Dual DeltaNet

Value-Side Memory Reorganization for Linear Attention

Paper by Yixiao Qian, Yaxin Zhang, Song Chen,
Jiaxu Liu, Shengze Cai, and Chao Xu

arXiv:2610.38578v1 | Research group reading | 13 November 2026

Reading presentation prepared with Codex; not an author presentation.

A second read direction improves fixed-state recall

Central idea

Correct the value associated with a key, then correct the address response associated with selected value features.

- One matrix state; complementary key-side and value-side delta rules.
- At 1B: NIAH average 74.52%, versus KDA's 64.27%.
- Higher latency than KDA; gains vary with task and scale.

Paper: arXiv v1, submitted 28 October 2026.
Publication not found as of 13 November 2026.
Official code: DualDeltaNet — implementation and model configurations available.

Paper, Secs. 3–5, PDF pp. 3–8; Table 3. Publication/code checked 2026-11-13.

A fixed-size state forces associations to share space

$$S_t \in \mathbb{R}^{d_k \times d_v}, \qquad S_t = S_{t-1} + k_t v_t^\top, \qquad o_t = S_t^\top q_t$$

- $q_t, k_t \in \mathbb{R}^{d_k}$; $v_t, o_t \in \mathbb{R}^{d_v}$.
- Rows act as d_k addresses storing d_v-dimensional content.
- Fixed recurrent memory avoids a growing token-level KV cache.
- Overlapping writes can interfere: useful associations must survive compression.

Research question

Can we improve memory organization while keeping the same state capacity?

Paper, Secs. 1–2, PDF pp. 1–3, Eqs. (1)–(2).

KDA corrects the value retrieved by the current key

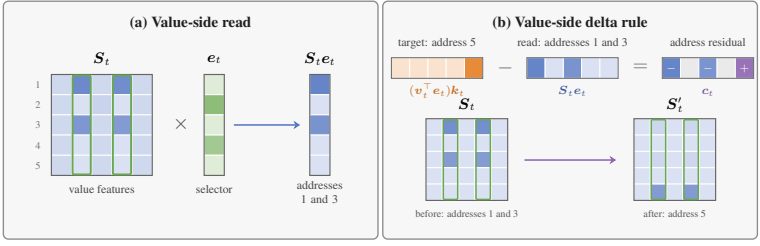
$$\bar{S}_{t-1} = D_t S_{t-1}, \qquad D_t = \mathrm{Diag}(\alpha_t)$$
$$S_t^{\mathrm{KDA}} = \bar{S}_{t-1} + \beta_t k_t \underbrace{(v_t - \bar{S}_{t-1}^{\top} k_t)^{\top}}_{\text{value residual}}$$

- q_t, k_t are L2-normalized; $\alpha_t \in (0,1)^{d_k}$.
- Decay controls retention separately for each key channel.
- $\beta_t \in (0,1)$ scales correction at the current key direction.
- This read does not explicitly inspect where related value features reside.

Paper, Secs. 2–3, PDF p. 3, Eqs. (4)–(6).

Reading from the value side reveals address occupancy

$$S^\top k = \sum_j v_j (k_j^\top k), \qquad Se = \sum_j k_j (v_j^\top e)$$



- $e \in \mathbb{R}^{d_v}$ selects value features; $Se \in \mathbb{R}^{d_k}$ aggregates their addresses.
- An address response need not recover an individual historical semantic key.

Paper, Sec. 1, PDF p. 2; original Fig. 1, PDF p. 3. Discrete relocation is schematic.

Select features for reading; use the value direction for writing

$$\hat{v}_t = \frac{v_t}{\|v_t\|_2}, \quad m_t = \mathrm{sigmoid}(s_t^m), \quad e_t = \frac{m_t \odot \hat{v}_t}{\|m_t \odot \hat{v}_t\|_2}, \quad \tau_t = v_t^\top e_t$$

Value-side delta rule

$$S_t = S_t^{\mathrm{KDA}} + \gamma_t \underbrace{(\tau_t k_t - S_t^{\mathrm{KDA}} e_t)}_{\text{address residual}} \hat{v}_t^\top$$

- $\gamma_t = \mathrm{sigmoid}(s_t^\gamma)$ controls correction strength.
- $\tau_t k_t$ is the read response of the target association $k_t v_t^\top$.
- The selector e_t probes memory; $\hat{v}_t$ is the write direction.

Paper, Sec. 3, PDF p. 3, Eqs. (7)–(8).

