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

Anonymous ACL submission

Abstract

Linear attention mechanisms have emerged as promising alternatives to softmax attention, offering linear-time complexity during inference. Recent advances such as Gated DeltaNet (GDN) and Kimi Delta Attention (KDA) have demonstrated that the delta rule—an online gradient descent update—enables superior associative recall compared to simple additive updates. While KDA refined the coarse head-wise decay gate into channel-wise decay, the learning rate β_t in the delta update remains a scalar, limiting the model's capacity for dimension-specific adaptation. We introduce **FG^2-GDN**, which replaces the scalar β_t with a channel-wise vector—analogous to the transition from SGD to per-coordinate adaptive optimizers such as AdaGrad and Adam. We further propose **FG^2-GDN+**, which decouples the scaling for keys and values, enabling independent control of erasure strength and write strength. Experiments on synthetic and real-world benchmarks show that FG^2-GDN and its variant improve associative recall and long-context understanding over GDN and KDA, with comparable computational efficiency.

1 Introduction

The Transformer architecture has revolutionized natural language processing through its powerful attention mechanism, enabling models to capture long-range dependencies effectively. However, the quadratic complexity of softmax attention with respect to sequence length poses significant challenges for both training efficiency and inference scalability, particularly for long-context applications such as agentic and multimodal scenarios.

Linear attention mechanisms (Katharopoulos et al., 2020) address this limitation by reformulating attention as a linear recurrence with matrix-valued hidden states, reducing inference complexity from $O(L^2)$ to $O(L)$. Early variants of linear attention underperformed softmax attention, but recent innovations have substantially narrowed this gap. Two key developments have driven this progress: (1) *gating mechanisms* that enable adaptive memory decay (Hua et al., 2022; Peng et al., 2023; Yang et al., 2024a), and (2) the *delta rule* that facilitates precise memory updates through online gradient descent (Schlag et al., 2021; Yang et al., 2024c).

The delta rule treats the recurrent state as a learnable associative memory that continuously corrects itself toward the target mapping $k_t \mapsto v_t$. Gated DeltaNet (GDN) (Yang et al., 2024a) subsequently unified the delta rule with gating mechanisms, introducing a scalar decay factor $\alpha_t \in (0, 1)$ that enables controlled memory forgetting.

More recently, Kimi Delta Attention (KDA) (Zhang et al., 2025) refined GDN's coarse head-wise decay into channel-wise decay through a diagonal gating matrix $\mathrm{Diag}(\alpha_t)$, which enables each feature dimension to maintain independent forgetting rates. KDA demonstrated that such fine-grained control significantly improves long-context modeling and associative recall.

Our Contribution. In this work, we propose a complementary improvement to the KDA framework. While KDA introduced channel-wise decay gates, the learning rate β_t in the delta update remains a scalar shared across all dimensions. This design choice limits the model's ability to perform dimension-specific adaptation.

We propose **FG^2-GDN**, which extends KDA by introducing fine-grained adaptive learning rates. Specifically, FG^2-GDN replaces the scalar $\beta_t \in \mathbb{R}$ with a vector $\beta_t \in \mathbb{R}^{d_k}$, enabling each dimension to maintain independent adaptation dynamics. The update rule transforms as follows:

$$\begin{aligned} S_t &= (I - \tilde{k}_t \tilde{k}_t^\top)\mathrm{Diag}(\alpha_t) S_{t-1} + \tilde{k}_t \tilde{v}_t^\top \\ o_t &= S_t^\top q_t \end{aligned} \tag{1}$$

where $\tilde{k}_t = \sqrt{\beta_t} \odot k_t$ and $\tilde{v}_t = \sqrt{\beta_t} \odot v_t$ are

channel-wise scaled versions of the key and value vectors.

From an online learning perspective, this modification corresponds to employing per-coordinate learning rates in stochastic gradient descent, analogous to the distinction between SGD and per-coordinate adaptive methods such as AdaGrad (Duchi et al., 2011) and Adam (Kingma and Ba, 2014).

Experiments at the 340M and 1.3B scales show that FG^2-GDN improves long-context retrieval on RULER and LongBench while maintaining competitive language modeling quality. Since the update rule preserves a variant of the Diagonal-Plus-Low-Rank (DPLR) structure (Gu et al., 2021), our method can be easily adapted to existing chunkwise parallel implementations (Yang and Zhang, 2024), incurring less than 5% prefill throughput overhead relative to KDA.

2 Preliminary

2.1 Linear Attention and the Delta Rule

Linear attention (Katharopoulos et al., 2020) reformulates the attention mechanism by replacing the exponential kernel in softmax attention with a dot-product kernel. Given query, key, and value vectors $q_t, k_t, v_t \in \mathbb{R}^d$, the output is computed as:

$$o_t = \sum_{i=1}^{t} \frac{\phi(q_t)^\top \phi(k_i) v_i}{\sum_{i=1}^{t} \phi(q_t)^\top \phi(k_i)} \tag{2}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^n$ is a feature map. By setting ϕ to the identity mapping and omitting the normalization denominator, we obtain the simplified linear attention recurrence:

$$S_t = S_{t-1} + v_t k_t^\top, \quad o_t = S_t^\top q_t \tag{3}$$

where $S_t \in \mathbb{R}^{d \times d}$ is the matrix-valued hidden state that accumulates key-value associations.

