Linear Attention Architectures

Mechanisms, Trade-offs, and Cross-Layer Routing

Paper: Tommaso Cerruti, Tim Rieder, George Rowlands,
Lingfeng Jin, and Imanol Schlag

ETH Zurich
Independent paper discussion prepared with Codex

arXiv:2607.07953v1 · 8 September 2026

Memory control, training trade-offs, and a small gain from routing values.

Versioned paper; submitted 8 July 2026. Main discussion: about 25–30 minutes, then backup material.

The paper yields three different answers

Best validation loss in the reported 350M / 15B sweep

Kimi Delta Attention + Muon + hybrid: **2.273**.

Fastest normalized training throughput in that sweep

Gated DeltaNet + AdamW + pure: **100%**, final loss **2.433**.

Cross-layer routing

CLVR lowers loss in five matched single runs, with **small gains**.

Paper, Table 2, PDF p.12; Table 7, p.16. Training speed only; no empirical inference benchmark.

Fixed memory trades explicit retrieval for compression

Softmax attention

Each query scores previous keys and forms a normalized value mixture.

- Explicit token-to-token retrieval.
- Pairwise computation grows as T^2.

Recurrent linear attention

History is compressed into a fixed-size associative memory.

- Read/write cost per token is independent of T at fixed dimensions.
- Correlated keys can interfere.

Two questions: how should memory be edited, and what should cross depth?

Paper, Secs.1–2, PDF pp.1–4. T: sequence length. Explicit matrix storage has quadratic memory cost; this is not a claim about every softmax implementation.

The memory maps key features into value space

$$\kappa^{(i)} = \phi(k^{(i)}) \in \mathbb{R}^{d_\phi}, \qquad W^{(i)} \in \mathbb{R}^{d_v \times d_\phi}$$

$$\underbrace{\bar{v}^{(i)} = W^{(i-1)}\kappa^{(i)}}_{\text{predict at current key}} \qquad \underbrace{y^{(i)} = W^{(i)}\phi(q^{(i)})}_{\text{output at current query}}$$

- $v^{(i)} \in \mathbb{R}^{d_v}$: current value target; W: dynamic state.
- Additive memory: $W^{(i)} = W^{(i-1)} + v^{(i)}\kappa^{(i)\top}$.
- Old-state key read and new-state query read have different roles.

Paper, Eqs.4–10, PDF pp.3–4. The kernel discussion uses an unnormalized read. We follow the consistent matrix orientation in Eqs.5/8; see backup note on Eq.3.

DeltaNet writes a correction to the current association

$$r^{(i)} = v^{(i)} - W^{(i-1)}\kappa^{(i)}$$

$$\boxed{W^{(i)} = W^{(i-1)} + \beta^{(i)} r^{(i)} \kappa^{(i)\top}}$$

- $\beta^{(i)} \in (0, 1)$ controls the delta update.
- A well-predicted association produces a small write.
- There is no separate gate for globally decaying stale state.

Teaching example: one unit-norm key

Prediction 2, target 5, $\beta = 0.5 \Rightarrow$ new prediction 3.5.

Paper, Sec.3.2, Eq.14, PDF p.5. Numerical example is illustrative, not an experiment.

GDN and KDA change the granularity of forgetting

Gated DeltaNet: one scalar for the whole state

$$r_\alpha = v - \alpha W_{\text{old}}\kappa, \qquad W_{\text{new}} = \alpha W_{\text{old}} + \beta r_\alpha \kappa^\top$$

Kimi Delta Attention: one decay per key-feature channel

$$D_\alpha = \text{Diag}(\boldsymbol{\alpha}), \quad r_{\boldsymbol{\alpha}} = v - W_{\text{old}} D_\alpha \kappa$$
$$W_{\text{new}} = W_{\text{old}} D_\alpha + \beta r_{\boldsymbol{\alpha}} \kappa^\top$$

D_α multiplies on the **right**. Both mechanisms still share one scalar β for active erase/write control.

Paper, Eqs.15–20, PDF pp.5–6. Token superscripts suppressed. Setting $\boldsymbol{\alpha} = \alpha\mathbf{1}$ recovers GDN.