The selected address response moves toward its target

Let $r = S_t^{\mathrm{KDA}} e_t$ and $c_t = \hat{v}_t^{\top} e_t$. Then

$$S_t e_t = (1 - \gamma_t c_t) r + \gamma_t c_t \tau_t k_t.$$

- Positive channel gates imply $0 < c_t \leq 1$ for a nonzero value.
- The selected response interpolates toward the target address response.
- With $e_t = \hat{v}_t$, $c_t = 1$: the effective step is exactly γ_t.
- A continuous rank-one correction changes rows of the shared matrix.

Interpretation

The rule acts on an aggregate response; other stored associations can also change.

Teaching derivation from paper Eq. (8), PDF p. 3; not a separate theorem in the paper.

Left and right transitions act on the same memory

$$A_t = I - \beta_t k_t k_t^\top, \qquad B_t = I - \gamma_t e_t \hat{v}_t^\top$$

Equivalent two-sided recurrence

$$S_t = A_t D_t S_{t-1} B_t + \eta_t k_t \hat{v}_t^\top$$
$$\eta_t = \beta_t \|v_t\|_2 + \gamma_t (1 - \beta_t) \tau_t$$

- A_t: key-side erasure; B_t: value-feature transition, generally asymmetric.
- The effective write strength includes both correction stages.
- Setting $\gamma_t = 0$ recovers KDA.

Paper, Sec. 3, PDF p. 3, Eqs. (9)–(10).

A rank-2 increment fuses both corrections for decoding

Read both sides from the decayed state:

$$\ell_t = \bar{S}_{t-1}^\top k_t, \quad r_t = \bar{S}_{t-1} e_t, \quad \tilde{v}_t = v_t - \ell_t$$

$$\tilde{k}_t = (\tau_t - \beta_t \tilde{v}_t^\top e_t) k_t - r_t$$

Pack two outer products:

$$\mathcal{K}_t = [k_t \ \tilde{k}_t], \quad \mathcal{V}_t = \begin{bmatrix} \beta_t \tilde{v}_t^\top \\ \gamma_t \hat{v}_t^\top \end{bmatrix}, \quad S_t = \bar{S}_{t-1} + \mathcal{K}_t \mathcal{V}_t$$

- The correction in $\tilde{k}_t$ accounts for the intervening KDA update.
- Increment rank ≤ 2; the full state can have much higher rank.
- Readout: $o_t = d_k^{-1/2} S_t^\top q_t$.

Paper, Algorithm 1, PDF p. 4.

More computation shares the same leading state traffic

	KDA	DDN
Leading FLOPs/token	$7d_k d_v$	$11d_k d_v$
State load + store	$4d_k d_v$ bytes	$4d_k d_v$ bytes
Asymptotic intensity	1.75 FLOPs/byte	2.75 FLOPs/byte

- Assumption: 2-byte state/input/output elements, state traffic dominant.
- Additional value-side input is only $\mathcal{O}(d_v)$.
- Theoretical intensity rises by 57.1%; this can improve memory-bound utilization.
- Actual latency also depends on reductions, scheduling, and kernel parallelism.

Paper, Table 1 and Sec. 3, PDF p. 4. Percent change calculated from 2.75/1.75.

Chunkwise training first removes channel-wise decay

Within a length-C chunk, suppress the chunk index and define

$$\Pi^i = \mathrm{Diag}(\alpha^i \odot \cdots \odot \alpha^1), \qquad \widehat{S}^i = (\Pi^i)^{-1} S^i.$$

$$\bar{k}^i = (\Pi^i)^{-1} k^i, \quad \bar{u}^i = \beta^i \Pi^i k^i, \quad \bar{A}^i = I - \bar{k}^i (\bar{u}^i)^\top$$

$$\widehat{S}^i = \bar{A}^i \widehat{S}^{i-1} B^i + \eta^i \bar{k}^i (\hat{v}^i)^\top$$

- Unrolling gives $P_L S_{\mathrm{in}} P_R + \sum_i \eta^i h^i (z^i)^\top$.
- Left and right products propagate each write to the chunk boundary.
- All α are positive; implementation uses FP32 cumulative log-decay.

Paper, Sec. 4, PDF pp. 4–5, Eqs. (11)–(16); App. B, PDF p. 11.

