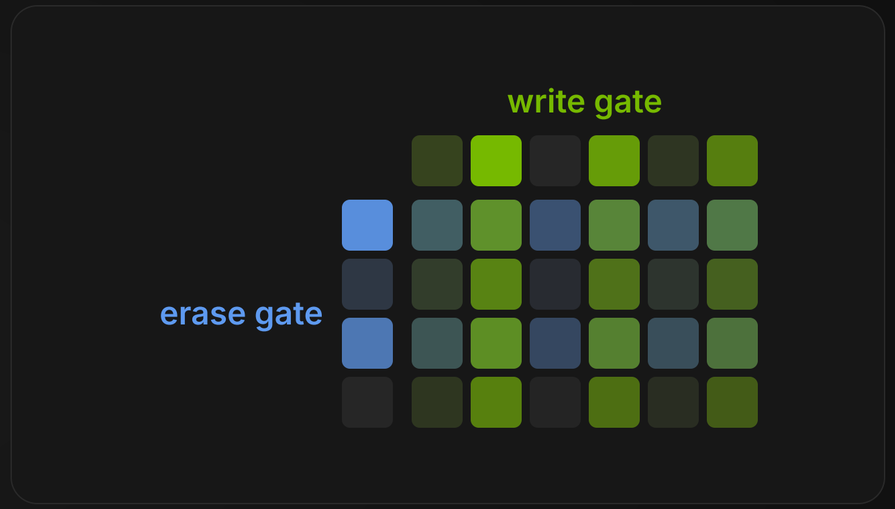
Gated DeltaNet-2

Decoupling Erase and Write in Linear Attention

Ali Hatamizadeh, Yejin Choi, Jan Kautz

NVIDIA

NeurIPS 2026



Built on layers that already run in leading open models

From Gated DeltaNet, through Kimi Delta Attention, to Gated DeltaNet-2

Gated DeltaNet

ICLR 2025

the linear attention layer in

- Qwen3-Next
- Qwen3.5
- Qwen3.6
- Qwen3.8

KDA

Kimi Delta Attention · 2025

powers

- Kimi K3.1 Pro

among the leading open-weight models

Gated DeltaNet-2

this work · NeurIPS 2026

one gate becomes two

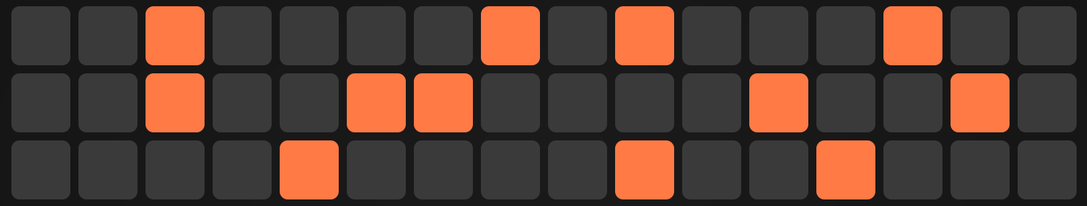
- erase
- write

Every association shares one fixed-size memory

Linear attention replaces the growing cache of softmax attention with one recurrent state



linear attention writes them all into one state



The state stays the same size at any context length. Orange: associations that collide.

linear
time in sequence length

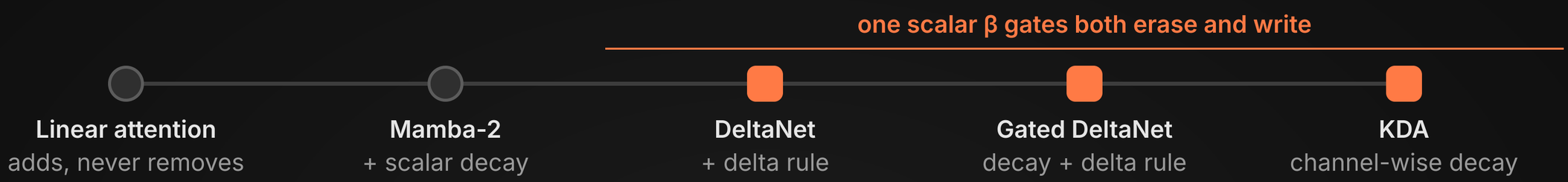
constant
memory at decoding

262,144
floats of state per layer in the 1.3B models, at any context length

The state is a compressed key–value memory: each new association lands on top of the ones already stored.

