FG^2-GDN: Enhancing Long-Context Gated Delta Networks with Doubly Fine-Grained Control

Channel-wise writing rates for recurrent attention memory

Paper by Pingwei Sun et al. (Meituan)

arXiv:2604.19021v3 | November 6, 2026

Published at EMNLP 2026 Official code: not found (checked 2026-11-06)

The paper gives the write gate the same granularity as forgetting

Research question

KDA learns a separate forgetting rate for every channel, but still shares one delta-rule writing rate across all channels. Can writing become channel-wise without losing KDA's efficient chunkwise algorithm?

- FG^2-GDN predicts a channel-wise β_t and uses it to scale keys and values symmetrically.
- FG^2-GDN+ adds separate key and value rates, β_t^k and β_t^v.
- At 1.3B, FG^2-GDN leads the language-modeling average; FG^2-GDN+ leads aggregate RULER and LongBench.

Paper, Abstract and Secs. 1, 4, PDF pp. 1, 5–9.

KDA refines forgetting; FG^2-GDN refines writing

Method	Forgetting rate α_t	Writing rate β_t
GDN	Scalar	Scalar
KDA	Channel-wise vector	Scalar
FG^2-GDN	Channel-wise vector	Channel-wise vector
FG^2-GDN+	Channel-wise vector	Separate channel-wise β_t^k, β_t^v

- α_t controls how much old state is retained.
- β_t controls the strength of the current delta update.
- The contribution is an orthogonal design axis to KDA's channel-wise decay.

Paper, Secs. 2.2–3.1, PDF pp. 2–3.

The delta rule corrects a fast associative memory

A linear-attention state stores key–value associations and is read by the query:

$$o_t = S_t^\top q_t, \qquad S_t \in \mathbb{R}^{d_k \times d_v}.$$

The delta update takes one gradient step on the reconstruction loss

$$\mathcal{L}_t(S) = \tfrac{1}{2}\|S^\top k_t - v_t\|^2, \qquad S_t = (I - \beta_t k_t k_t^\top) S_{t-1} + \beta_t k_t v_t^\top.$$

- $S_{t-1}^\top k_t$ is the memory's current prediction for key k_t.
- The rank-1 correction moves that prediction toward v_t.
- Gating adds data-dependent forgetting so the state can discard old associations.

Paper, Sec. 2.1, PDF p. 2, Eqs. (5)–(7).

FG^2-GDN makes the writing rate channel-wise

For each head, let $\alpha_t, \beta_t \in (0,1)^{d_k}$ and define

$$\tilde{k}_t = \sqrt{\beta_t} \odot k_t, \qquad \tilde{v}_t = \sqrt{\beta_t} \odot v_t.$$

$$S_t = (I - \tilde{k}_t \tilde{k}_t^\top)\,\mathrm{Diag}(\alpha_t) S_{t-1} + \tilde{k}_t \tilde{v}_t^\top, \qquad o_t = S_t^\top q_t.$$

- α_t still chooses what to forget in each key dimension.
- β_t now chooses how strongly each dimension participates in the update.
- In the model, a low-rank projection predicts β_t from the token representation.

Paper, Secs. 3.2 and 3.5, PDF pp. 3–4, Eqs. (10)–(11).

Square-root scaling preserves KDA's tied rank-1 correction

Naive vector replacement

$$\mathrm{Diag}(\beta_t)k_t k_t^\top = (\beta_t \odot k_t)k_t^\top$$

Left and right factors differ, so the symmetric tied form used by KDA's specialized WY-based algorithm is lost.

FG^2-GDN

$$\tilde{k}_t \tilde{k}_t^\top = \mathrm{Diag}(\sqrt{\beta_t})k_t k_t^\top \mathrm{Diag}(\sqrt{\beta_t})$$

The correction remains symmetric rank-1, so KDA's chunkwise structure is retained.

Important interpretation

Channel-wise does not mean an unconstrained learning rate for every matrix entry: pairwise channel effects remain coupled by the rank-1 factorization.

Paper, Sec. 3.2–3.3, PDF p. 3, Eqs. (10)–(12).

FG^2-GDN+ separates erasing from writing values

$$\hat{k}_t = \sqrt{\beta_t^k} \odot k_t, \qquad \tilde{v}_t = \sqrt{\beta_t^v} \odot v_t,$$
$$S_t = (I - \hat{k}_t \hat{k}_t^\top) \operatorname{Diag}(\alpha_t) S_{t-1} + \hat{k}_t \tilde{v}_t^\top.$$

- β_t^k controls erasure and the key-side factor of the write.
- β_t^v independently scales the value-side write.
- The extra gate projection adds a small parameter/FLOP cost while preserving the update structure.