Two compact WY factors replace long transition products

Stack $\bar{k}^i, \bar{u}^i, e^i, \hat{v}^i$ as rows of $\bar{K}, \bar{U}, E, \hat{V}$.

$$T_L = \mathrm{tril}(\bar{U}\bar{K}^\top, -1), \quad T_R = \mathrm{tril}(\Gamma E \hat{V}^\top, -1), \quad \Gamma = \mathrm{Diag}(\gamma^i)$$

$$M_L = (I + T_L)^{-1}, \quad M_R = (I + T_R)^{-1}, \quad H = \bar{K}^\top M_L, \quad Z = M_R^\top \hat{V}$$

Chunk boundary, with $N = \mathrm{Diag}(\eta^i)$

$$\widehat{S}^C = (I - H\bar{U})S_{\mathrm{in}}(I - E^\top \Gamma Z) + HNZ$$

- H: surviving key directions; Z: surviving value directions.
- Unit-triangular systems encode within-chunk interactions.
- Chunks still exchange recurrent boundary states sequentially.

Paper, Sec. 4, PDF p. 5, Eqs. (17)–(25); Lemma A.1, PDF p. 11.

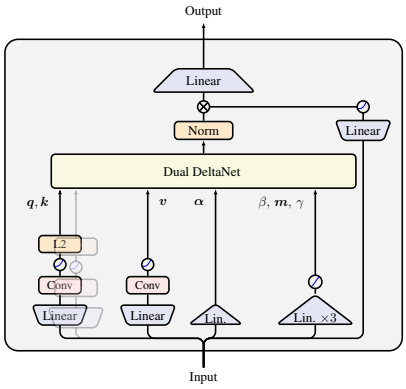
Causal masks keep every query inside its own prefix

$$
\begin{aligned}
(Q_\pi)_{j,:} &= d_k^{-1/2}(\pi^j \odot q^j)^\top, \qquad F_R = M_R \Gamma E \\
G_L &= \mathrm{tril}(Q_\pi \bar{K}^\top) M_L, \\
Q_{\mathrm{eff}} &= Q_\pi - G_L \bar{U}, \qquad O_{\mathrm{base}} = Q_{\mathrm{eff}} S_{\mathrm{in}}, \\
\Phi &= G_L N M_R^\top, \\
O &= O_{\mathrm{base}} + \mathrm{tril}(\Phi - O_{\mathrm{base}} F_R^\top)\hat{V}.
\end{aligned}
$$

- O_{base}: incoming-state contribution after left-side transitions.
- The remaining term combines writes and right-side correction.
- Here tril includes the diagonal: token j may read its own update.

Paper, Sec. 4, PDF pp. 5–6, Eqs. (26)–(33), Algorithm 2.

Value-side reductions constrain kernel parallelism



Original Fig. 2, PDF p. 4; App. B, PDF p. 11.

- KDA-like projections, causal convolutions, SiLU; normalized q, k.
- New gates: feature selector m_t and correction strength γ_t.
- The *Se* reduction couples value columns, limiting independent value tiling.
- TileLang kernel: one program per sequence/head; full state in shared memory.
- Small batch sizes can leave too few independent programs.

Comparisons match tokens, total parameters, and state size

Scale	Tokens	Layers	State elements/layer
130M	∼5B	12	98,304
340M	∼15B	24	131,072
1B	∼50B	24	229,376

- FineWeb-Edu; 2,048-token training context; batch 256 sequences.
- Shared LLaMA tokenizer (32K vocabulary), AdamW and cosine schedule.
- Baselines: KDA, GDN-2, Mamba-3 SISO and rank-4 MIMO.
- Model/MLP widths are adjusted: equal totals do not imply identical blocks.

Paper, Sec. 5, PDF p. 6; App. C, Tables 7–8, PDF pp. 11–12.

General modeling improves modestly over KDA

Scale	Model	Train ppl ↓	WT2 ppl ↓	LAMB. ppl ↓	Avg. acc. ↑
130M	KDA	14.79	42.53	69.91	37.38
	DDN	14.78	42.36	60.93	**38.03**
340M	KDA	10.93	28.13	29.09	42.40
	DDN	10.88	27.43	28.41	**43.44**
1B	KDA	8.23	19.07	13.86	50.37
	DDN	8.22	19.00	13.56	**50.59**

- Seven-task mean is highest among all tested models at each scale.
- At 1B, DDN trails KDA on WinoGrande: 54.46% vs. 56.12%.
- Training-ppl gaps can be 0.01; run-to-run variation is not reported.

Paper, Table 2, PDF p. 7. Accuracies are percentages; HellaSwag uses normalized accuracy.

NIAH provides the strongest average gains

Scale	M3 SISO	M3 MIMO	GDN-2	KDA	DDN
130M	12.95	11.65	26.88	38.10	**40.57**
340M	35.02	41.53	48.57	51.20	**53.87**
1B	47.25	60.73	69.40	64.27	**74.52**

- Mean accuracy (%) over four RULER variants and three context lengths.
- Contexts: 1,024 / 2,048 / 4,096; 500 examples per variant/length.
- Tasks: repeated-text numbers, essay numbers, essay UUIDs, multiple keys.
- At 1B, DDN improves over KDA by 10.25 percentage points.

