B'MOJO: Hybrid State Space Realizations of Foundation Models with Eidetic and Fading Memory

They who forget the pasta are bound to reheat it.

Luca Zancato @ AWS AI Labs

ASAP Seminar | 03-12-2025

Memory for sequence models

Processing long sequences: Modern long context models have nominal support for sequences of length 128k-1M tokens.

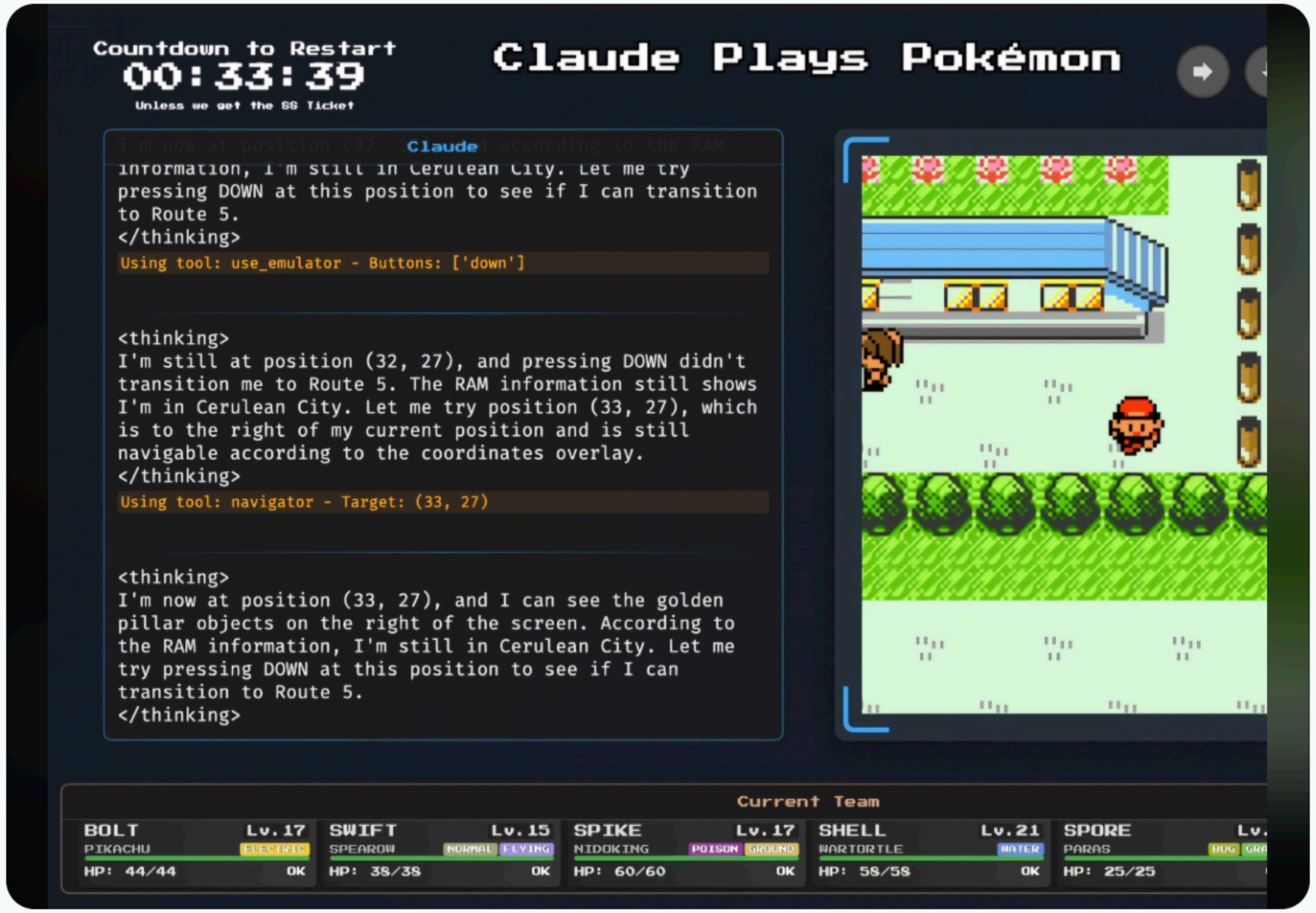
API PROVIDER	MODEL	CONTEXT WINDOW	ARTIFICIAL ANALYSIS INTELLIGENCE INDEX
OpenAI	o3-mini (high)	200k	66
deepseek	DeepSeek R1	64k	60
ANTHROP\C	Claude 3.7 Sonnet Thinking	200k	57
OpenAI	o1-mini	128k	54
Google	Gemini 2.0 Pro Experimental (AI Studio)	2m	49
Google	Gemini 2.0 Flash Vertex	1m	48

Source: https://huggingface.co/spaces/ArtificialAnalysis/LLM-Performance-Leaderboard

Reasoning: Better memory means better exploration/ more consistent reasoning traces.

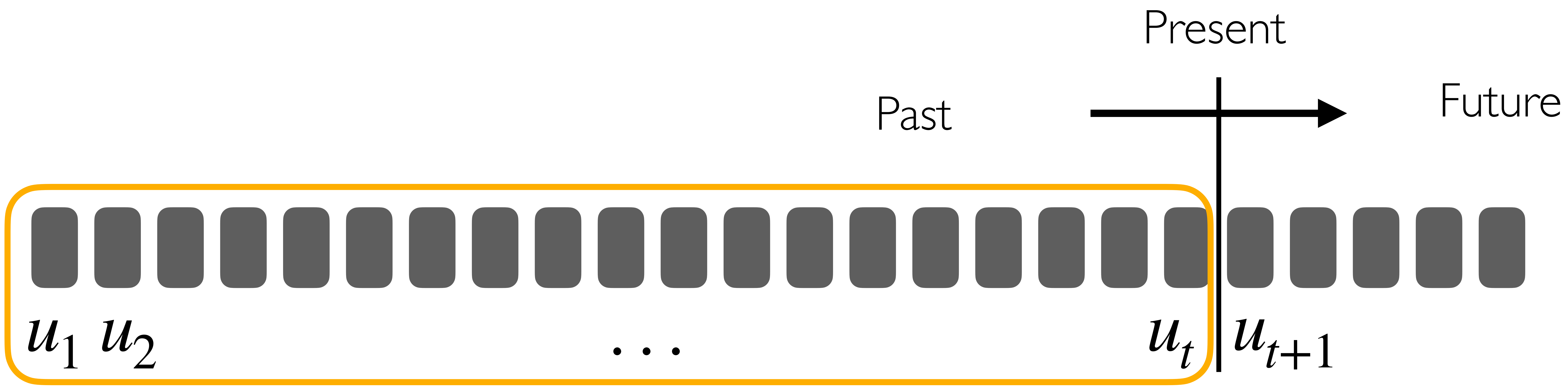
Claude Plays Pokemon - Claude Sonnet 3.7 has been stuck in a loop in Cerulean City for two straight days - restart planned