One scalar gate still controls both erase and write

The delta-rule edit shared by DeltaNet, Gated DeltaNet and KDA



KDA update $$\mathbf{S}_t = (\mathbf{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top)\, \mathbf{D}_t \mathbf{S}_{t-1} + \beta_t \boldsymbol{k}_t \boldsymbol{v}_t^\top$$

erases old content, key side

writes new content, value side

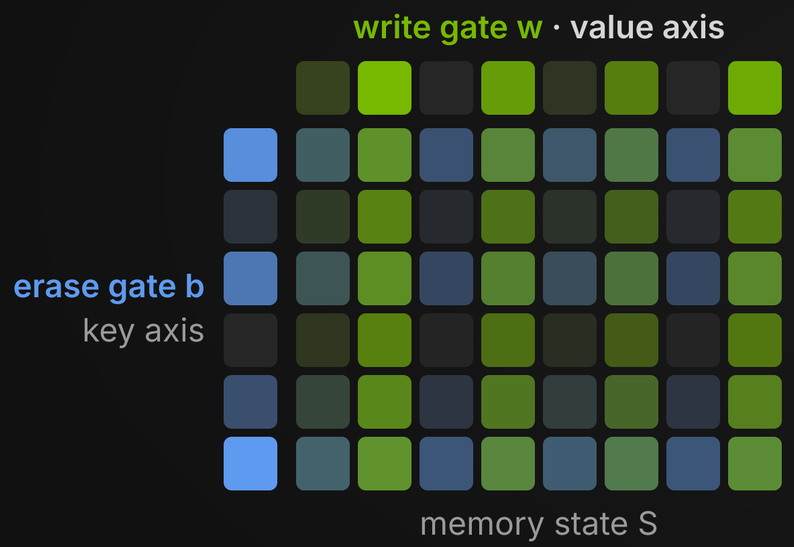
Two decisions, one number

Which key coordinates of the old read to erase, and which value coordinates of the new value to commit, need not agree.

Gated DeltaNet-2: one gate to erase, one gate to write

Gated Delta Rule-2 replaces the tied scalar with two channel-wise gates

$$\mathbf{S}_t = \left(\mathbf{I} - \boldsymbol{k}_t(\boldsymbol{b}_t \odot \boldsymbol{k}_t)^\top\right)\mathbf{D}_t\mathbf{S}_{t-1} + \boldsymbol{k}_t(\mathbf{w}_t \odot \boldsymbol{v}_t)^\top$$



1 Decay

Channel-wise forgetting, inherited from KDA. It clears broad context.

2 Erase gate b

Key side. It chooses which coordinates of the old read to remove.

3 Write gate w

Value side. It chooses which coordinates of the new value to commit.

Tie both gates to a single scalar and it is KDA. Tie the decay too and it is Gated DeltaNet.

Finer control, same efficient training

Training throughput on one H100 GPU, hybrid 1.3B models, fixed token budget (thousand tokens per second)



Gated DeltaNet-2: 38.0 → 36.1
Nearly flat from 2K to 16K sequences.

Transformer
Drops sharply as sequences grow.

Two extra gates, small cost
A small constant gap to KDA in training.
Modeled decoding latency: +4.3 to 6.5%.

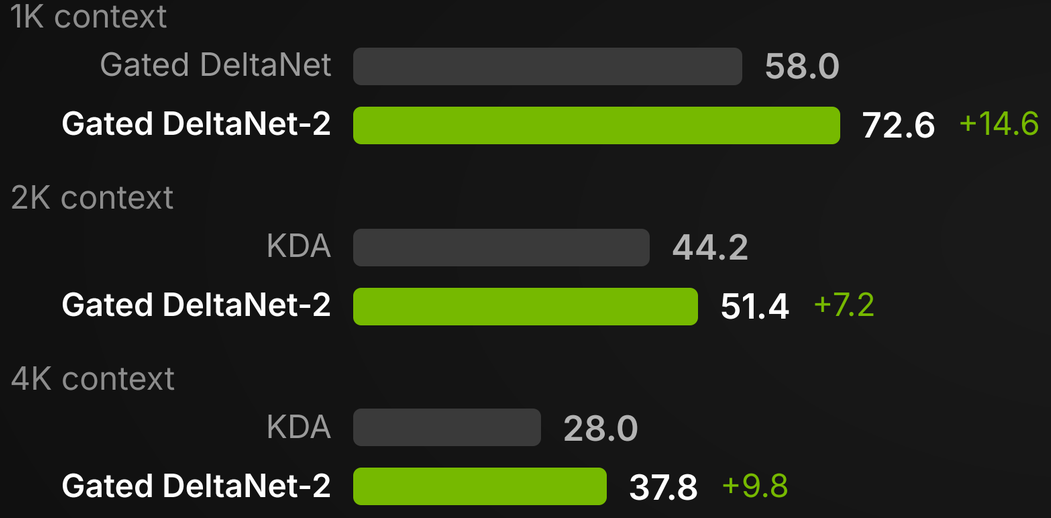
Same chunkwise form as KDA
Channel-wise decay is absorbed into the erase factors, giving a compact WY form with a gate-aware backward pass.