Paper, Sec. 3.4–3.5, PDF p. 4, Eq. (13).

The channel-wise gate reuses KDA's chunkwise training path

- Expand the transition as $\mathrm{Diag}(\alpha_t) + a_t b_t^\top$, with $a_t = -\tilde{k}_t$ and $b_t = \tilde{k}_t \odot \alpha_t$.
- The tied diagonal-plus-rank-1 form allows the same WY representation and UT transform used by KDA.
- For sequence length L, chunk size C, head dimension d:

$$O(Ld^2 + LCd + LC^2).$$

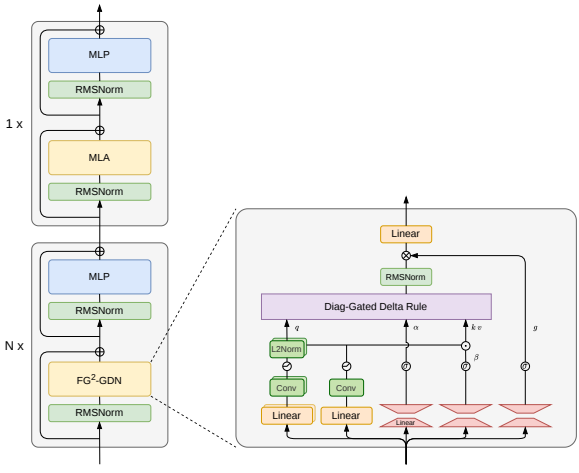
- Added elementwise scaling is $O(Ld)$, estimated at about 0.8% of dominant state-update arithmetic for $d_k = d_v = 128$.

Measured overhead

FG^2-GDN prefill throughput is within 5% of KDA in all tested settings; the separate value gate adds a little more cost.

Paper, Secs. 3.6 and 4.5, PDF pp. 4, 7–8.

A 3:1 linear-attention/MLA hybrid tests the memory cell



Paper, Fig. 1, PDF p. 5. Three linear-attention layers are followed by one MLA layer.

Experiments hold the backbone and training recipe fixed

Two model scales

- 340M: 20 layers, hidden size 1024, 15B tokens.
- 1.3B: 24 layers, hidden size 2048, 100B tokens.
- 16 heads; linear-attention head dimension 128.

Shared protocol

- 3:1 linear attention to MLA; same SlimPajama data and schedule.
- 8K training length; BF16, AdamW, cosine schedule.
- Peak LR 3×10^{-4}; each configuration trained once.

- Language modeling: LAMBADA perplexity plus seven zero-shot accuracy scores.
- Long context: RULER at 4K/8K/16K and LongBench, both at 1.3B.

Paper, Sec. 4 and Appendix A, PDF pp. 5, 11.

Channel-wise writing improves the language-modeling average

Scale	Method	Accuracy avg. ↑	LAMBADA PPL ↓
340M	GDN	43.51	39.86
	KDA	43.14	41.30
	FG^2-GDN	**44.88**	38.27
	FG^2-GDN+	44.84	**37.93**
1.3B	GDN	52.79	14.01
	KDA	53.19	13.80
	FG^2-GDN	**53.95**	**13.09**
	FG^2-GDN+	53.40	13.91

- The mean is over seven task accuracies; perplexity is reported separately.
- At 1.3B, FG^2-GDN is +0.76 percentage points over KDA on mean accuracy.

Paper, Table 1, PDF p. 6.

FG^2-GDN+ leads RULER at all tested lengths

1.3B model	4K	8K	16K
GDN	61.6	51.0	42.4
KDA	61.5	55.0	44.9
FG^2-GDN	64.8	57.3	44.6
FG^2-GDN+	**67.2**	**61.4**	**48.9**

- Scores are average accuracy (%) over 13 RULER task categories.
- Models train at 8K; 16K evaluates extrapolation beyond training length.
- Some individual tasks remain non-monotonic across methods.

Paper, Sec. 4.2 and Table 2, PDF pp. 5–6.

Separating key and value control lifts LongBench overall

1.3B model	LongBench average (%) ↑
GDN	15.1
KDA	16.4
FG^2-GDN	16.0
FG^2-GDN+	**18.3**

Read the ablation, not only the final score

FG^2-GDN improves language modeling but is slightly below KDA on LongBench. The full FG^2-GDN+ variant gives the strongest aggregate long-context score; code-completion subtasks still favor KDA.

Paper, Sec. 4.2 and Table 3, PDF pp. 5, 7.