AI



aws

Memory realizations for sequence models

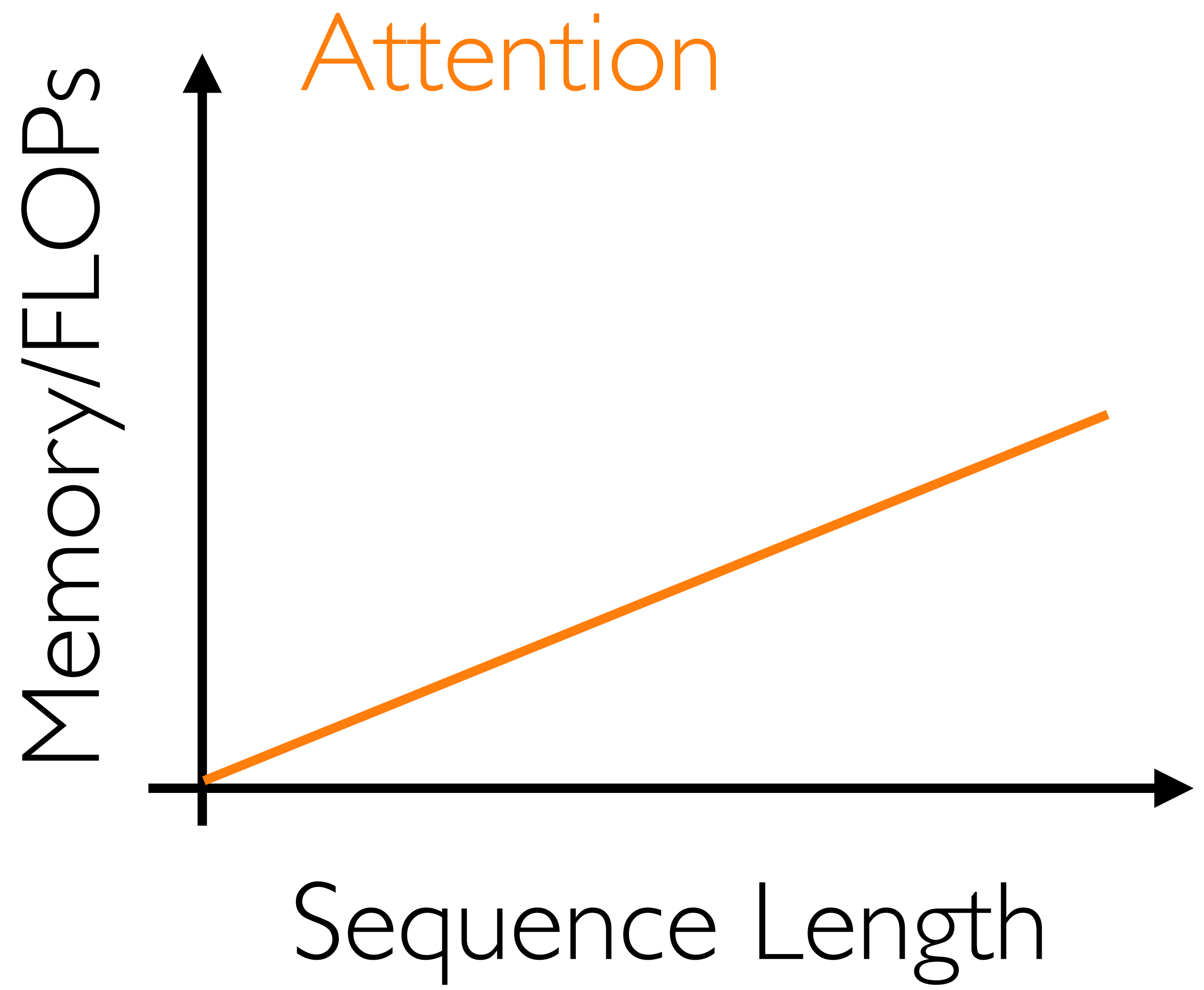


State: Function of the past that makes the future conditionally independent of the past

$$P(u_{t+1} \,|\, x_t) = P(u_{t+1} \,|\, u_{-\infty}^{t})$$

Easier said than done! We typically don't know the complexity of the data generation mechanism at the outset.

Benefit of better memory/state



Benefits for long context

Better memory → More effective and cheaper long contexts
→ Reduce reliance on ad-hoc "context editing/cleaning" heuristics

Benefits for Reasoning

Idea: On verifiable domains we know we can filter Best-of-N and learn to prefer successful generations.