Online Learning Interpretation. From the fast-weight perspective (Schmidhuber, 1992; Irie et al., 2021), S_t serves as an associative memory storing transient mappings from keys to values. The update in Eq. 3 can be viewed as performing gradient descent on the unbounded correlation objective:

$$\mathcal{L}_t(S) = -\langle S^\top k_t, v_t \rangle \tag{4}$$

which continually reinforces recent key-value pairs without any forgetting mechanism. This leads to unbounded state growth and interference over long contexts.

DeltaNet. DeltaNet (Schlag et al., 2021) addresses this limitation by interpreting the recurrence as online gradient descent on a reconstruction objective:

$$\mathcal{L}_t(S) = \frac{1}{2} \| S^\top k_t - v_t \|^2 \tag{5}$$

Taking a gradient step with learning rate β_t yields the delta update rule:

$$\begin{aligned} S_t &= S_{t-1} - \beta_t \nabla_S \mathcal{L}_t(S_{t-1}) \\ &= (I - \beta_t k_t k_t^\top) S_{t-1} + \beta_t k_t v_t^\top \end{aligned} \tag{6}$$

This rule treats S as a learnable associative memory that continuously corrects itself toward the mapping $k_t \mapsto v_t$. The rank-1 update structure, equivalent to a generalized Householder transformation, supports hardware-efficient chunkwise parallelization.

2.2 Delta Rule with Forgetting Gates

DeltaNet retains all past associations indefinitely, which can lead to interference over long contexts. Gated DeltaNet (GDN) (Yang et al., 2024a) addresses this by introducing a scalar forget gate $\alpha_t \in (0, 1)$:

$$S_t = \alpha_t (I - \beta_t k_t k_t^\top) S_{t-1} + \beta_t k_t v_t^\top \tag{7}$$

where α_t acts as a data-dependent decay on the hidden state, providing explicit control over memory lifespan. In GDN, both α_t and β_t are scalars shared across all feature dimensions: every dimension forgets at the same rate and writes with the same strength.

Kimi Delta Attention (KDA) (Zhang et al., 2025) refines this design by replacing the scalar α_t with a channel-wise decay vector $\alpha_t \in \mathbb{R}^{d_k}$:

$$S_t = (I - \beta_t k_t k_t^\top) \mathrm{Diag}(\alpha_t) S_{t-1} + \beta_t k_t v_t^\top \tag{8}$$

where $\mathrm{Diag}(\alpha_t) \in \mathbb{R}^{d_k \times d_k}$ is the diagonal matrix with α_t on its diagonal. This allows each feature dimension to maintain an independent forgetting rate, giving the model finer control over which information to retain or discard. However, the learning rate β_t in the delta update remains a scalar—all dimensions still share the same writing strength.

Relation to DPLR. Both GDN and KDA can be viewed as constrained variants of the Diagonal-Plus-Low-Rank (DPLR) transition structure (Gu et al., 2021):

$$S_t = (D_t - a_t b_t^\top) S_{t-1} + k_t v_t^\top \tag{9}$$

where $D_t = \mathrm{Diag}(\alpha_t)$, $a_t = \beta_t k_t$, and $b_t = k_t \odot \alpha_t$. This constrained form makes efficient chunkwise parallel training possible via the WY representation of generalized Householder products (Yang et al., 2024c; Zhang et al., 2025).

2.3 Per-Coordinate Adaptive Learning Rates

The idea of assigning different learning rates to different parameters has a long history in optimization. AdaGrad (Duchi et al., 2011) accumulates squared gradients to scale each coordinate inversely by its historical magnitude, enabling faster progress along infrequent-gradient dimensions. Adam (Kingma and Ba, 2014) further combines per-coordinate scaling with momentum, and has become the default optimizer for training deep networks. A common insight underlying these methods is that a single global step size is often suboptimal when different coordinates of the parameter space exhibit different curvatures or update frequencies. Per-coordinate adaptation can yield faster and more stable convergence in such settings. This observation motivates our extension of the delta rule's scalar learning rate to a channel-wise vector, as described in the next section.

3 Gated Delta Net with Fine-grained Beta

3.1 Motivation

As discussed in §2.3, per-coordinate adaptive learning rates have proven broadly effective in optimization. A similar asymmetry exists in the gated delta rule: KDA introduced channel-wise forgetting ($\alpha_t \in \mathbb{R}^{d_k}$) but retained a scalar writing rate ($\beta_t \in \mathbb{R}$). This means all dimensions share the same learning rate during each delta update, regardless of their individual relevance to the current context. We propose to close this gap by extending β_t to a channel-wise vector as well.

3.2 FG^2-GDN Update Rule

A straightforward way to introduce per-dimension learning rates would be to replace $\beta_t k_t k_t^\top$ in Eq. 8 with $\mathrm{Diag}(\beta_t)\, k_t k_t^\top$, where $\beta_t \in \mathbb{R}^{d_k}$. However, this asymmetric form, where β_t scales only the rows of the outer product, breaks the rank-1 Householder structure that underlies the efficient WY-based parallel algorithm used by GDN and KDA. A similar asymmetric design appears in RWKV-7 (Peng et al., 2025), whose update $\mathrm{Diag}(\beta_t)\, \hat{\kappa}\, \hat{\kappa}^\top$ requires a different parallelization strategy.