Paper, Table 3, PDF p. 7; evaluation definitions in App. C, PDF p. 12.

The average hides a major 340M Single-2 regression

Scale / task	Context	KDA (%)	DDN (%)	Difference (pp)
1B / Single-2	4,096	54.2	80.8	+26.6
1B / Single-3	1,024	59.4	97.0	+37.6
1B / Single-3	4,096	19.0	36.0	+17.0
340M / Single-2	2,048	95.6	67.2	−28.4
340M / Single-2	4,096	40.6	14.4	−26.2

- At 340M, Single-1/Single-3 improvements outweigh Single-2 losses.
- MultiKey-1 also remains difficult; at 1B/1K, M3 MIMO scores 53.2%, DDN 37.6%.
- The paper does not explain the Single-2 regression.

Paper, Table 3, PDF p. 7. Selected cells; differences calculated from original values.

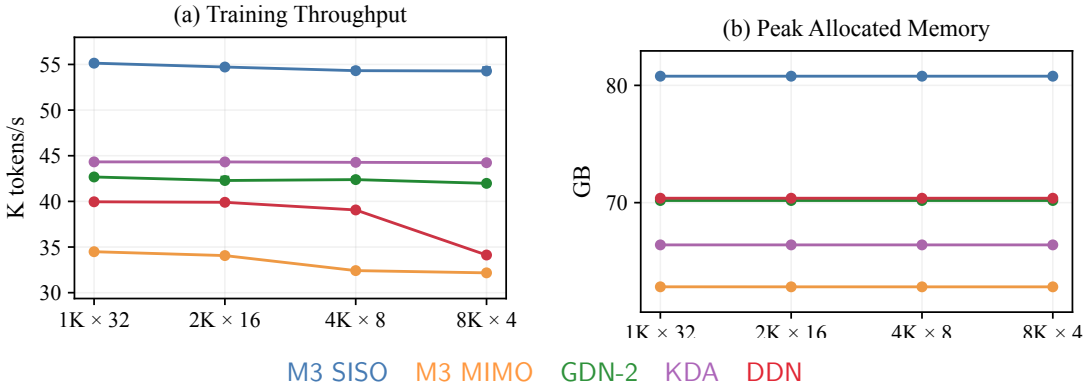
Real-world retrieval favors different updates by task

Scale	KDA	GDN-2	DDN
130M	15.18	15.53	**16.18**
340M	21.31	21.54	**23.22**
1B	30.90	**31.80**	31.20

- Six-task mean contains accuracy (%); inputs capped at 2,048 tokens.
- SWDE, SQuAD Completion, FDA, TriviaQA/NQ/DROP Cloze.
- At 1B, GDN-2's FDA score is 17.42%, versus DDN's 12.89%.
- DDN leads NQ (17.45%), but trails KDA on SWDE (32.58% vs. 35.46%).

Paper, Table 4, PDF p. 8; App. C, PDF p. 12. Greedy generation, at most 48 new tokens.

Training speed depends on memory strategy and parallelism



- 1B, BF16, H200; roughly fixed tokens. DDN slows at 8K $\times$ 4.
- Default recomputation differs: SISO spends more memory for speed.

Original Fig. 3 panels, PDF p. 8; legend retypeset. See App. B, PDF p. 11 for parallelism.

DDN adds about 14% total inference latency over KDA

	4,096 tokens		8,192 tokens	
Model	Prefill	Prefill+Decode	Prefill	Prefill+Decode
M3 SISO	1.35	14.66	2.69	27.44
M3 MIMO	1.64	16.70	3.35	33.46
GDN-2	1.58	16.53	3.16	33.11
KDA	1.39	14.66	2.81	29.34
DDN	1.57	16.68	3.15	33.37

- Seconds; 1B, H200, batch 64; generated length equals input length.
- At 4K: prefill +12.95%; prefill+decode +13.78% versus KDA.
- DDN latency is close to GDN-2 and M3 MIMO.

Paper, Table 5, PDF p. 8. Overheads calculated as (DDN/KDA -1)$\times$100%.

Feature gating helps NIAH, with a retrieval tradeoff

Scale	Variant	Downstream	Retrieval	NIAH
130M	KDA	37.38	15.18	38.10
	DDN, $e = \hat{v}$	37.41	15.87	39.45
	DDN, gated	**38.03**	**16.18**	**40.57**
340M	KDA	42.40	21.30	51.20
	DDN, $e = \hat{v}$	42.16	**23.48**	52.42
	DDN, gated	**43.44**	23.22	**53.87**

- All columns are mean accuracies (%). Both DDN variants keep the value-side rule.
- Gating improves NIAH by 1.12/1.45 pp over ungated DDN.
- 340M retrieval favors the ungated selector by 0.26 pp.