GDN2 separates key-side erase and value-side write

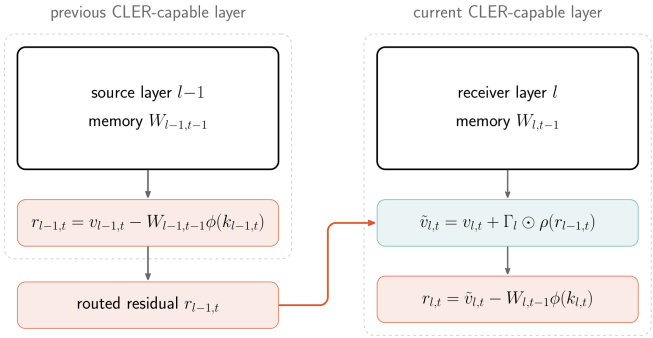
$$\widetilde{W} = W_{\mathrm{old}} D_\alpha, \qquad e = \mathbf{b} \odot \kappa, \qquad z = \mathbf{w} \odot v$$

$$\boxed{W_{\mathrm{new}} = \widetilde{W} + \underbrace{(z - \widetilde{W} e)}_{r_{\mathrm{GDN2}}} \kappa^\top}$$

- $\mathbf{b} \in (0,1)^{d_\phi}$ controls the erase direction.
- $\mathbf{w} \in (0,1)^{d_v}$ controls the write target.
- The outer product still uses $\kappa^\top$, not $e^\top$.
- $\mathbf{b} = \beta\mathbf{1}$, $\mathbf{w} = \beta\mathbf{1}$ recovers KDA.

Paper, Sec.3.5, Eqs.21–24, PDF p.6. Greater control does not guarantee better finite-budget loss or throughput.

CLER sends a local write error into another value space



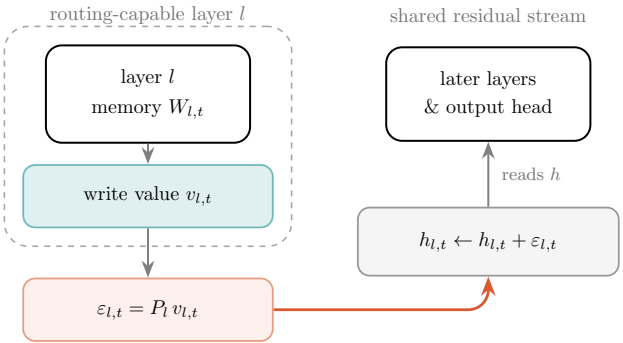
$$\tilde{v}_{l,t} = v_{l,t} + \Gamma_l r_{p(l),t}$$

- $p(l)$: nearest lower routing layer.
- Scalar Γ_l; raw residual in main tests.
- Softmax layers pass the side-channel.

Author hypothesis: independently learned value spaces may be misaligned.

Paper, Fig.1, PDF p.8; Eqs.27–28, p.7. Original figure crop. Space mismatch is an interpretation, not independently established causality.

CLVR projects the write value into the residual stream



$$\varepsilon_{l,t} = P_l v_{l,t}$$

$$h_{l,t} \leftarrow h_{l,t} + \varepsilon_{l,t}$$

- $P_l : \mathbb{R}^{d_v} \to \mathbb{R}^{d_{\text{model}}}$.
- Zero initial contribution.
- Later layers and output head use the ordinary residual path.

Paper, Fig.2, PDF p.8; Eq.29 and Sec.4.3, pp.7–9. Original crop. Adds a projection; empirical routing overhead is not tabulated.

The cleanest ablation compares value with error

Variant	Signal	Destination	Mapping
CLER	r	Later value target	Scalar Γ_l
CLER-H	r	Residual stream	Projection P_l
CLVR	v	Residual stream	Projection P_l

$$v_{l,t} = r_{l,t} + \bar{v}_{l,t}$$

- CLER-H and CLVR use **identical projection shapes/ranks**.
- Value retains the memory prediction subtracted from the error.
- CLER $\rightarrow$ CLER-H also changes parameterization; it is a less isolated test of injection space.

Paper, Secs.4.2–4.3, PDF pp.7–9; last bullet is reading analysis. Cross-table baselines also differ.

A shared recipe still contains distinct experiment groups

- **Main sweep:** 350M class / 15B tokens; FineWeb-Edu, LLaMA2 tokenizer, length 4096, global batch 128, bf16, 4 GH200 GPUs.
- **Architecture-specific depths:** 20–24 layers; recorded counts near 312M, approximately $\pm 2\%$.
- **Routing:** matched host, optimizer, schedule, seed; 1B/15B tokens at 350M, plus GDN at 1.3B/40B.
- **Critical boundary:** different data slices across routing budgets; compare gains within matched groups.

Metric and statistical scope

Final held-out cross-entropy, not the best intermediate point.
Single runs: no seed-averaged estimates or standard deviations.

Paper, Sec.5, Table 1, PDF pp.9–11; Sec.6, p.11; Appendix B, p.20.