Strongest where memory editing matters most

1.3B parameters, 100B FineWeb-Edu tokens, matched state size, against Mamba-2, Gated DeltaNet, KDA and Mamba-3

Multi-key retrieval, RULER MK-NIAH-1 (accuracy, %)

Recurrent models: the strongest baseline against Gated DeltaNet-2



Same state size, stronger update rule

Recurrent state size is matched across models, so the gain comes from the update rule, not from a larger memory.

Gain over the strongest baseline (points)

LAMBADA and commonsense reasoning, average accuracy

Recurrent	**+0.72**	53.11 vs 52.39
Hybrid	**+1.25**	53.97 vs 52.72

Real-world retrieval, average of six tasks

Recurrent	**+1.21**	29.88 vs 28.67
Hybrid	**+1.27**	42.28 vs 41.01

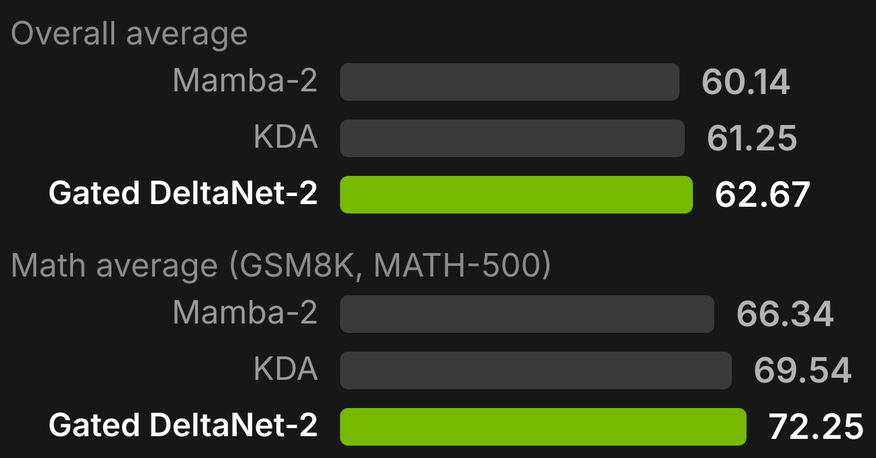
Hybrid: the same mixer plus sliding-window attention

The gain holds at scale, and both gates earn it

A 31.6B-parameter hybrid latent-MoE with full global attention, trained on 300B tokens, and ablations at 1.3B

31.6B hybrid with global attention

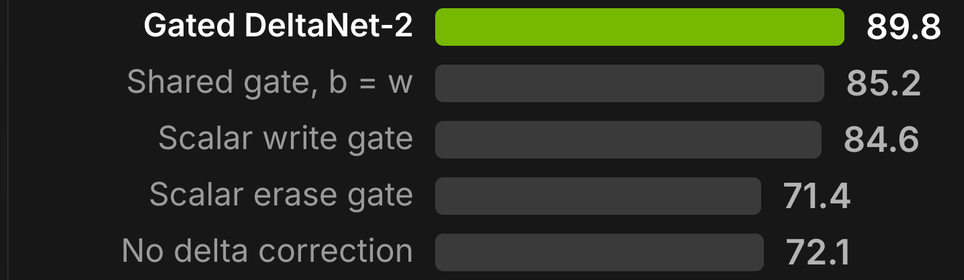
Only the recurrent mixer changes. Accuracy, %



Best on 9 of 10 benchmarks, and on every category average

What each piece buys

S-NIAH-3 at 2K, recurrent 1.3B models. Accuracy, %



Both gates contribute. The erase gate accounts for most of the gain.