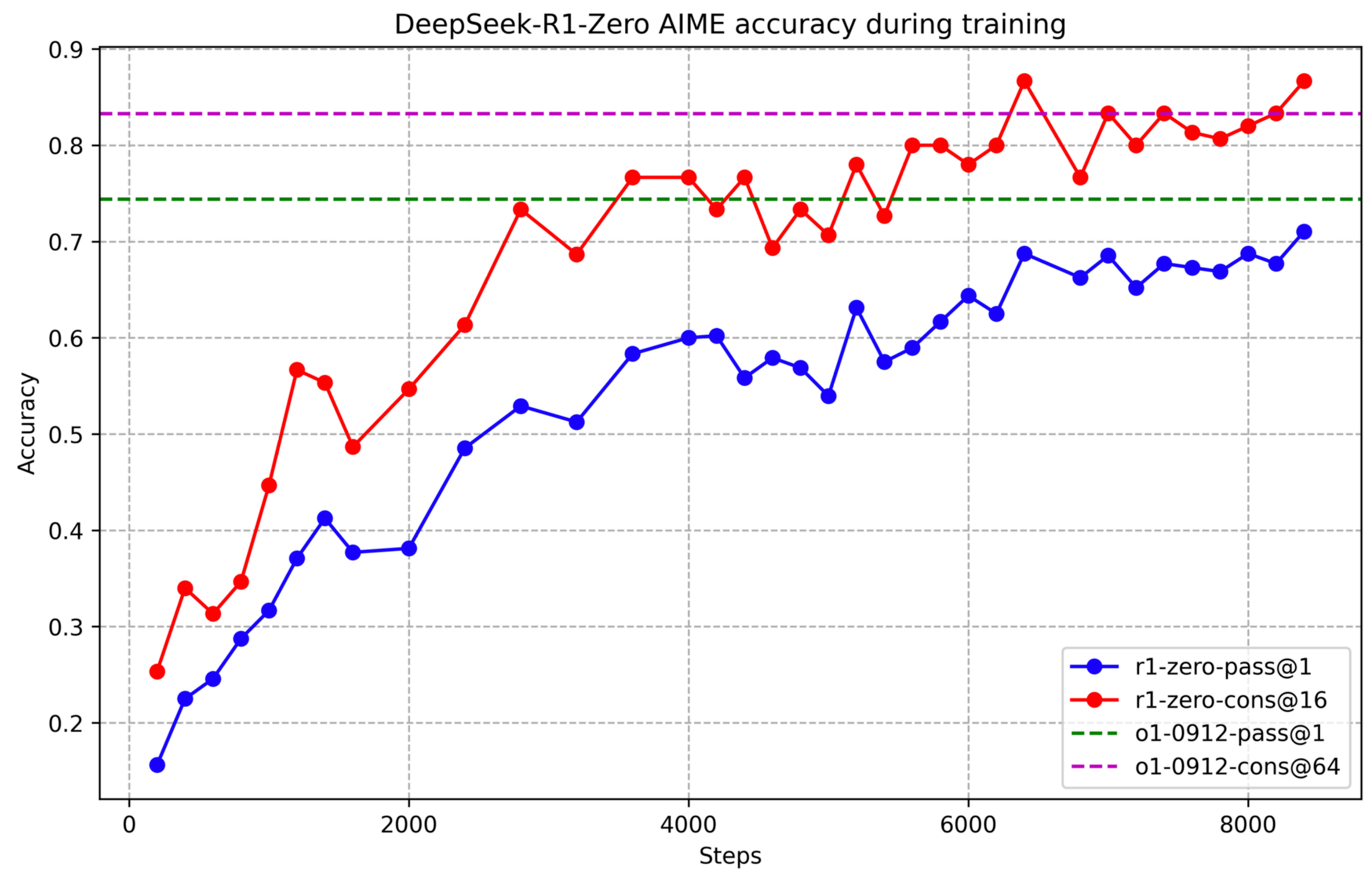
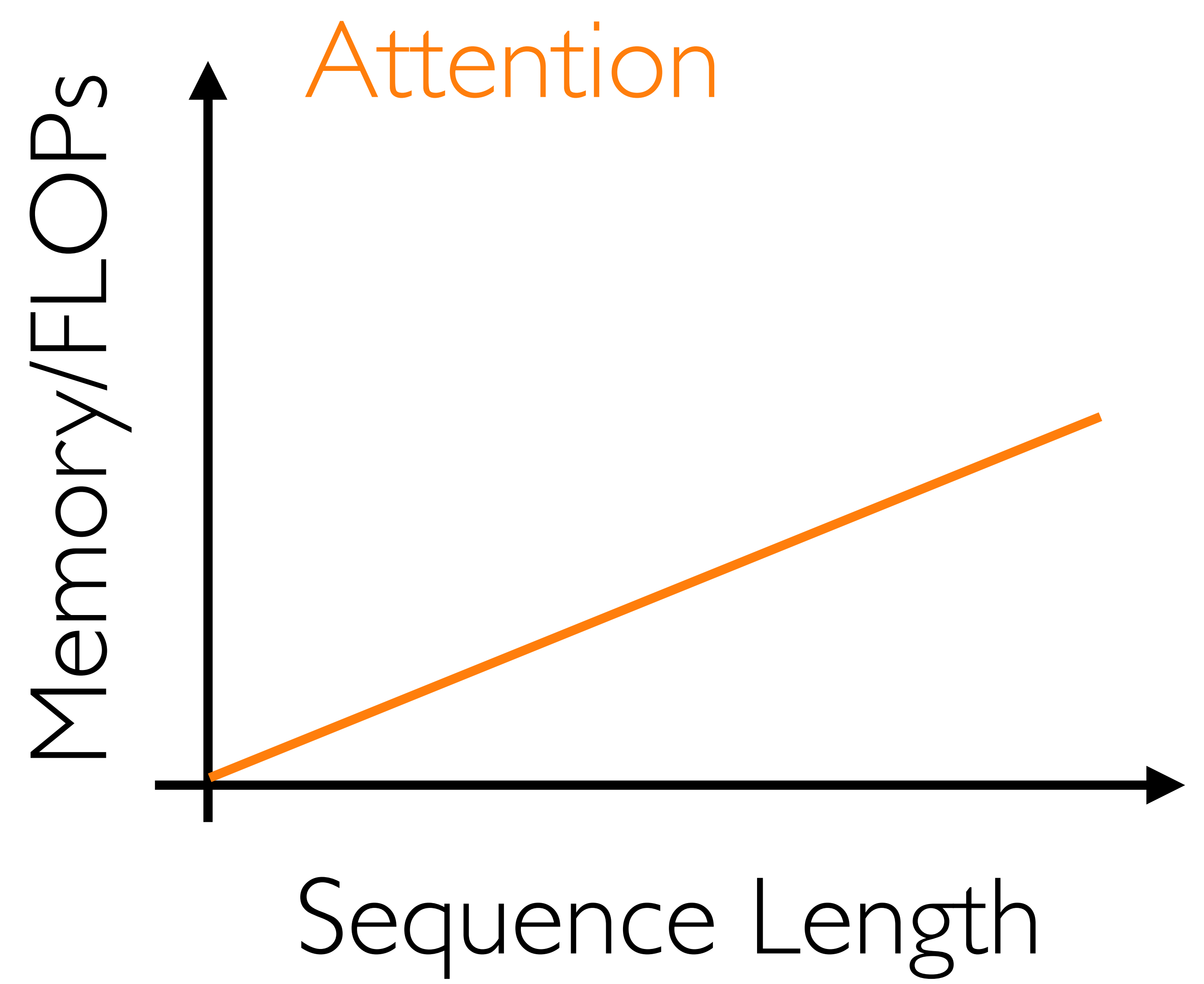


Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

DeepSeek-AI Team, "DeepSeek-R1", 2025

Benefit of better memory/state



Benefits for long context

Better memory → More effective and cheaper long contexts
→ Reduce reliance on ad-hoc "context editing/ cleaning" heuristics

Benefits for Reasoning

Idea: On <u>verifiable domains</u> we know we can filter Best-of-N and learn to prefer successful generations.

Cons: Throw spaghetti on the wall an hope they stick. Computationally VERY inefficient.