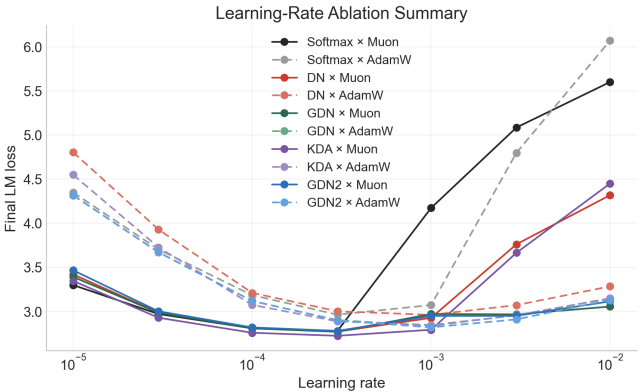
KDA achieves lower loss at lower training throughput

Configuration	Stack	Final loss ↓	Throughput ↑
Softmax / Muon	dense	2.349	79.7%
DeltaNet / Muon	hybrid	2.299	83.8%
GDN / Muon	hybrid	2.321	89.5%
KDA / Muon	hybrid	**2.273**	70.9%
GDN2 / Muon	hybrid	2.345	81.3%
GDN / AdamW	pure	2.433	**100.0%**

- Versus softmax/Muon: KDA hybrid lowers loss by 0.076, with about 11.0% lower throughput (calculated).
- Hybrid loss is lower within each pair; throughput need not fall: KDA/Muon hybrid 70.9% vs. pure 68.5%.

Paper, Table 2, PDF p.12; 350M / 15B, single runs. Hybrid: linear + softmax; pure: linear mixers only. Throughput normalized to GDN/AdamW/pure.

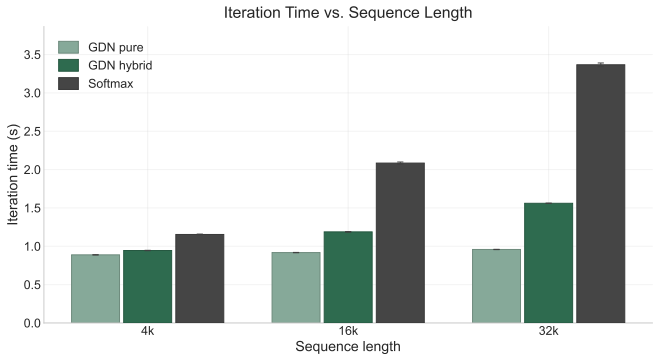
Optimizer effects must be read with learning-rate choice



- 350M hybrid, 2000 steps ($\sim$1.05B tokens).
- Solid: Muon; dashed: AdamW.
- Muon favors roughly 3×10^{-4}.
- Poor learning rates can reverse a comparison.

Paper, Fig.5, PDF p.13 (original crop), Sec.6.2, p.12. Sec.5.2 and Sec.6.2 differ in how broadly they assign AdamW's optimum to 10^{-3}; treat the individual curves as the evidence.

Pure GDN has flatter measured training iteration times



At 32k tokens:

- Softmax: 3.37 s/iteration.
- GDN hybrid: 1.56 s.
- GDN pure: 0.96 s.

Softmax/pure time ratio: **3.51** (calculated).

Per-length token/batch control is not fully specified in the text.

This is a training measurement; inference speed is unmeasured.

Paper, Fig.6, PDF p.14, Sec.6.3, p.12. Original crop. Error bars show step-to-step variation, not training-seed uncertainty.

Larger DeltaNet runs remain learning-rate sensitive

Params / tokens	Configuration	Learning rate	Final loss
1.3B / 40B	DN hybrid	3.6×10^{-4}	2.066
1.3B / 40B	DN pure	1.5×10^{-4}	2.063
1.3B / 40B	DN pure + CLER	1.5×10^{-4}	2.112
3B / 60B	DN hybrid	5×10^{-4}	2.332
3B / 60B	DN hybrid	3×10^{-4}	1.955
3B / 60B	DN hybrid	1.5×10^{-4}	1.955

- A larger-scale DeltaNet study, not an all-architecture sweep.
- The CLER row is not a CLVR result.

Paper, Table 3, PDF p.14; selected rows. Optimizer is not listed row-by-row in this table.

Original CLER produces no reliable validation gain

Optimizer	Host	Baseline	CLER	Reported Δ
Muon	GDN	2.8319	2.8333	$+0.0013^*$
Muon	DeltaNet	2.8511	2.8507	-0.0004
AdamW	GDN	3.2554	3.2562	$+0.0008$
AdamW	DeltaNet	3.4068	3.4258	$+0.0190$

- 350M / approximately 1B tokens; matched per host and optimizer.
- The only favorable difference is below 10^{-3} in one run.
- Nonzero routed residuals do not establish a useful causal effect.