To preserve compatibility with the efficient chunkwise parallel framework, we instead absorb β_t symmetrically into both k_t and v_t. Concretely, we define the scaled key and value as

$$\tilde{k}_t = \sqrt{\beta_t} \odot k_t, \quad \tilde{v}_t = \sqrt{\beta_t} \odot v_t \tag{10}$$

and write the FG^2-GDN update as:

$$S_t = (I - \tilde{k}_t \tilde{k}_t^\top)\, \mathrm{Diag}(\alpha_t)\, S_{t-1} + \tilde{k}_t \tilde{v}_t^\top \tag{11}$$

where $\odot$ denotes element-wise multiplication and $\sqrt{\cdot}$ is applied element-wise. The delta subtraction term is now $\tilde{k}_t \tilde{k}_t^\top = \mathrm{Diag}(\sqrt{\beta_t})\, k_t k_t^\top\, \mathrm{Diag}(\sqrt{\beta_t})$, a *symmetric* rank-1 matrix that remains a valid generalized Householder reflector. This ensures that the chunkwise parallel algorithm of KDA can be reused (see §3.5), while each dimension obtains a distinct effective learning rate.

3.3 Online Learning Interpretation

Following the online learning view of DeltaNet (Schlag et al., 2021), the delta update can be seen as a gradient step on the reconstruction loss $\mathcal{L}_t(S) = \frac{1}{2}\|S^\top k_t - v_t\|^2$. In KDA, this step uses a single scalar β_t as the learning rate. FG^2-GDN generalizes this to per-coordinate learning rates:

$$\begin{aligned} S &\leftarrow S - \eta_t \odot (k_t k_t^\top S - k_t v_t^\top) \\ \eta_t &= \sqrt{\beta_t}\sqrt{\beta_t}^\top, \quad \eta_t \in R^{d \times d} \end{aligned} \tag{12}$$

Combined with the decay term $\mathrm{Diag}(\alpha_t)$, this yields Eq. 11. Due to the symmetric parameterization described above, the per-coordinate learning rate matrix takes a low-rank form $\mathrm{Diag}(\sqrt{\beta_t})\, k_t k_t^\top\, \mathrm{Diag}(\sqrt{\beta_t})$ rather than the full diagonal $\mathrm{Diag}(\beta_t)\, k_t k_t^\top$. While this constrains the effective learning rates to be coupled through the rank-1 structure of $k_t k_t^\top$, it preserves the efficient form needed for chunk-wise parallel training, which is a trade-off between representational capacity and computational efficiency mentioned by KDA.

3.4 FG^2-GDN+ Variant

We further consider a variant, **FG^2-GDN+**, that decouples the channel-wise scaling for keys and values by introducing separate vectors $\beta_t^k, \beta_t^v \in \mathbb{R}^{d_k}$:

$$\begin{aligned} S_t &= (I - \hat{k}_t \hat{k}_t^\top)\, \mathrm{Diag}(\alpha_t)\, S_{t-1} + \hat{k}_t \tilde{v}_t^\top \\ \hat{k}_t &= \sqrt{\beta_t^k} \odot k_t, \quad \tilde{v}_t = \sqrt{\beta_t^v} \odot v_t \end{aligned} \tag{13}$$

In FG^2-GDN, the same β_t scales both k_t and v_t, tying the "erasure strength" and "write strength" together. FG^2-GDN+ relaxes this constraint, allowing the model to control how aggressively it erases old information (via β_t^k) independently of how strongly it writes new values (via β_t^v). This adds a modest number of extra parameters (one additional projection for β_t^v) while preserving the same update structure.

3.5 Chunkwise Parallel Training

A key practical advantage of FG^2-GDN is that its update rule retains the DPLR form. Specifically, Eq. 11 can be rewritten as

$$S_t = (\mathrm{Diag}(\alpha_t) + a_t b_t^\top)\, S_{t-1} + \tilde{k}_t \tilde{v}_t^\top \qquad (14)$$

where $a_t = -\tilde{k}_t$ and $b_t = \tilde{k}_t \odot \alpha_t$. Since a_t and b_t are both derived from $\tilde{k}_t = \sqrt{\beta_t} \odot k_t$, the transition remains a diagonal-plus-rank-1 matrix, the same structural class as KDA.

This structural equivalence means that FG^2-GDN can directly reuse previous chunkwise parallel algorithm, which splits the sequence into chunks of size C and applies the WY representation (Bischof and Van Loan, 1987) together with the UT transform (Joffrain et al., 2006) to efficiently compute products of generalized Householder matrices within each chunk. The only difference is that the scalar β_t in FLA's implementation (Yang and Zhang, 2024) is replaced by the vector $\beta_t \in \mathbb{R}^{d_k}$ when constructing the scaled keys $\tilde{k}_t$. All subsequent matrix operations (the triangular solve for the UT factor, the inter-chunk state propagation, and the intra-chunk causal dot-product) remain identical in form and complexity.

As a result, FG^2-GDN inherits the same $O(LCd + LC^2)$ time complexity per chunk as previous methods, and the additional overhead from channel-wise β_t is limited to the element-wise scaling of k_t and v_t, which is negligible relative to the matrix multiplications that dominate runtime.

3.6 Hybrid Architecture

Purely linear attention models are limited by their finite-state capacity (Gu and Dao, 2024). Following recent hybrid designs (Yang et al., 2024a; Zhang et al., 2025), we interleave FG^2-GDN layers with Multi-head Latent Attention (MLA) (DeepSeek-AI et al., 2024) layers at a 3:1 ratio (Figure 1). The MLA layers handle tasks that benefit from full pairwise interaction, while the linear attention layers provide efficient long-range recurrence.