Better memory → Can increase "N" in Best-of-N
→ More scalable inference time compute

B'MOJO: Hybrid State Space Realizations of Foundation Models with Eidetic and Fading Memory

@NeurIPS 2024

Luca Zancato* **Arjun Seshadri** **Yonatan Dukler** **Aditya Golatkar** **Yantao Shen**

Benjamin Bowman **Matthew Trager** **Alessandro Achille** **Stefano Soatto**

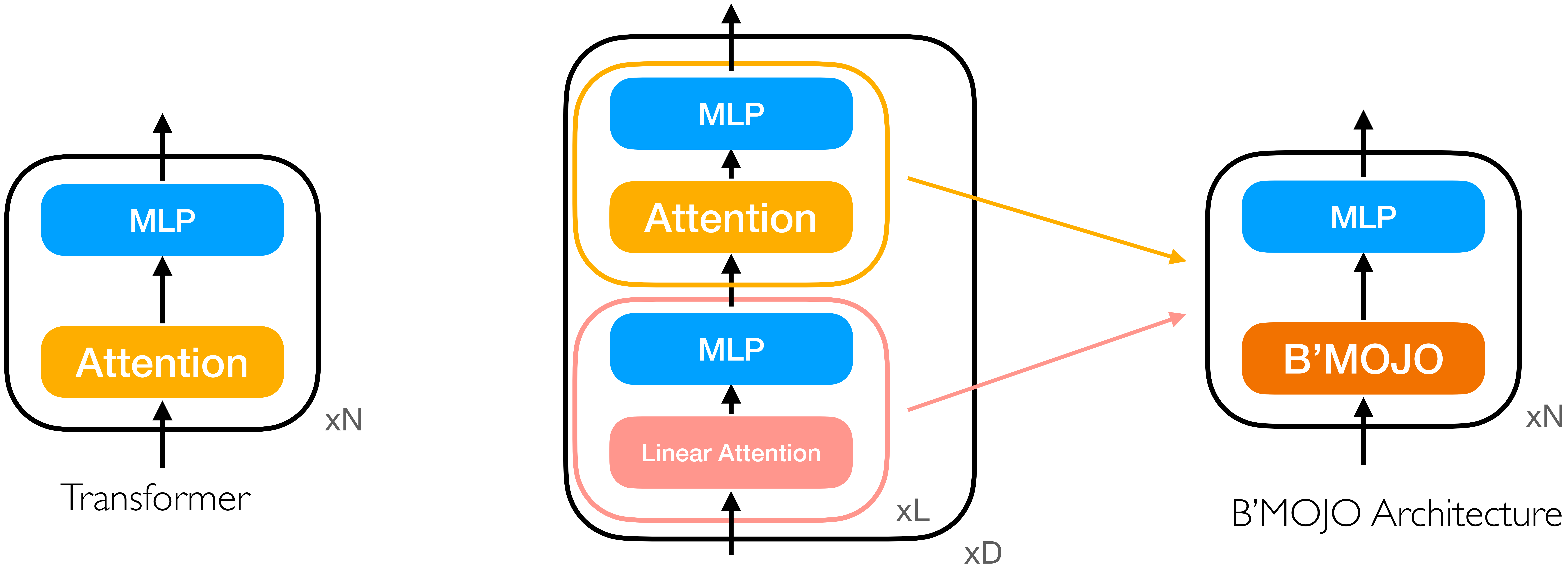
AWS AI Labs

[cs.LG] 8 Jul 2024

Abstract

We describe a family of architectures to support transductive inference by allowing memory to grow to a finite but a-priori unknown bound while making efficient use of finite resources for inference. Current architectures use such resources to represent data either eidetically over a finite span ("context" in Transformers), or fading over an infinite span (in State Space Models, or SSMs). Recent hybrid architectures have combined eidetic and fading memory, but with limitations that do not allow the designer or the learning process to seamlessly modulate the two, nor to extend the eidetic memory span. We leverage ideas from Stochastic