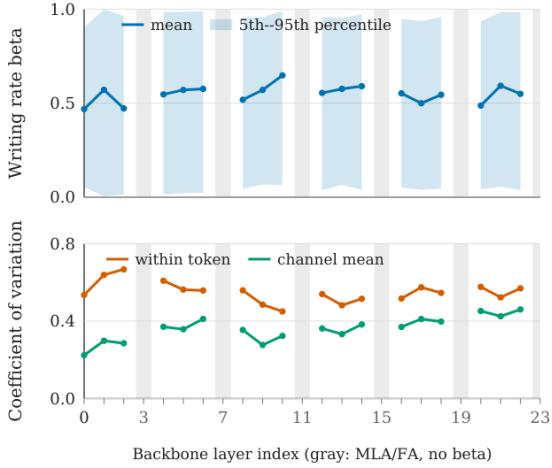
Ablations isolate both sources of gain

Variant	LM avg. ↑	PPL ↓	LongBench ↑
GDN	52.79	14.01	15.1
Scalar α, vector β (FGBGDN)	53.25	13.96	16.8
KDA (vector α)	53.19	13.80	16.4
FG^2-GDN (vector α, β)	**53.95**	**13.09**	16.0
Vector α, separate scalar $\beta^{k/v}$	53.9	13.56	16.7
FG^2-GDN+ (both refinements)	53.40	13.91	**18.3**

- Vector β helps over GDN and further improves language modeling over KDA.
- Key/value separation boosts LongBench, especially when combined with vector β.

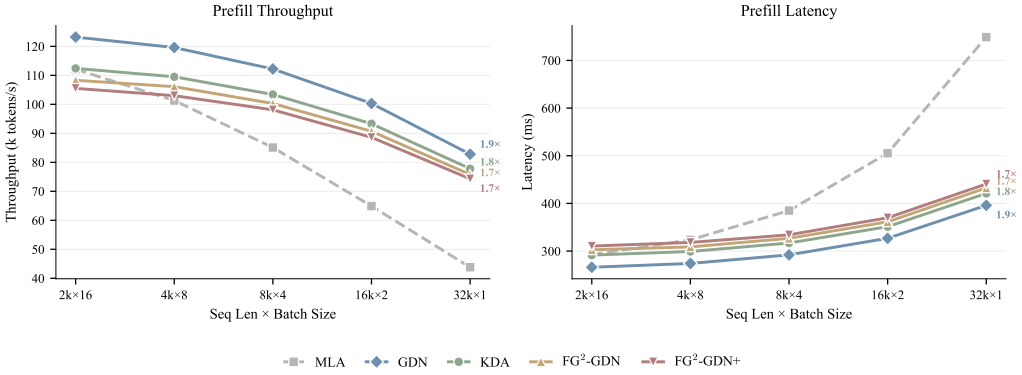
Paper, Sec. 4.4 and Table 4, PDF p. 7.

The learned writing rate varies across channels



Paper, Sec. 4.3 and Fig. 2, PDF p. 7. 1,408 held-out tokens; 18 linear-attention layers.

Long prefill keeps the linear-attention speed advantage



Paper, Sec. 4.5 and Fig. 3, PDF p. 8. H800-80G, BF16; 32,768 total tokens per setting.

Evidence is promising at 340M–1.3B; deployment remains open

What the evidence supports

- Channel-wise β adds a useful writing-control axis to KDA's channel-wise decay.
- FG^2-GDN+ has the best aggregate RULER and LongBench scores reported.
- The structured update preserves chunkwise training and competitive prefill speed.

Authors' stated limits

- Dense hybrids at 340M and 1.3B; one pretraining run per setting.
- No 7B+, MoE, seed variance, per-method tuning, or KDA warm start.
- Prefill is measured; serving and speculative-decoding integration remain open.

Paper, Conclusion and Limitations, PDF p. 9.

Takeaways

1. KDA makes forgetting channel-wise; FG^2-GDN also makes writing channel-wise.
2. Symmetric square-root scaling retains a tied rank-1 transition and KDA's chunkwise algorithm.
3. Decoupling β^k from β^v gives the strongest aggregate long-context results in this study.
4. The gains are demonstrated in 340M/1.3B hybrid pretraining, with measured prefill efficiency on H800.

One-sentence summary

Fine-grained write control improves what the recurrent memory can learn per token while preserving its efficient update structure.

Paper, Sec. 6, PDF p. 9.

Paper and publication

- Pingwei Sun et al. 2026. "FG^2-GDN: Enhancing Long-Context Gated Delta Networks with Doubly Fine-Grained Control."
- Read version: arXiv:2604.19021v3.
- Published in Proceedings of EMNLP 2026, pp. 51013–51023.
- Official/public implementation matching the paper: not found as of 2026-11-06.

Related methods cited by the paper: DeltaNet; Gated DeltaNet; Kimi Delta Attention; RULER; LongBench.