4 Experiments

Setup Our experiments evaluate FG^2-GDN alongside FG^2-GDN+, which uses separate channel-wise β_t^k and β_t^v for keys and values respectively. We compare against two strong baselines: Gated DeltaNet (GDN) (Yang et al., 2024a) with scalar α_t and scalar β_t, and Kimi Delta Attention (KDA) (Zhang et al., 2025) with channel-wise $\alpha_t \in \mathbb{R}^{d_k}$ but scalar β_t. For fair comparison, all models share an identical hybrid architecture that interleaves Multi-head Latent Attention (DeepSeek-AI et al., 2024) and linear attention layers at a 3:1 ratio. We sample tokens from the SlimPajama dataset and train at two scales: **340M** (15B tokens) and **1.3B** (100B tokens) using AdamW with 8K sequence length. Full details are shown in Appendix A.

4.1 Language Modeling

Table 1 summarizes language modeling perplexity and zero-shot accuracy on seven commonsense reasoning benchmarks: LAMBADA (Paperno et al., 2016), ARC-Easy/Challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and Winogrande (Sakaguchi et al., 2021).

Fine-grained β_t yields consistent improvements on language modeling. FG^2-GDN achieves the best average accuracy at both scales and the lowest LAMBADA perplexity at 1.3B, while FG^2-GDN+ obtains the lowest perplexity at 340M. An informative comparison can be drawn between KDA and FG^2-GDN, as the two models refine different components of the gated delta rule: KDA makes the *forgetting* gate α_t channel-wise, while FG^2-GDN further makes the *writing* rate β_t channel-wise. In our experiments, KDA performs comparably to GDN on average at 340M, whereas FG^2-GDN improves over both baselines at both scales.

4.2 Long-Context Evaluation

We evaluate long-context capability on two benchmarks with the 1.3B models: RULER (Hsieh et al., 2024), which tests 13 retrieval categories at sequence lengths 4k–16k (Table 2), and LongBench (Table 3), which covers long context understanding tasks like document QA, summarization, few-shot learning, and code completion.

Fine-grained β_t sharpens key-value bindings. On single-key retrieval tasks with varied query pat-

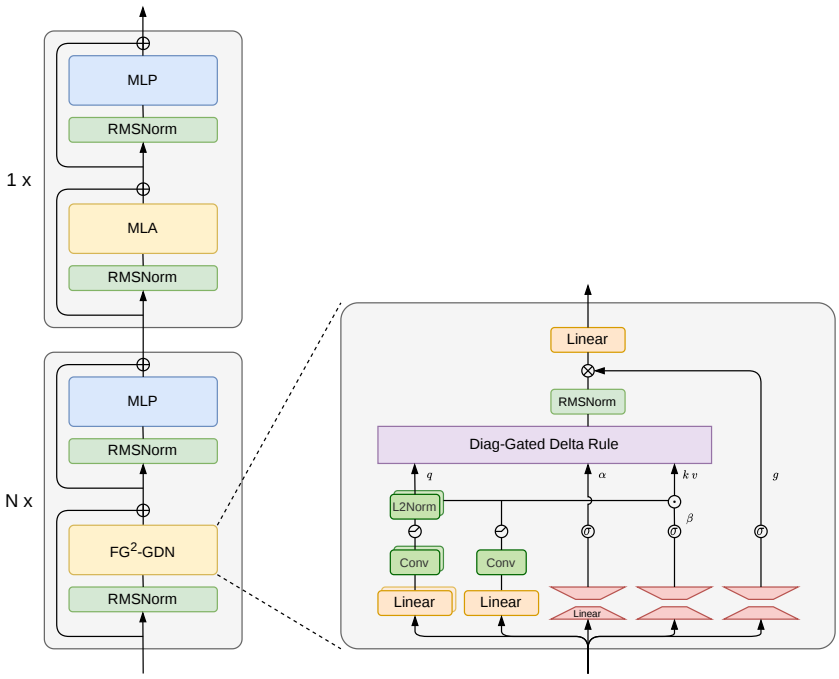

Figure 1: Overview of the Linear-MLA hybrid architecture. The backbone follows standard pre-norm residual blocks with SwiGLU MLP. Every N_{ratio} layers, the linear attention module is replaced by MLA. Within the FG^2-GDN module, low-rank projections follow KDA, and the channel-wise β pre-scales k and v before the delta update.

terns (MQ, CWE), FG^2-GDN delivers substantial gains over both GDN and KDA. We attribute this to the per-coordinate writing mechanism: by assigning different learning rates to different feature dimensions, the model can concentrate its write energy on the most discriminative dimensions of each key, producing tighter key-value bindings that survive query-time perturbation. This effect is most visible on tasks where the retrieval target is unique, but the query formulation varies.

Complex retrieval benefits from decoupled writing and forgetting. On multi-key retrieval (MK2), FG^2-GDN+ leads across all sequence lengths, suggesting that decoupling erasure and write strength helps manage multiple competing associations. The advantage is most pronounced at longer contexts where interference between stored key-value pairs intensifies.

Graceful extrapolation beyond training length. Though all models are trained with an 8K sequence length, we extend the test settings to 16K to further explore the extrapolation ability of NoPE-linear hybrid models. As shown in the bottom line of each category, FG^2-GDN+ retains the highest accuracy at 16K, suggesting that fine-grained memory control provides a more structured state representation that better tolerates distribution shift in position.

4.3 Ablation Study

To disentangle the contributions of our two design choices, namely channel-wise (vector) β and separate β^k/β^v for keys and values, we train additional 1.3B variants that enable each modification independently. Table **??** reports language modeling and LongBench results.