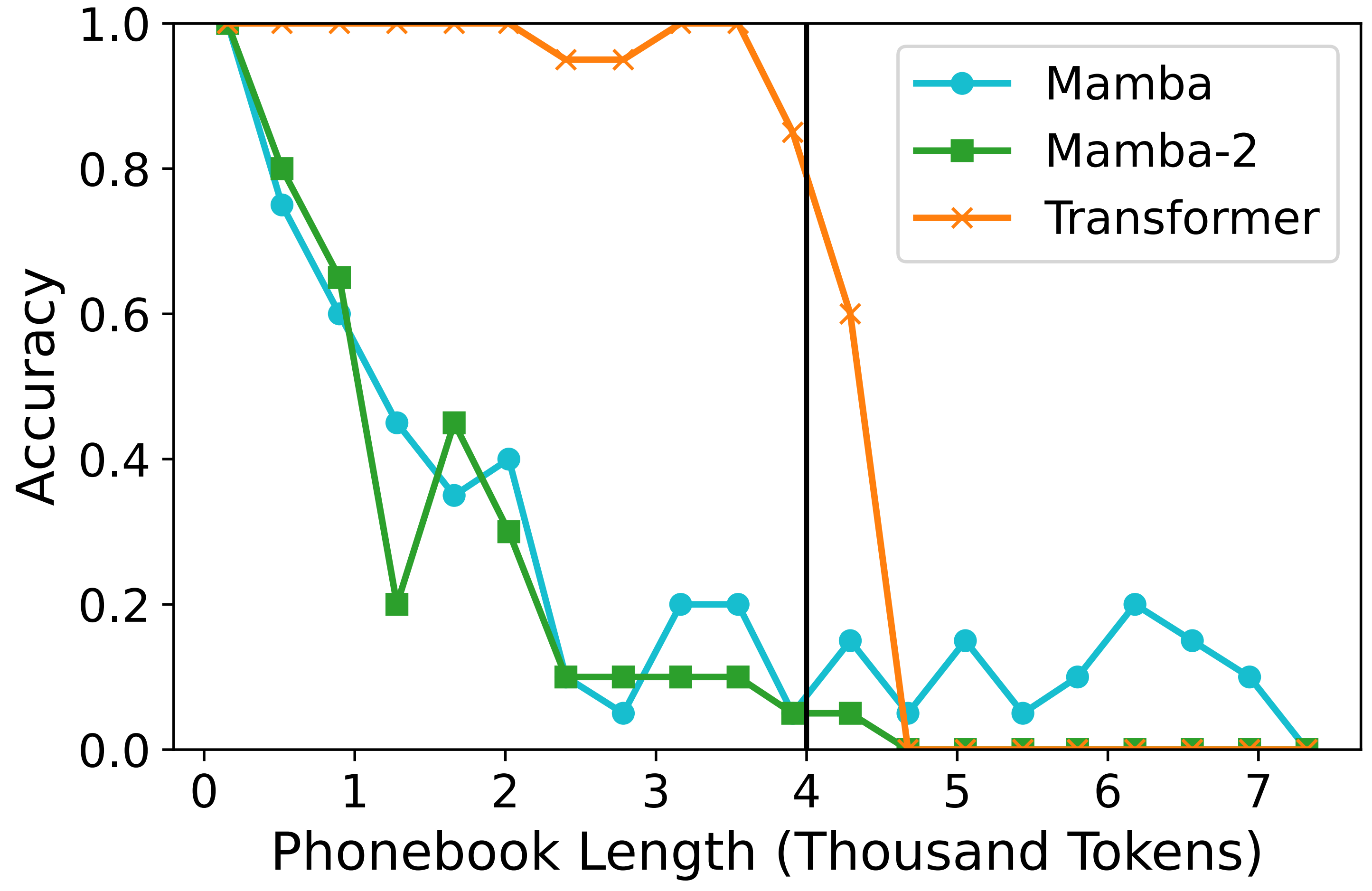
Goal: build HW-aware Memory layers



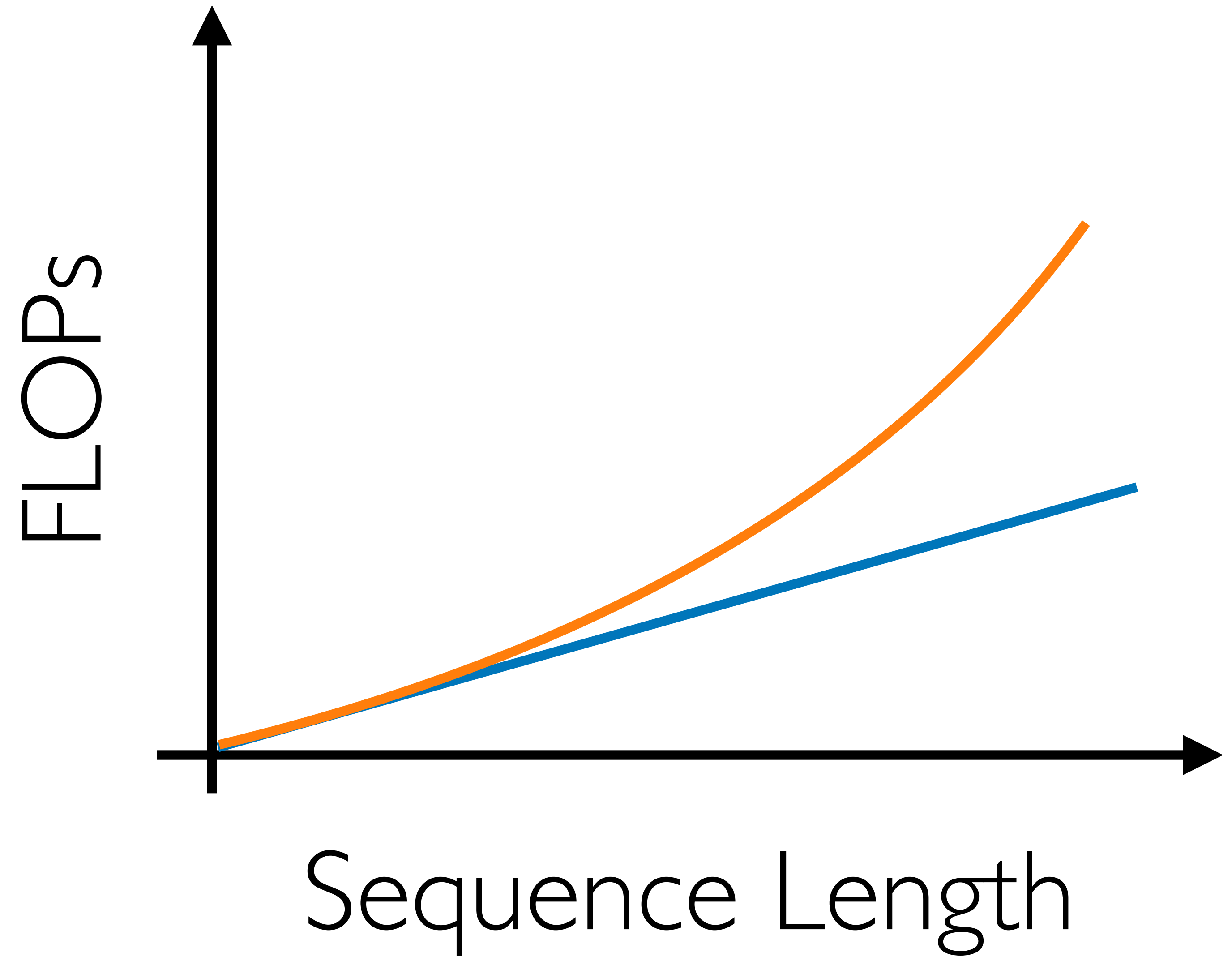
Hybrid Models (Jamba, Zamba, Samba, Griffin, ...)

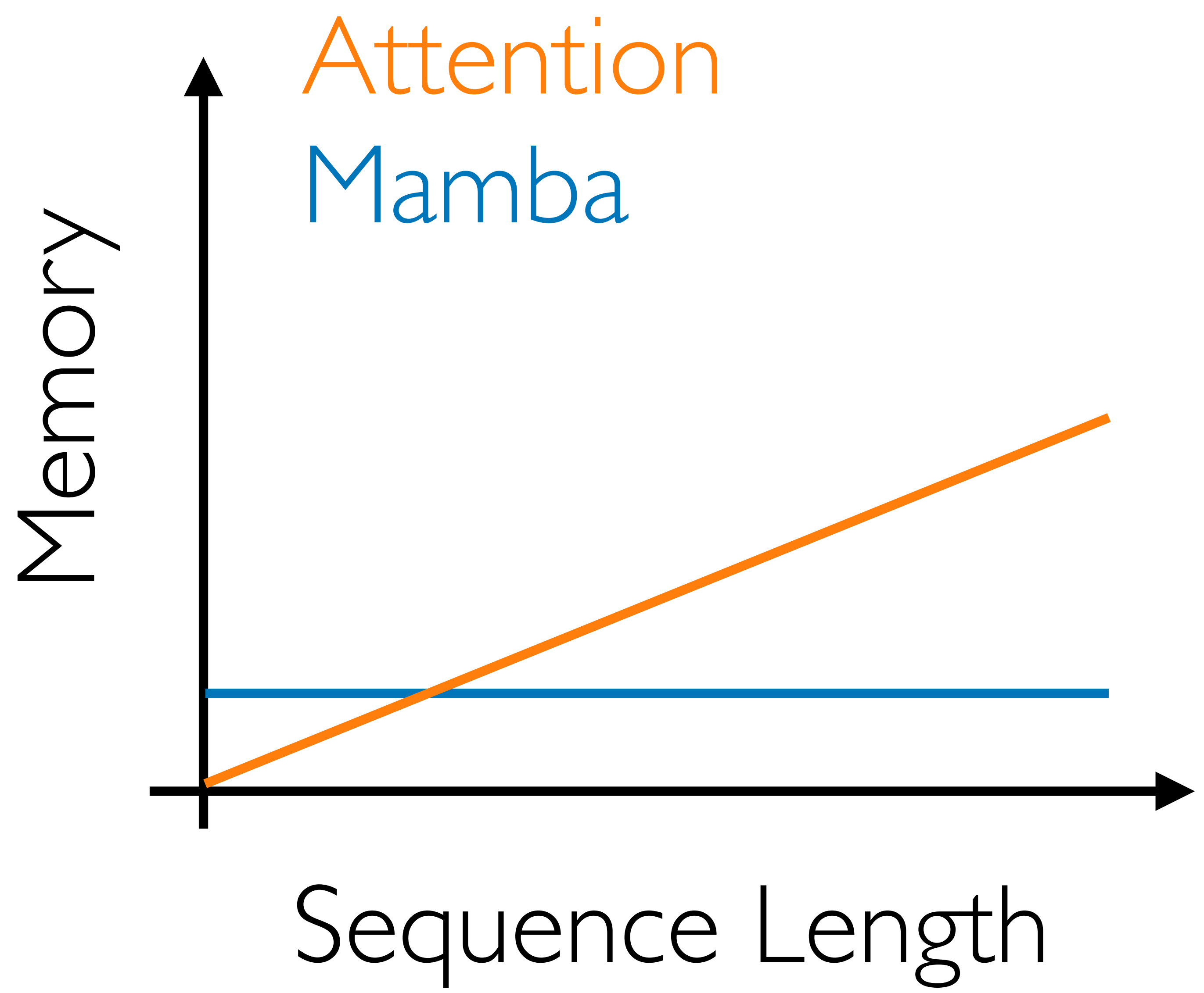
How to build the model's memory? What to keep? What to discard? → *Realization Theory*

