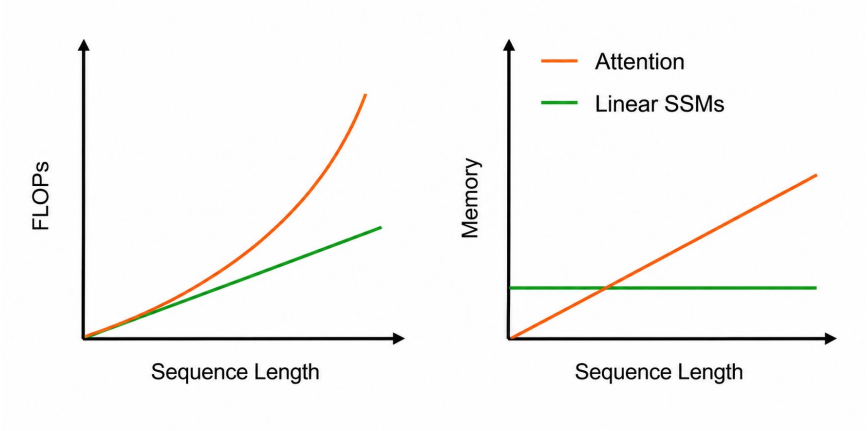
Gated KalmaNet: A Fading Memory Layer Through Test-Time Ridge Regression

Liangzu Peng, Aditya Chattopadhyay, Luca Zancato, Elvis Nunez, Wei Xia, Stefano Soatto

Rise of long-context models

- Long-context processing imperative to unlocking new model capabilities.
  - Code automation over large repositories.
  - Aggregate information across long videos.
  - Long-consistent reasoning traces for agentic applications.

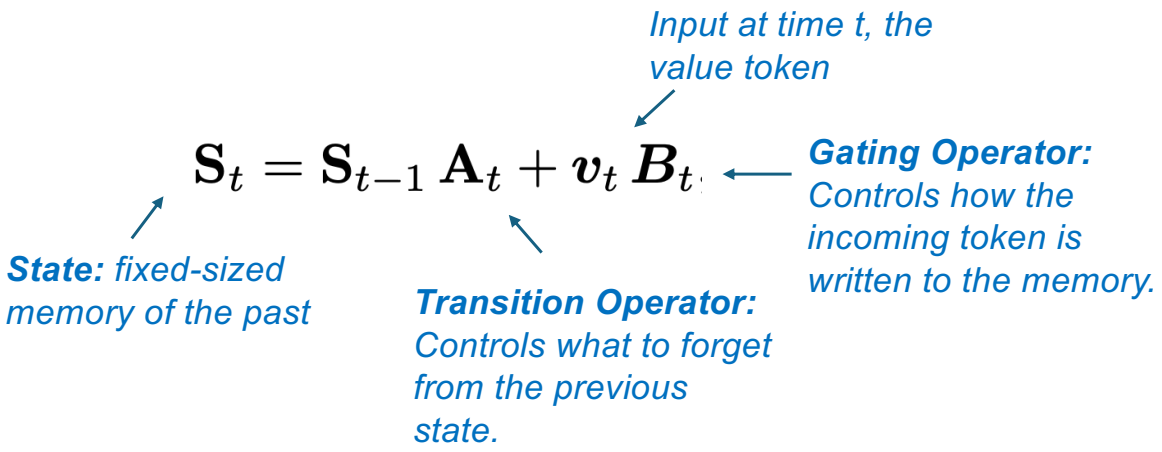
Background: Eidetic vs Fading Memory



- Eidetic (Attention): Keeps entire past verbatim in memory, high compute and storage costs, Perfect recall.
- Fading (SSMs): Compresses the past into a fixed-sized memory, low compute and storage costs, Low recall.

Current State-Space models are myopic

- All scalable SSM layers share the same linear recurrence.



- At every step, **what to forget** (A_t) and **what to write** (B_t) are functions of *only the current token* → low recall capabilities.

Gated KalmaNet: non-myopic SSM at scale!

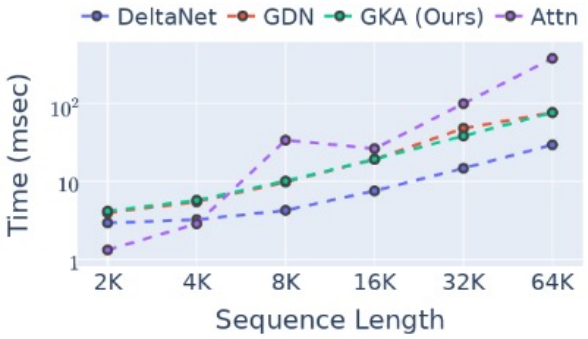
- We introduce Gated KalmaNet (GKA, pronounced "gee-ka"), that
  - updates its state using the entire past.
    - Specifically, A_t and B_t depend on all key tokens seen so far.

$$\mathbf{S}_t = \mathbf{S}_{t-1} \underbrace{\left(\mathbf{I} - \boldsymbol{k}_t\, \mathbf{g}_t^\top\right)}_{\mathbf{A}_t} + \boldsymbol{v}_t \underbrace{\mathbf{g}_t^\top}_{\mathbf{B}_t}$$

Non-linear function of all the keys seen so far; the entire past

  - enjoys linear-time compute and constant-storage costs like existing SSMs and it scales.
    - We implemented efficient Triton kernels and validated GKA up to 32B parameters.