Channel-wise β improves language modeling. FG^2-GDN (vector β, shared for k and v) achieves the best LM average accuracy and LAMBADA perplexity, outperforming KDA by 0.8 points and reducing perplexity from 13.80 to 13.09. This suggests that per-dimension writing rates are particularly beneficial for short-context language understanding tasks.

Separate β^k/β^v improves long-context tasks. The variant with separate scalar β^k and β^v (without vectorization) also improves over KDA on both LM and LongBench metrics, indicating that decoupling erasure and write strength is beneficial even without channel-wise granularity.

The two improvements are complementary. FG^2-GDN+ combines both modifications and achieves the highest LongBench average (18.3%) and few-shot score (32.1%), substantially outper-

Model		LAMB.	LAMB.	ARC-e	ARC-c	BoolQ	Hellaswag	PIQA	Winogrande	AVG
		ppl	acc	acc	acc_norm	acc	acc_norm	acc	acc	
340M	GDN	39.86	32.47	43.10	**24.57**	54.77	35.75	63.82	50.12	43.51
	KDA	41.30	32.12	44.40	22.95	50.86	36.47	64.25	50.99	43.14
	FG^2-GDN	38.27	**33.59**	**45.79**	22.78	58.90	36.37	64.85	**51.85**	**44.88**
	FG^2-GDN+	**37.93**	32.97	44.19	23.98	**59.30**	**36.87**	**65.29**	51.30	44.84
1.3B	GDN	14.01	48.92	**58.63**	27.99	55.69	52.51	70.73	55.09	52.79
	KDA	13.80	**48.94**	56.52	26.54	**60.43**	**53.98**	71.22	54.70	50.33
	FG^2-GDN	**13.09**	48.85	57.95	**28.92**	59.54	53.67	**71.65**	57.06	**53.95**
	FG^2-GDN+	13.91	48.36	56.86	27.56	58.04	53.62	71.60	**57.77**	53.40

Table 1: Language modeling perplexity (LAMBADA) and zero-shot accuracy (%) on commonsense reasoning benchmarks at 340M and 1.3B scales.

Model	Seq Len	NIAH Single			NIAH MultiKey			MQ	MV	CWE	FWE	QA-H	QA-S	VT	**Avg**
		S1	S2	S3	MK1	MK2	MK3								
GDN	4k	100.0	100.0	86.6	**94.4**	56.4	10.6	78.1	90.2	67.1	36.1	28.2	33.9	19.5	61.6
	8k	100.0	100.0	75.0	**91.4**	24.6	3.2	65.8	72.1	36.7	31.1	25.0	23.3	14.2	51.0
	16k	99.8	99.0	58.8	71.6	12.6	0.4	41.7	42.4	19.5	58.5	20.0	19.5	**7.9**	42.4
KDA	4k	100.0	100.0	**93.8**	87.6	71.4	**31.4**	77.4	65.5	69.1	43.5	29.8	25.3	4.1	61.5
	8k	100.0	100.0	74.4	87.8	51.0	18.2	78.8	67.2	45.7	43.5	25.2	22.1	0.8	55.0
	16k	100.0	100.0	74.4	**74.6**	19.6	**7.8**	54.6	35.8	**26.7**	48.7	**22.0**	18.6	1.0	44.9
FG^2-GDN	4k	100.0	100.0	89.6	86.0	65.2	14.2	87.3	69.9	**77.3**	**44.3**	28.4	33.6	**47.1**	64.8
	8k	100.0	100.0	**88.2**	86.2	38.8	5.0	**89.1**	55.1	**47.4**	**59.9**	27.2	28.2	**19.6**	57.3
	16k	100.0	100.0	**82.6**	72.4	26.2	2.4	55.7	27.6	24.2	45.7	19.4	21.1	3.1	44.6
FG^2-GDN+	4k	100.0	100.0	88.0	87.8	**84.6**	30.8	**87.4**	**91.1**	70.9	39.3	**32.8**	**39.3**	21.2	**67.2**
	8k	100.0	100.0	87.2	79.2	**75.6**	**19.6**	82.1	**80.6**	38.5	55.5	**27.8**	**33.1**	19.4	**61.4**
	16k	100.0	98.6	81.2	59.6	**30.2**	4.8	**68.5**	**54.2**	23.2	**62.8**	19.8	**28.5**	3.8	**48.9**

Table 2: RULER benchmark results (%) for 1.3B models at sequence lengths 4k, 8k, and 16k. Categories include single-key NIAH (S1–S3), multi-key NIAH (MK1–MK3), multi-query (MQ), multi-value (MV), common/frequent word extraction (CWE/FWE), question answering (QA-H/QA-S), and variable tracking (VT).

forming all other variants on long-context tasks. While its LM accuracy (53.4%) is slightly below FG^2-GDN, it provides the best overall trade-off across short- and long-context settings, consistent with its strong performance on RULER (Table 2).

4.4 Efficiency

Expressiveness at near-zero cost. Figure 2 compares the prefill throughput and latency across configurations from 2k×16 to 32k×1. At the longest sequence (32k), all linear attention variants achieve 1.7–1.9× speedup over MLA. The channel-wise β_t parameterization introduces less than 5% overhead relative to KDA across all settings, and the gap narrows as sequences grow—precisely the regime where the expressiveness benefits matter most. FG^2-GDN+ incurs a slightly larger overhead from the additional β_t^v projection but remains competitive with all baselines.