Eidetic vs Fading Memory



From [1]



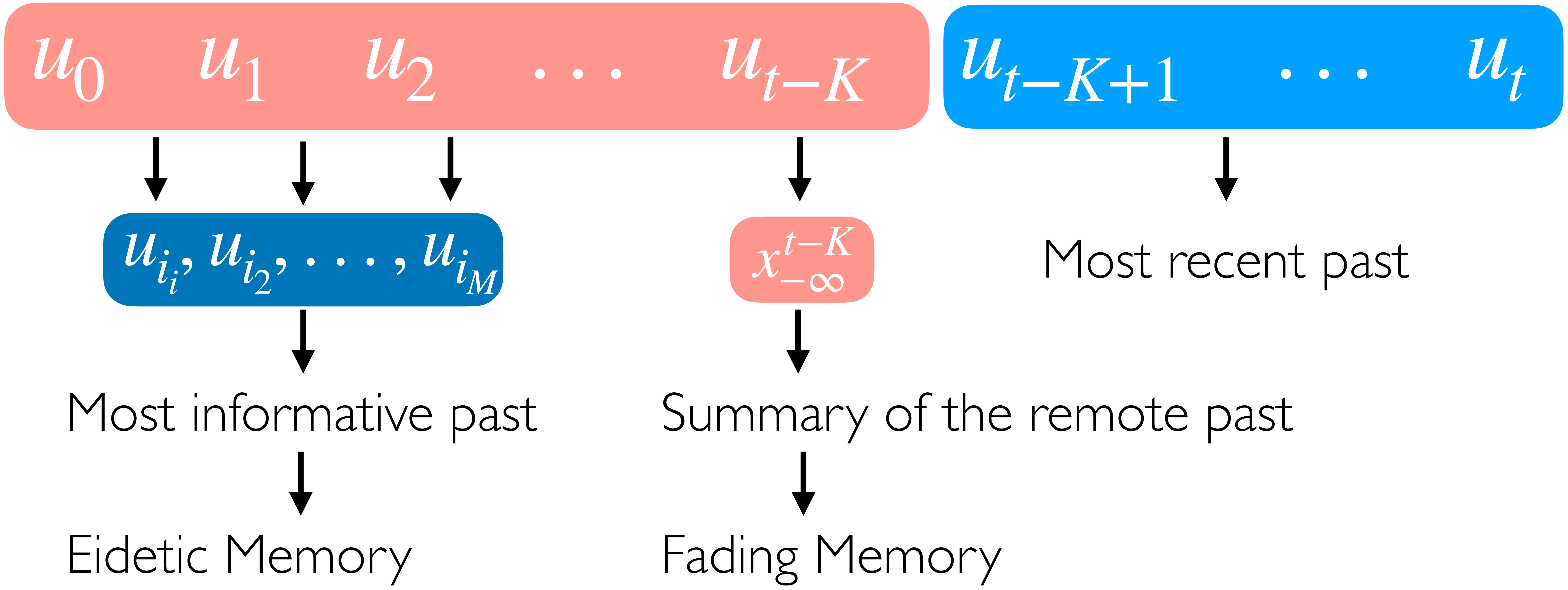


1. Attention: <u>Perfect recall</u>, good long context performance, high compute

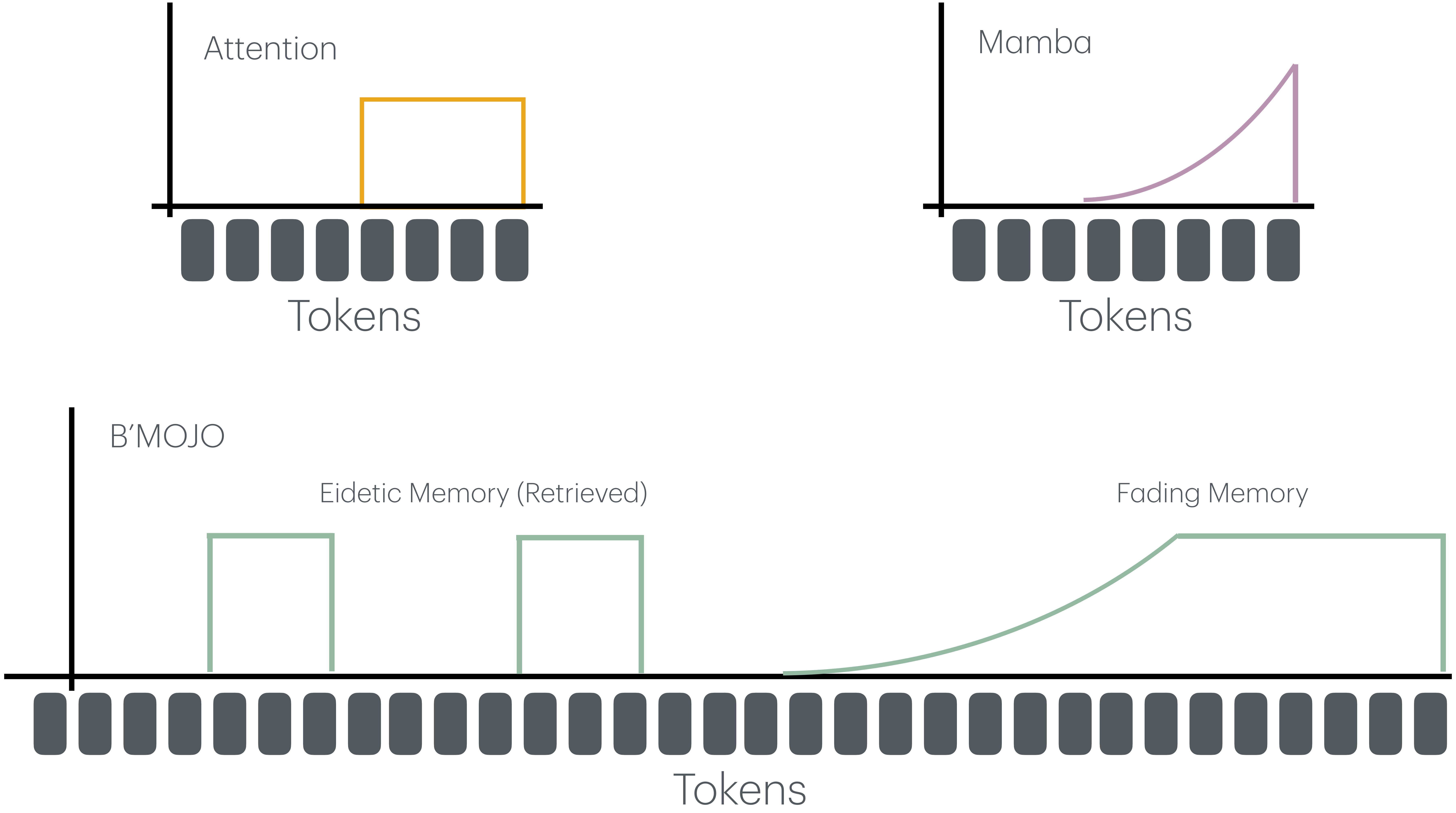
2. SSMs: Low recall, <u>fading memory</u>, low cost