Paper, Table 6, Sec.6.5, PDF p.15. Negative Δ is better. *Paper reports $+0.0013$; displayed losses subtract to $+0.0014$. Retained as a rounding/reporting discrepancy.

CLVR improves all five matched rows by small amounts

Host	Params / tokens	Baseline	CLER-H Δ	CLVR Δ
GDN	350M / 1B	2.8331	−0.0073	**−0.0103**
GDN	350M / 15B	2.3417	−0.0042	**−0.0059**
GDN	1.3B / 40B	2.0635	−0.0010	**−0.0019**
DeltaNet	350M / 1B	2.8469	−0.0047	**−0.0119**
DeltaNet	350M / 15B	2.3347	−0.0002	**−0.0016**

- Muon; final validation loss; negative Δ is better.
- Value beats error at equal projection shape; gains diminish in the longer/larger reported runs.
- Single matched runs; no larger DeltaNet hidden-stream row.

Paper, Table 7, PDF p.16. Compare within each row. Do not apply these deltas to Table 2 baselines; data slices/run groups differ.

Controls support a signal hypothesis, with limited evidence

- **Exploratory 350M/1B group:**
  plain CLVR 2.8152; surprise gate 2.8220; error+value 2.8194.
- **Capacity controls, one seed:**
  hidden-state routing $\Delta \approx -0.0014$; larger baseline $+0.0021$.
- **Attention Residuals, 350M/15B:**
  DeltaNet $\Delta = -0.0101$ (larger gain than CLVR); GDN $+0.0030$.

Evidence boundary

These controls do not establish statistical significance or universal superiority. The exploratory 2.8152 reference is a separate group from Table 7.

Paper, Sec.6.6, PDF pp.16–17; Appendix B, p.20. Attention Residuals numbers occur in prose, not as Table 7 rows.

Downstream checks show mixed, small task differences

Host	Variant	HS acc_norm	PIQA acc	WG acc
GDN	Baseline	0.4131	0.6692	0.5146
GDN	CLVR	0.4113	0.6719	0.5328
DeltaNet	Baseline	0.4113	0.6600	0.5193
DeltaNet	CLVR	0.4104	0.6616	0.5359

- Matched 350M/15B/Muon, zero-shot, full-sequence scoring.
- HS: HellaSwag; WG: WinoGrande; higher is better.
- GDN/CLVR WG is +1.82 percentage points (calculated), but a single checkpoint does not establish a routing effect.

Paper, Table 5, PDF p.15; selected columns. Authors use these tasks to check for clear degradation. Table 4 uses a separate evaluation/run group.

The paper's strongest boundaries are empirical

1. No repeated training seeds; small loss differences are suggestive.
2. Uneven hyperparameter coverage; architecture and tuning interact.
3. Routing tested only on DeltaNet and GDN; larger hidden-stream routing only on GDN.
4. No long-context retrieval/QA study and no inference benchmark.
5. Space mismatch is a plausible explanation; CLER and CLER-H also differ in parameterization.

Paper, Secs.5–8 and Appendix B, PDF pp.10–11, 17–18, 20; final item includes reading analysis from Eqs.27/29.

Useful next tests separate content, space, and scale

- Repeat matched seeds and longer budgets to measure uncertainty.
- Try KDA/GDN2 hosts; distinguish raw value, gated target, and residual.
- Measure associative recall, passkeys, long-context QA, decoding memory and throughput.
- **Reading proposal:** hold the projection budget fixed while changing injection space; intervene by disabling routing after training.

Take-home message

Choose memory control, stack, optimizer, and learning rate together.
CLVR is a simple, promising signal route with modest current evidence.

Paper, Secs.8–9, PDF pp.17–18. The proposed matched-space and post-training interventions are analysis suggestions, not reported experiments.

Paper and reproducibility trail

Tommaso Cerruti, Tim Rieder, George Rowlands, Lingfeng Jin, and Imanol Schlag.

Linear Attention Architectures: Mechanisms, Trade-offs, and Cross-Layer Routing.

arXiv:2607.07953v1 Submitted 8 July 2026.

Author-linked implementation and experiment records

This reading covers the 20-page paper and Appendices A/B.
It does not reproduce training or audit the repository.

Paper, Sec.6.7, PDF p.17. All displayed paper figures are original PDF crops; page locators refer to the paper, not this deck.