Linear scaling preserved. The throughput ratio between 32k×1 and 2k×16 is approximately 0.70× for all linear attention variants, compared to 0.39× for MLA, confirming that FG^2-GDN inherits the favorable linear-time scaling of the delta-rule family.

5 Related Work and Discussion

Linear Attention as Recurrent Memory. Linear attention (Katharopoulos et al., 2020; Choromanski et al., 2020) reformulates softmax attention into a linear recurrence with a matrix-valued hidden state $S_t \in \mathbb{R}^{d_k \times d_v}$, which can be viewed as a fast-weight associative memory (Schmidhuber, 1992; Irie et al., 2021). A unified framework for these models writes the state transition as $S_t = A_t S_{t-1} + k_t v_t^\top$, where A_t is a transition matrix and $o_t = S_t^\top q_t$ reads out the memory. The primary axis of variation across recent models lies in the parameterization of A_t and the form of the write term. Table 5 summarizes representative methods under this lens.

Early gated variants employ scalar or diagonal

Model	Single-Doc QA			Multi-Doc QA			Summarization			Few-shot			Code		Avg
	NQA	QQA	MFQ	HQA	2WM	Mus	GvR	QMS	MNs	TRC	TQA	SSM	LCC	RBP	
GDN	2.6	7.5	13.7	5.1	7.8	2.5	24.0	14.8	18.5	28.0	15.8	13.3	30.3	27.4	15.1
KDA	**3.7**	7.8	**15.5**	4.7	7.1	3.1	**26.9**	14.2	20.3	25.0	29.1	7.8	**33.8**	**30.9**	16.4
FG^2-GDN	3.3	**9.2**	14.1	4.9	6.8	3.4	24.5	14.6	17.8	32.5	24.6	16.0	27.3	24.9	16.0
FG^2-GDN+	3.1	9.1	12.5	**5.2**	**7.9**	**3.8**	24.8	**15.0**	**22.0**	**45.5**	**34.5**	**16.4**	30.2	26.2	**18.3**

Table 3: LongBench evaluation results for 1.3B models across single-doc QA, multi-doc QA, summarization, few-shot learning, and code completion tasks.

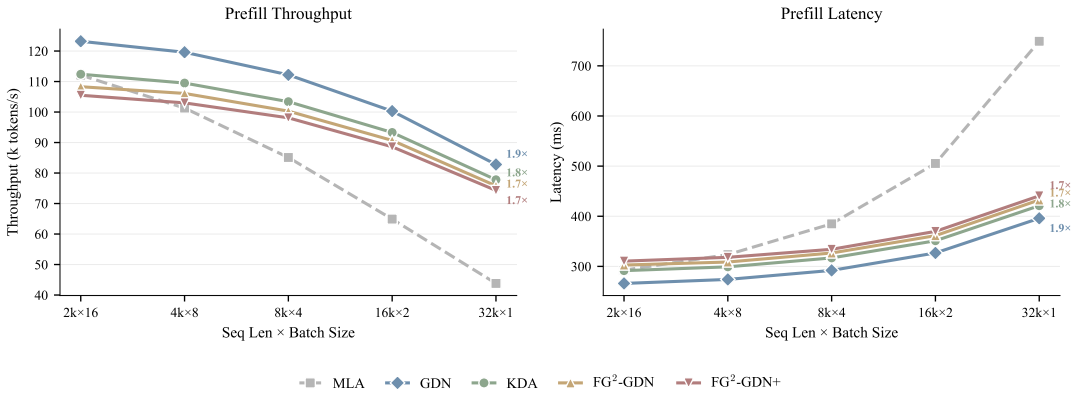

Figure 2: Prefill throughput (left) and latency (right) on NVIDIA H800-80G (BF16) with a fixed total token budget of 32,768. Numbers at the rightmost point indicate the speedup over MLA at the 32k×1 configuration. MLA (dashed) degrades sharply with increasing sequence length, while all linear attention variants scale similarly.

Model	Vec. β	Sep. β^k/β^v	Language Modeling		LongBench	
			LM Avg (↑)	PPL (↓)	LB Avg (↑)	Few-shot (↑)
KDA (baseline)	✗	✗	53.2	13.80	16.4	20.6
FG^2-GDN	✓	✗	**54.0**	**13.09**	16.0	24.4
+ sep. scalar β^k/β^v	✗	✓	53.9	13.56	16.7	27.4
FG^2-GDN+ (both)	✓	✓	53.4	13.91	**18.3**	**32.1**

Table 4: Ablation study (1.3B scale) isolating the two design axes of FG^2-GDN+: channel-wise β (scalar → vector) and separate β^k/β^v for keys and values. LM Avg: average zero-shot accuracy (%) over seven benchmarks; LB Avg: LongBench average score (%).

decay: RetNet (Sun et al., 2023) uses a fixed scalar γ; Mamba/Mamba-2 (Gu and Dao, 2024; Dao and Gu, 2024) introduce data-dependent scalar gates; GLA (Yang et al., 2024b) and RWKV-6 (Peng et al., 2023) extend to channel-wise diagonal gates $\mathrm{Diag}(\alpha_t)$, providing each feature dimension with an independent forgetting rate. HGRN-2 (Qin et al., 2024) further adopts diagonal gating with a coupled write term. The most recent RWKV-7 (Peng et al., 2025) goes beyond purely diagonal transitions by introducing a data-dependent low-rank correction, yielding an update $S_t = (\mathrm{Diag}(\alpha_t) - (\vec{\beta}_t \odot \hat{\kappa})\hat{\kappa}^\top)S_{t-1} + k_t v_t^\top$ that shares structural similarity with the delta rule but replaces the exponential decay with a normalized rank-1 subtraction. This design also admits an online learning interpretation analogous to DeltaNet's.