One gate to erase. One gate to write.

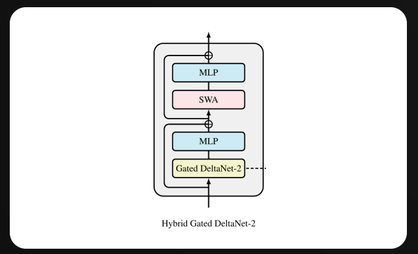
Channel-wise control over what leaves a fixed-size memory, and what enters it.

NVIDIA

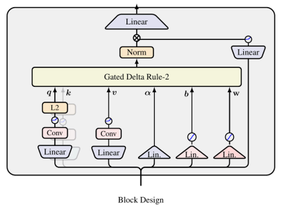
Gated DeltaNet-2

Decoupling Erase and Write in Linear Attention

NVIDIA · NeurIPS 2026



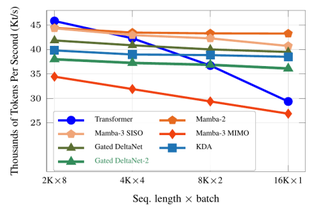
Hybrid Gated DeltaNet-2



Block Design

$$\mathbf{S}_t = \left(\mathbf{I} - \boldsymbol{k}_t(\boldsymbol{b}_t \odot \boldsymbol{k}_t)^\top\right)\mathbf{D}_t\mathbf{S}_{t-1} + \boldsymbol{k}_t(\mathbf{w}_t \odot \boldsymbol{v}_t)^\top$$

Model	S-NIAH-1				S-NIAH-2				S-NIAH-3				MK-NIAH-1			
	1K	2K	4K	8K	1K	2K	4K	8K	1K	2K	4K		1K	2K	4K	
Recurrent models																
Mamba-2	**100.0**	**100.0**	97.0	55.8	99.6	99.6	62.6	21.0	59.2	38.6	14.4		29.0	21.2	21.4	
Gated DeltaNet	99.8	**100.0**	**100.0**	97.6	**100.0**	**100.0**	87.2	32.0	89.8	54.2	**60.6**		58.0	37.0	27.8	
KDA	**100.0**	**100.0**	99.2	70.6	**100.0**	**100.0**	89.0	30.6	77.4	63.2	26.2		54.0	44.2	28.0	
Mamba-3 (SISO)	**100.0**	99.0	63.4	27.8	99.8	99.0	59.4	25.2	60.2	35.6	12.2		44.8	27.4	20.2	
Mamba-3 (MIMO)	**100.0**	99.8	93.0	35.6	99.8	98.8	64.2	27.2	89.2	72.4	29.2		49.4	19.2	18.0	
Gated DeltaNet-2	**100.0**	**100.0**	**100.0**	**97.8**	**100.0**	**100.0**	**93.0**	**39.2**	**92.0**	**89.8**	31.8		**72.6**	**51.4**	**37.8**	
Attention or hybrid models																
Transformer	**100.0**	**100.0**	51.2	0.0	**100.0**	**100.0**	44.2	0.0	95.8	94.8	37.0		75.6	66.6	38.2	
Mamba-2	**100.0**	**100.0**	51.8	25.4	**100.0**	99.6	52.4	25.8	97.8	86.8	48.0		82.0	58.6	39.0	
Gated DeltaNet	**100.0**	**100.0**	47.2	22.4	**100.0**	99.8	57.3	25.6	94.8	91.2	47.2		91.0	78.4	44.8	
KDA	**100.0**	**100.0**	51.8	26.2	**100.0**	**100.0**	56.0	23.0	97.2	93.4	51.6		91.4	84.0	40.4	
Mamba-3 (SISO)	**100.0**	**100.0**	49.6	26.0	**100.0**	**100.0**	**58.2**	27.8	95.0	90.4	44.0		78.8	65.6	33.6	
Mamba-3 (MIMO)	**100.0**	**100.0**	49.0	22.8	**100.0**	**100.0**	53.0	27.8	99.4	98.4	54.2		82.4	79.0	46.6	
Gated DeltaNet-2	**100.0**	**100.0**	**55.2**	**27.4**	**100.0**	**100.0**	57.9	**29.2**	**99.6**	**99.0**	**55.6**		**93.0**	**84.6**	**48.0**	



github.com/NVlabs/GatedDeltaNet-2