Backup: the delta rule is a local gradient step

For fixed κ and v, consider the teaching objective

$$\ell(W) = \tfrac{1}{2}\|W\kappa - v\|_2^2, \qquad \nabla_W \ell = (W\kappa - v)\kappa^\top.$$

Then $W_{\text{new}} = W_{\text{old}} - \beta \nabla_W \ell$ is Eq.14.

$$\|\kappa\|_2 = 1 \quad \Longrightarrow \quad W_{\text{new}}\kappa = (1-\beta) W_{\text{old}}\kappa + \beta v.$$

For another direction u:

$$(W_{\text{new}} - W_{\text{old}})u = \beta(v - W_{\text{old}}\kappa)(\kappa^\top u).$$

Teaching derivation from Paper Eq.14, PDF p.5. The unit-norm assumption is explicit; this local objective is not the outer language-model training loss.

Backup: full AdamW architecture rows

Architecture	Stack	Speed	Saturation	Final
Softmax	dense	81.7%	2.587	2.413
DeltaNet	hybrid	87.0%	2.517	2.348
DeltaNet	pure	91.9%	2.555	2.382
GDN	hybrid	90.5%	2.517	2.356
GDN	pure	100.0%	2.601	2.433
KDA	hybrid	74.4%	2.471	2.328
KDA	pure	73.6%	2.490	2.347
GDN2	hybrid	83.3%	2.520	2.379
GDN2	pure	81.9%	2.601	2.452

350M class / 15B tokens. Speed: normalized training throughput.
Lower loss and higher speed are better; each row is one run.

Paper, Table 2, PDF p.12. DN = DeltaNet, GDN = Gated DeltaNet, KDA = Kimi Delta Attention, GDN2 = Gated DeltaNet-2.

Backup: full Muon architecture rows

Architecture	Stack	Speed	Saturation	Final
Softmax	dense	79.7%	2.489	2.349
DeltaNet	hybrid	83.8%	2.440	2.299
DeltaNet	pure	89.2%	2.476	2.334
GDN	hybrid	89.5%	2.464	2.321
GDN	pure	95.0%	2.543	2.397
KDA	hybrid	70.9%	2.409	2.273
KDA	pure	68.5%	2.427	2.290
GDN2	hybrid	81.3%	2.466	2.345
GDN2	pure	80.7%	2.542	2.415

350M class / 15B tokens. Speed: normalized training throughput.
Lower loss and higher speed are better; each row is one run.

Paper, Table 2, PDF p.12. DN = DeltaNet, GDN = Gated DeltaNet, KDA = Kimi Delta Attention, GDN2 = Gated DeltaNet-2.

Backup: early 1B baselines are a separate comparison

Architecture	Optimizer	Val loss	Val PPL	ktok/s/GPU
Softmax	AdamW	2.9019	18.2095	144.8
Softmax	Muon	2.7240	15.2410	136.7
GDN	AdamW	2.8562	17.3950	159.5
GDN	Muon	2.7097	15.0254	154.8
DeltaNet	AdamW	2.9391	18.8987	153.4
DeltaNet	Muon	2.7137	15.0847	146.7

- Earlier 350M / approximately 1B-token runs.
- Useful optimizer context, not the final 15B frontier.
- Do not use these losses as Table 6 or Table 7 baselines.

Paper, Table 8 and Appendix A, PDF p.20. Throughput is in thousands of training tokens per second per GPU, without normalization.

Backup: projection cost and initialization

$$P_l \in \mathbb{R}^{d_{\text{model}} \times d_v}, \qquad P_l = U_l D_l \quad \text{(optional rank } d_p\text{)}$$

- Full-rank parameter count per layer: $d_{\text{model}} d_v$.
- Low-rank count: $d_p(d_{\text{model}} + d_v)$.
- At fixed width/rank, added work is linear in sequence length.
- A hybrid host still retains its softmax-layer costs.
- For a factorized projection, zeroing both factors blocks ordinary gradients; zero the product with a trainable path.

Paper, Eq.29, PDF p.7. Counts and gradient observation are teaching analysis. Exact per-run rank, added parameter counts, and measured routing overhead are not tabulated in the paper.

Backup: keep formula conventions and run groups explicit

- **Eq.3, p.3:** multiplication order conflicts with $W \in \mathbb{R}^{d_v \times d_\phi}$. Eqs.5/8 give the consistent read $W\phi(q)$.
- **GDN residual:** Sec.3.3 uses a decay-adjusted read; Sec.4 writes a simplified ungated expression. Verify the actual internal signal before implementation.
- **Initial state:** zero P_l means zero added route, not a zero host memory or zero network parameters.
- **Baselines:** Table 2, Table 6, Table 7, and the exploratory controls are separate run groups.

Reading audit of Paper Eqs.3–5, 15–17, 25–29; Tables 2, 6–7; Sec.6.6. This is not a code audit.