Delta Rule and Online Learning. The delta rule, also known as the Widrow–Hoff learning rule (Widrow and Hoff, 1988), interprets each recurrent update as a single SGD step on the objective $\|S^\top k_t - v_t\|^2$. DeltaNet (Schlag et al., 2021; Yang et al., 2024c) introduced this principle to linear attention, achieving superior associative recall through rank-1 corrective updates. Gated DeltaNet (GDN) (Yang et al., 2024a) unified the delta rule with a scalar forget gate α_t, and Longhorn (Liu et al., 2024) formalized the connection to amortized online learning. KDA (Zhang et al., 2025) refined the gate from scalar to channel-wise $\mathrm{Diag}(\alpha_t)$, significantly improving long-context modeling. Our FG^2-GDN extends this line of work along the orthogonal axis of *writing granularity*, replacing the scalar learning rate β_t with a per-coordinate vector $\beta_t \in \mathbb{R}^{d_k}$.

Hybrid Architectures. Purely linear models are fundamentally constrained by finite-state capacity, making exact copying and fine-grained long-context retrieval challenging (Gu and Dao, 2024). Hybrid architectures that interleave linear and full attention layers have emerged as a practical com-

Model	Transition A_t	Delta rule?	State update
Linear Attention (Katharopoulos et al., 2020)	I	✗	$S_t = S_{t-1} + k_t v_t^\top$
RetNet (Sun et al., 2023)	scalar	✗	$S_t = \gamma\, S_{t-1} + k_t v_t^\top$
Mamba-2 (Dao and Gu, 2024)	scalar	✗	$S_t = \alpha_t\, S_{t-1} + k_t v_t^\top$
GLA (Yang et al., 2024b)	diagonal	✗	$S_t = \mathrm{Diag}(\alpha_t)\, S_{t-1} + k_t v_t^\top$
RWKV-6 (Peng et al., 2023)	diagonal	✗	$S_t = \mathrm{Diag}(\alpha_t)\, S_{t-1} + k_t v_t^\top$
HGRN-2 (Qin et al., 2024)	diagonal	✗	$S_t = \mathrm{Diag}(\alpha_t)\, S_{t-1} + (1-\alpha_t) v_t^\top$
DeltaNet (Schlag et al., 2021)	diag. + low-rank	✓	$S_t = (I - \beta_t k_t k_t^\top) S_{t-1} + \beta_t k_t v_t^\top$
RWKV-7 (Peng et al., 2025)	diag. + low-rank	✓	$S_t = (\mathrm{Diag}(\alpha_t) - (\vec{\beta}_t \odot \hat{\kappa})\hat{\kappa}^\top) S_{t-1} + k_t v_t^\top$
GDN (Yang et al., 2024a)	diag. + low-rank	✓	$S_t = (I - \beta_t k_t k_t^\top)\alpha_t S_{t-1} + \beta_t k_t v_t^\top$
KDA (Zhang et al., 2025)	diag. + low-rank	✓	$S_t = (I - \beta_t k_t k_t^\top)\mathrm{Diag}(\alpha_t)\, S_{t-1} + \beta_t k_t v_t^\top$
FG^2-GDN	diag. + low-rank	✓	$S_t = (I - \hat{k}_t \hat{k}_t^\top)\mathrm{Diag}(\alpha_t)\, S_{t-1} + \hat{k}_t \hat{v}_t^\top$
FG^2-GDN+	diag. + low-rank	✓	$S_t = (I - \hat{k}_t \hat{k}_t^\top)\mathrm{Diag}(\alpha_t)\, S_{t-1} + \hat{k}_t \tilde{v}_t^\top$

Table 5: Unified view of linear recurrent models through state update rules. We classify transition matrices A_t by structure (scalar, diagonal, or low-rank) and whether the write term involves a delta-rule correction.

promise. Jamba (Lieber et al., 2024) combines Mamba with Transformer blocks in a MoE setting; Gated DeltaNet (Yang et al., 2024a) explored hybrids with sliding-window and global attention; Kimi Linear (Zhang et al., 2025) adopts a 3:1 KDA-to-MLA ratio, demonstrating that such hybrids can *surpass* full-attention baselines in both quality and efficiency at scale. Our work follows this inter-layer hybrid paradigm, using the same 3:1 ratio to ensure fair comparison, and focuses on improving the linear attention component.

6 Conclusion

We present FG^2-GDN, an extension of the delta rule-based linear model that replaces the scalar learning rate β_t in the delta update with a channel-wise vector, allowing each feature dimension to adapt its writing strength independently. Experiments at the two scales show that this fine-grained control consistently improves long-context retrieval performance on RULER and LongBench while maintaining competitive language modeling quality. The additional parameter overhead is marginal, and prefill throughput remains comparable to the baseline GDN variant. We hope this work provides a useful data point on the benefits of fine-grained adaptivity in linear recurrent models.

7 Limitations

Scale and Architecture Coverage. Our experiments are conducted at the 340M and 1.3B parameter scales with dense hybrid architectures. Due to computational resource constraints, we have not validated FG^2-GDN on larger-scale models (e.g., 7B or above) or Mixture-of-Experts (MoE) architectures, where the interplay between channel-wise adaptive writing and sparse expert routing may introduce additional dynamics that are absent in the dense setting.