Paper, Table 6, PDF p. 9. Its 340M KDA retrieval is 21.30; Table 4 reports 21.31.

The next controls should explain when reorganization helps

- **Attribution:** a parameter-matched $\gamma = 0$ control would help isolate the value-side correction from architecture adjustments.
- **Failure mechanism:** why does 340M Single-2 regress by 26–28 pp while other variants improve?
- **Precision of small gains:** repeated training would clarify the 0.01-ppl and sub-point downstream differences.
- **Deployment:** value-column coupling limits small-batch parallelism; the current benchmark uses batch 64.

Reading assessment

Strong algebraic specification and promising recall results; task-specific behavior and kernel cost remain central to adoption.

Analysis based on Tables 2–6, PDF pp. 7–9; architecture/implementation, App. B–C, pp. 11–12.

Two-sided memory maintenance is the main contribution

1. **Mechanism:** key-side value correction and value-side address-response correction share one fixed-size state.
2. **Computation:** two-sided compact WY supports chunkwise training; a rank-2 increment fuses recurrent decoding.
3. **Evidence:** the largest gains appear in NIAH; real retrieval and latency expose the practical tradeoffs.

Paper, Secs. 3–6, PDF pp. 3–9. Main talk: approximately 25 minutes; derivation/details follow.

References and implementation

- Qian et al. (2026). *Dual DeltaNet: Value-Side Memory Reorganization for Linear Attention*.
  arXiv:2610.38578v1; all paper locators refer to this version.
- Official implementation: ReichtumQian/DualDeltaNet.
  Includes DDN layers, models, kernels and training configurations.
- Current arXiv identity page: no formal publication record found as of 13 November 2026.

Figures 1–2 cropped from the paper; Fig. 3 reused from author TeX assets. Tables are selected/retypeset original values.

Backup: the compact WY identity follows by telescoping

For $Q_i = I - u_i w_i^\top$, let $f_i = Q_m \cdots Q_{i+1} u_i$.

$$Q_m \cdots Q_{i+1} = I - \sum_{j>i} f_j w_j^\top$$

$$f_i = u_i - \sum_{j>i} f_j (w_j^\top u_i)$$

Stack columns in F, U, W and set $T = \mathrm{tril}(W^\top U, -1)$:

$$F(I + T) = U, \qquad Q_m \cdots Q_1 = I - U(I + T)^{-1} W^\top.$$

- $I + T$ is unit lower triangular, so it is always invertible.
- No orthogonality assumption on u_i, w_i is required.

Paper, App. A, Lemma A.1, PDF p. 11, Eqs. (34)–(37).

Backup: collect the chunk boundary into matrix products

With $H = \bar{K}^\top M_L$ and $Z = M_R^\top \hat{V}$,

$$
\begin{aligned}
F_R &= M_R \Gamma E, \\
R &= S_{\text{in}} - S_{\text{in}} F_R^\top \hat{V}, \\
\Psi &= NZ - \bar{U} R, \\
S_{\text{out}} &= \Pi^C (R + H\Psi).
\end{aligned}
$$

- R: incoming state after all right-side transitions in the chunk.
- $H\Psi$: collects new writes and left-side correction.
- Restore cumulative decay only after the normalized boundary update.

Paper, Sec. 4, PDF p. 5, Eqs. (23)–(25), Algorithm 2.

Backup: parameterization, precision, and training details

$$\alpha_t = \exp[-\exp(a_{\text{decay}}) \odot \Delta_t], \qquad \Delta_t = \text{softplus}(W_\Delta x_t + b_\Delta)$$

- Separate low-rank decay/selector projections; bottleneck width d_v.
- Initialize $m_t = 0.5$, $\gamma_t = 0.05$, hence $e_t = \hat{v}_t$ initially.
- BF16 operands/state; FP32 accumulation, normalization, log-decay and triangular intermediates.
- AdamW: $(0.9, 0.95)$, $\epsilon = 10^{-8}$, decay 0.01, gradient clip 1.0.
- Steps: 9,500 / 30,000 / 95,000; peak LR: 6×10^{-4} / 4×10^{-4} / 4×10^{-4}.
- NIAH: greedy decoding, at most 128 new tokens; real retrieval: 48.

Paper, App. B–C, PDF pp. 11–12, Eqs. (38)–(39), Tables 7–8.