[1] R. Waleffe et al., "An Empirical Study of Mamba-based Language Models"

B'MOJO's key ideas



B'MOJO's Associative Recall

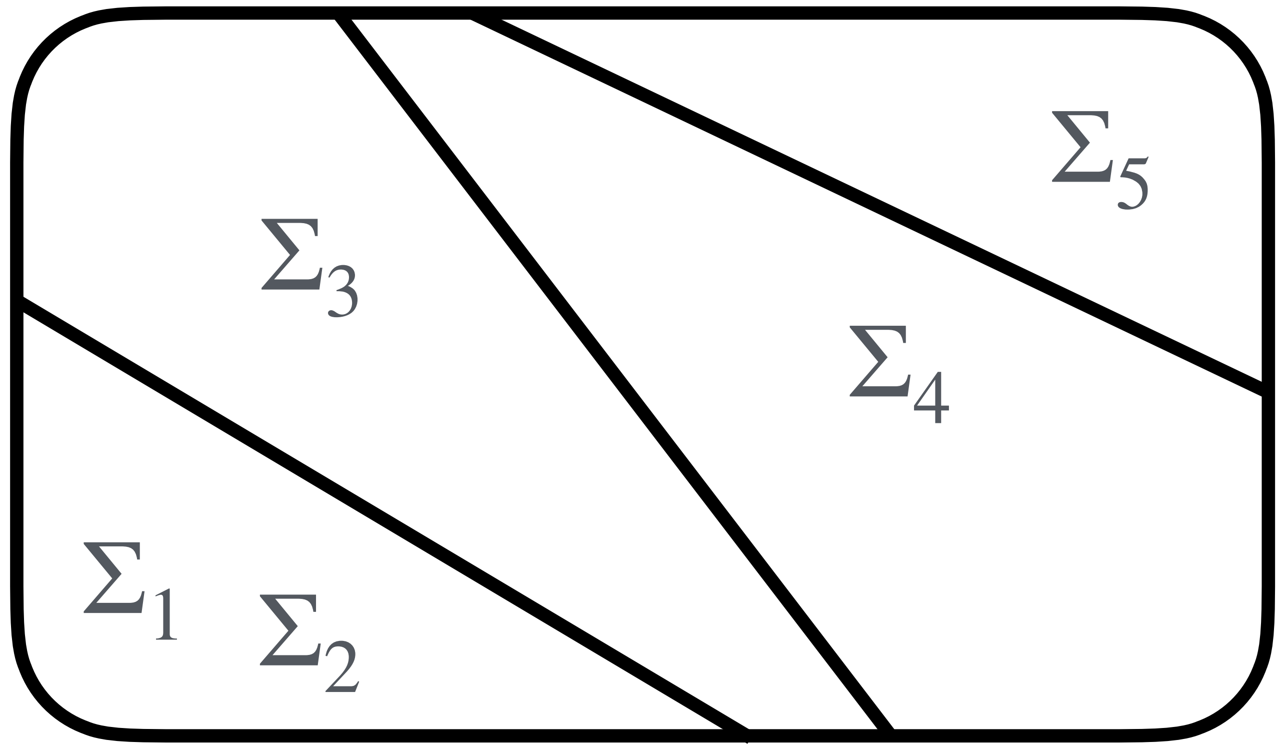


Stochastic Realization Problem

Idea: Find a dynamical model Σ and its state such that it generates a sequence of observations u_1^t and future continuations.

Problem: Even restricting to LTI systems there exist infinitely many realizations of given measurements u_1^t

Thm: Any given data process has not a unique realization, but an equivalence class of models that realize it.



$$\begin{cases} \Sigma_1 \equiv \Sigma_2 \\ \Sigma_1 \neq \Sigma_3 \\ \Sigma_1 \neq \Sigma_4 \end{cases}$$

However, Canonical Realizations are representative of each equivalence class! There are many canonical forms: *Observable, Controllable, Minimal, Balanced*…

Canonical Realizations

$$\Sigma = \begin{cases} x_{t+1} = Ax_t + Bu_t \\ y_t = Cx_t \end{cases}$$

Controllable Canonical From

$$A = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ a_0 & a_1 & \dots & a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad C = [c_0, c_1, \dots, c_{n-1}]$$

Observable Canonical From

$$A = \begin{bmatrix} 0 & & & a_0 \\ 1 & & & a_1 \\ & \ddots & & \\ & & 1 & a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix} \quad C = [0,\dots,0,1]$$

Both have poor numerical properties and are not necessarily minimal (least # of FLOPs).

Minimal Canonical Form (smallest possible state)

Obtained by dropping the non-controllable and non-observable subspace.

Balanced Canonical Form

Obtained by equalizing the energy required to control and observe the state.

Canonical Realizations

Nilpotent Model (in Controllable Canonical Form)

$$A = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \qquad C = [c_0, c_1, \dots, c_{n-1}]$$

All poles in A are equal to zero.

Using an input dependent non-linear readout function (softmax) we get **Causal Attention**

Fading Memory Model

$$A = \begin{bmatrix} a_0 & & & \\ & \ddots & & \\ & & a_{n-2} & \\ & & & a_{n-1} \end{bmatrix} \qquad B = \begin{bmatrix} b_0 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix} \qquad C = [c_0, c_1, \dots, c_{n-1}]$$

Poles of $A < 1$

Thm: Differently from Fading memory systems, Nilpotent systems are not diagonalizable.