Inference Challenges. While we demonstrate that FG^2-GDN preserves favorable training efficiency, the deployment of linear attention-based hybrid models within existing inference systems presents engineering challenges that waiting for further investigation. Current serving frameworks are heavily optimized around the KV-cache paradigm of softmax attention, whereas linear attention maintains a fixed-size recurrent state ($d_k \times d_v$ per head) that requires different cache management strategies. Moreover, speculative decoding in such hybrid architectures requires profiling heterogeneous memory access patterns across layer types and managing recurrent state rollback upon token reception. Adapting these inference-time optimizations to linear attention hybrids remains an open problem.

References

Christian Bischof and Charles Van Loan. 1987. The wy representation for products of householder matrices. *SIAM Journal on Scientific and Statistical Computing*, 8(1):s2–s13.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. PIQA: Reasoning about physical intuition in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, and 1 others. 2020. Re-

thinking attention with performers. *arXiv preprint arXiv:2009.14794*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*.

Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Joshua Ainslie, Dhruv Aithal, and 1 others. 2024. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. 2022. Transformer quality in linear time. In *International conference on machine learning*, pages 9099–9117. PMLR.

Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. 2021. Going beyond linear transformers with recurrent fast weight programmers. *Advances in neural information processing systems*, 34:7703–7717.

Thierry Joffrain, Tze Meng Low, Enrique S Quintana-Ortí, Robert van de Geijn, and Field G Van Zee. 2006. Accumulating householder transformations, revisited. *ACM Transactions on Mathematical Software (TOMS)*, 32(2):169–179.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, and 1 others. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.

Bo Liu, Rui Wang, Lemeng Wu, Yihao Feng, Peter Stone, and Qiang Liu. 2024. Longhorn: State space models are amortized online learners. *arXiv preprint arXiv:2407.14207*.

Denis Paperno, Germán Kruszewski, Angeliki Dufter, and Marco Baroni. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1525–1534.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, and 1 others. 2023. Rwkv: Reinventing rnns for the transformer era. In *Findings of the association for computational linguistics: EMNLP 2023*, pages 14048–14077.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, and 1 others. 2025. RWKV-7: Goose with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*.

Zhen Qin, Songlin Li, Weixuan Sun, Xuyang Sun, Dong Li, Weigao Zhong, and 1 others. 2024. HGRN2: Gated linear rnns with state expansion. *arXiv preprint arXiv:2404.07904*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pages 9355–9366. PMLR.

Jürgen Schmidhuber. 1992. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Bernard Widrow and Marcian E Hoff. 1988. Adaptive switching circuits. In *Neurocomputing: foundations of research*, pages 123–134.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2024a. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. 2024b. Gated linear attention transformers with hardware-efficient training. *Proceedings of the 41st International Conference on Machine Learning*.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. 2024c. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37:115491–115522.

Songlin Yang and Yu Zhang. 2024. Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, Wentao Li, Enzhe Lu, Weizhou Liu, Yanru Chen, Weixin Xu, Longhui Yu, Yejie Wang, Yu Fan, Longguang Zhong, Enming Yuan, and 40 others. 2025. Kimi linear: An expressive, efficient attention architecture. *Preprint*, arXiv:2510.26692.

A Experimental Details

All models in our experiments share an identical hybrid architecture that interleaves Multi-head Latent Attention (MLA) (DeepSeek-AI et al., 2024) layers with linear attention layers in a 3:1 ratio (i.e., three linear attention layers followed by one MLA layer). This design follows the inter-layer hybrid paradigm adopted by Kimi Linear (Zhang et al., 2025) and ensures that the only variable across compared models is the linear attention mechanism itself.

We experiment at two scales: **340M** parameters (20 layers, hidden size 1024) and **1.3B** parameters (24 layers, hidden size 2048). Both MLA and linear attention use 16 heads. The MLA layers adopt low-rank projections for queries and key-value pairs (q_lora_rank and kv_lora_rank in Table 6), and we disable RoPE in MLA layers so that the linear attention layers can serve as the sole source of positional information through their recurrent dynamics, following the design choice in KDA.

Model Architecture		**Training Hyperparameters**	
Parameter	**340M / 1.3B**	**Hyperparameter**	**340M / 1.3B**
Number of layers	20 / 24	Optimizer	AdamW
Hidden size	1024 / 2048	Learning rate	$3 \times 10^{-4} \rightarrow 3 \times 10^{-5}$
Hybrid mode	3:1	LR Scheduler	Cosine w/ Warmup
MLA / Linear heads	16	Warmup tokens	0.5B / 1B
MLA q_lora_rank	384 / 1536	Training tokens	15B / 100B
MLA kv_lora_rank	128 / 512	Sequence length	8192
Linear head dim	128	Tokens per step	0.5M / 2M
Intermediate size	3072 / 6144	Precision	BF16
Activation function	SiLU	MLA RoPE	Off

Table 6: Model architecture and training hyperparameters for 340M and 1.3B scales.

The linear attention head dimension is fixed at 128 across both scales.

Both scales are trained with the AdamW optimizer (Kingma and Ba, 2014) using a cosine annealing schedule with linear warmup. The peak learning rate is 3×10^{-4}, decaying to 3×10^{-5}. The 340M model is trained on 15B tokens with 0.5B warmup tokens and a per-step batch size of 0.5M tokens; the 1.3B model is trained on 100B tokens with 1B warmup tokens and 2M tokens per step. All training uses a sequence length of 8192 and BF16 mixed precision.