pp. 28, 31.