GKA as comparable runtime with existing SSMs



(b) Runtime of a single memory layer

- GKA has linear time-complexity with sequence length.
- Comparable to GDN in (forward+backward) pass.
- Our parallel Triton implementation of GKA is fast.

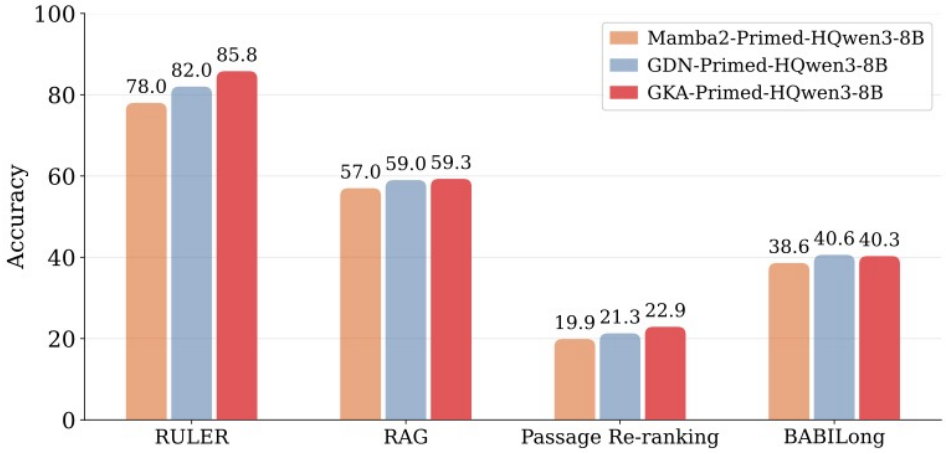
GKA on language modeling

Table 5. **GKA shows stronger scaling with compute that other SSM baseline models.** LM-Harness results for models at different scales: 440M, 1B and 2.8B. All models were trained from scratch. 440M and 1B models were trained on 8B and 20B tokens respectively in accordance to the Chinchila scaling laws [24]. For the 2.8B model we trained on 100B tokens.

Model	**ARC-C** acc_n ↑	**ARC-E** acc_n ↑	**BoolQ** acc ↑	**COPA** acc ↑	**HellaSWAG** acc_n ↑	**PIQA** acc_n ↑	**SciQ** acc_n ↑	**Winogrande** acc ↑	**FDA** contains ↑	**SWDE** contains ↑	**Avg**
2.8B Models											
Transformer	32.25	56.10	**64.28**	80.00	60.96	73.56	79.50	61.72	**58.53**	**72.28**	**63.92**
Mamba2	32.24	59.64	58.72	82.00	62.23	73.78	79.80	62.19	7.71	41.13	55.94
Gated Linear Attention	27.82	50.80	52.57	78.00	48.83	70.13	69.60	54.54	2.81	20.43	47.55
Gated DeltaNet	32.59	**60.02**	62.75	82.00	62.8	74.32	80.6	62.35	8.26	44.28	57.00
DeltaNet	**32.85**	58.16	42.51	81.00	61.13	73.78	43.90	61.72	11.80	46.08	51.29
Gated KalmaNet (Ours)	32.51	59.89	61.68	**85.00**	**63.84**	**74.81**	**83.2**	**64.17**	12.89	50.95	58.89

Hybrid GKA models on long-context tasks

- Hybrid models mix SSM and Attention layers to combine SSM efficiency with Attention's superior recall.
- We trained several Hybrid models at 8B scale – Hybrid GKA showed the strongest long-context capabilities.



GKA on Image Classification

Table 3. **GKAVision outperforms MambaVision on ImageNet classification** (averaged over 5 seeds). Models are ∼31.8M parameters; training throughput measured on 8 H200 GPUs with batch size 256. Unlike MambaVision, GKAVision uses the GKA layer without any vision-specific modifications.

Model	**Top-1 Accuracy (%) ↑**	**Throughput (K img/s) ↑**
MambaVision-T	81.18	**16.25**
GKAVision-T	<u>81.27</u>	<u>13.72</u>
NextViT-S	**81.99**	10.32

Gated KalmaNet: A Fading Memory Layer Through Test-Time Ridge Regression

Come visit us at Poster Session 3, #557