Modern Realizations (Attention/SSMs)

Causal Attention

$$\begin{cases} x_{t+1} = A_{ATT}x_t + B_{ATT}u_t \\ y_t = \text{softmax}(u_t, x_t) \end{cases}$$

$$A_{ATT} = \begin{bmatrix} 0 & I & & \\ & & \ddots & \\ & & & I \\ & & & 0 \end{bmatrix}; \quad B_{ATT} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \begin{bmatrix} k_t \\ v_t \end{bmatrix} \end{bmatrix}$$

Attention only has **short term eidetic memory** that is deadbeat in N steps.

SSMs (e.g Mamba or Linear Attention variants)

$$\begin{cases} x_{t+1} = \bar{A}(u_t)x_t + \bar{B}(u_t)u_t \\ y_t = \bar{C}(u_t)x_t \end{cases}$$

$$\bar{A}(u_t) = \begin{bmatrix} \bar{a}_0(u_t) & & & \\ & \ddots & & \\ & & \bar{a}_{n-2}(u_t) & \\ & & & \bar{a}_{n-1}(u_t) \end{bmatrix} \quad \bar{B}(u_t) = \begin{bmatrix} \bar{b}_0(u_t) \\ \vdots \\ \bar{b}_{n-2}(u_t) \\ \bar{b}_{n-1}(u_t) \end{bmatrix} \quad \bar{C}(u_t) = [c_0(u_t), c_1(u_t), \ldots, c_{n-1}(u_t)]$$

Only has fading memory with decoupled dynamics (cannot retain information indefinitely).

B'MOJO's realization

Idea: B'MOJO layers generalize Nilpotent dynamics (of AR Transformers) and fading diagonal dynamics (of Mamba/Linear Attention).

B'MOJO

$$\begin{cases} x_{t+1} = A(u_t)x_t + B(u_t)u_t \\ y_t = \text{softmax}(u_t, x_t) \end{cases}$$

$$A(u_t) = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ a_0(u_t) & a_1(u_t) & \dots & a_{n-1}(u_t) \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b(u_t) \end{bmatrix}$$

Note: B'MOJO has a non-diagonal input dependent dynamics that realizes any dynamical layer (similar to the *Hammerstein-Wiener* model in the Control literature).

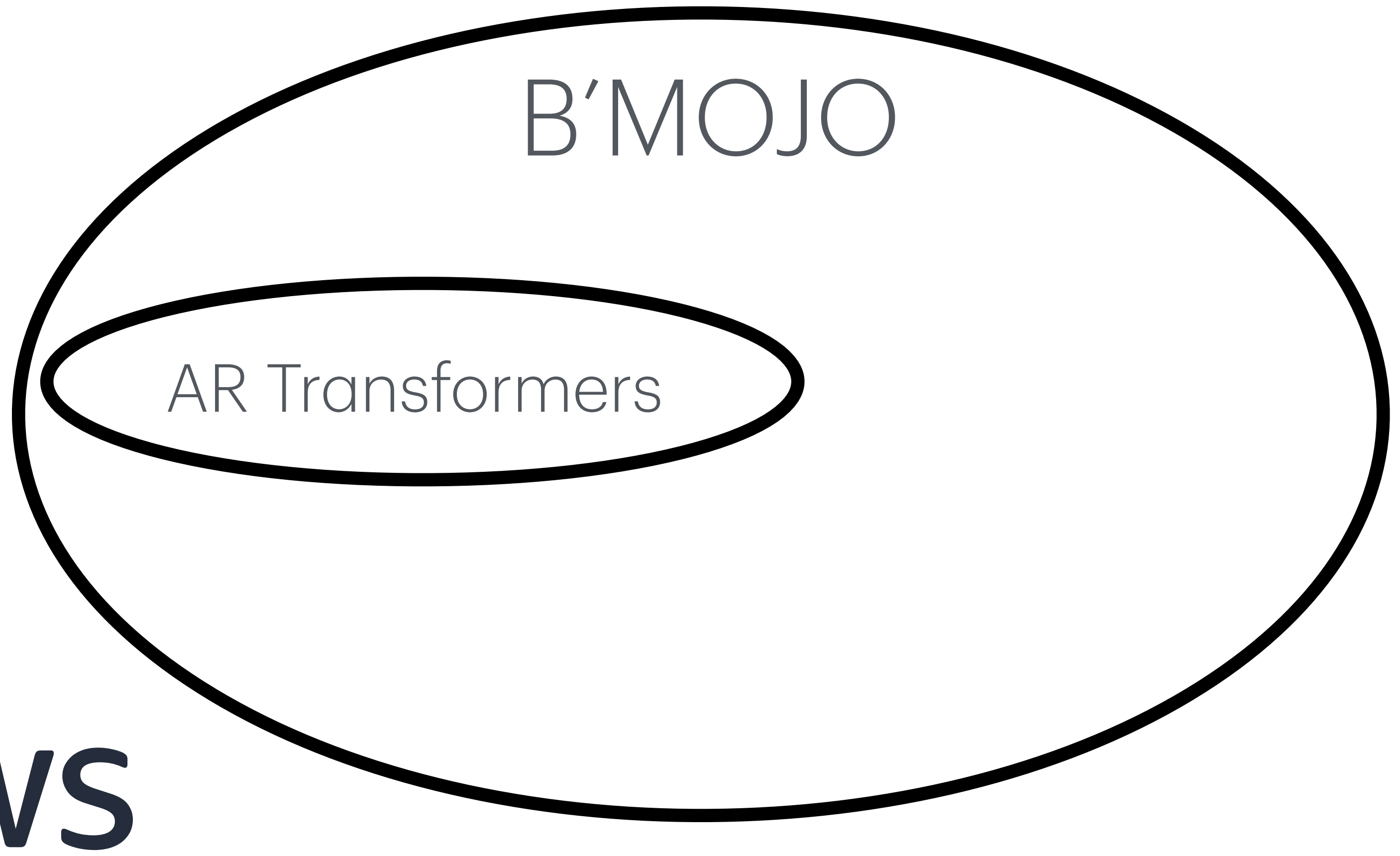
B’MOJO generalizes Transformers

Idea: B’MOJO layers generalize Nilpotent dynamics (of AR Transformers) and fading diagonal dynamics (of Mamba/Linear Attention).

B’MOJO

$$\begin{cases} x_{t+1} = A(u_t)x_t + B(u_t)u_t \\ y_t = \text{softmax}(u_t, x_t) \end{cases}$$

$$A(u_t) = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ a_0(u_t) & a_1(u_t) & \dots & a_{n-1}(u_t) \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b(u_t) \end{bmatrix}$$



Attention Layer

$$\begin{cases} z_{t+1} = A_{ATT} z_t + B_{ATT}(u_t) \\ x_t = softmax(u_t, z_t) \end{cases}$$

$$A_{ATT} = \begin{bmatrix} 0 & I & & \\ & & \ddots & \\ & & & I \\ 0 & 0 & \dots & 0 \end{bmatrix}; \quad B_{ATT} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \begin{bmatrix} k_t \\ v_t \end{bmatrix} \end{bmatrix}$$

Set last row to zero

Row set to zero

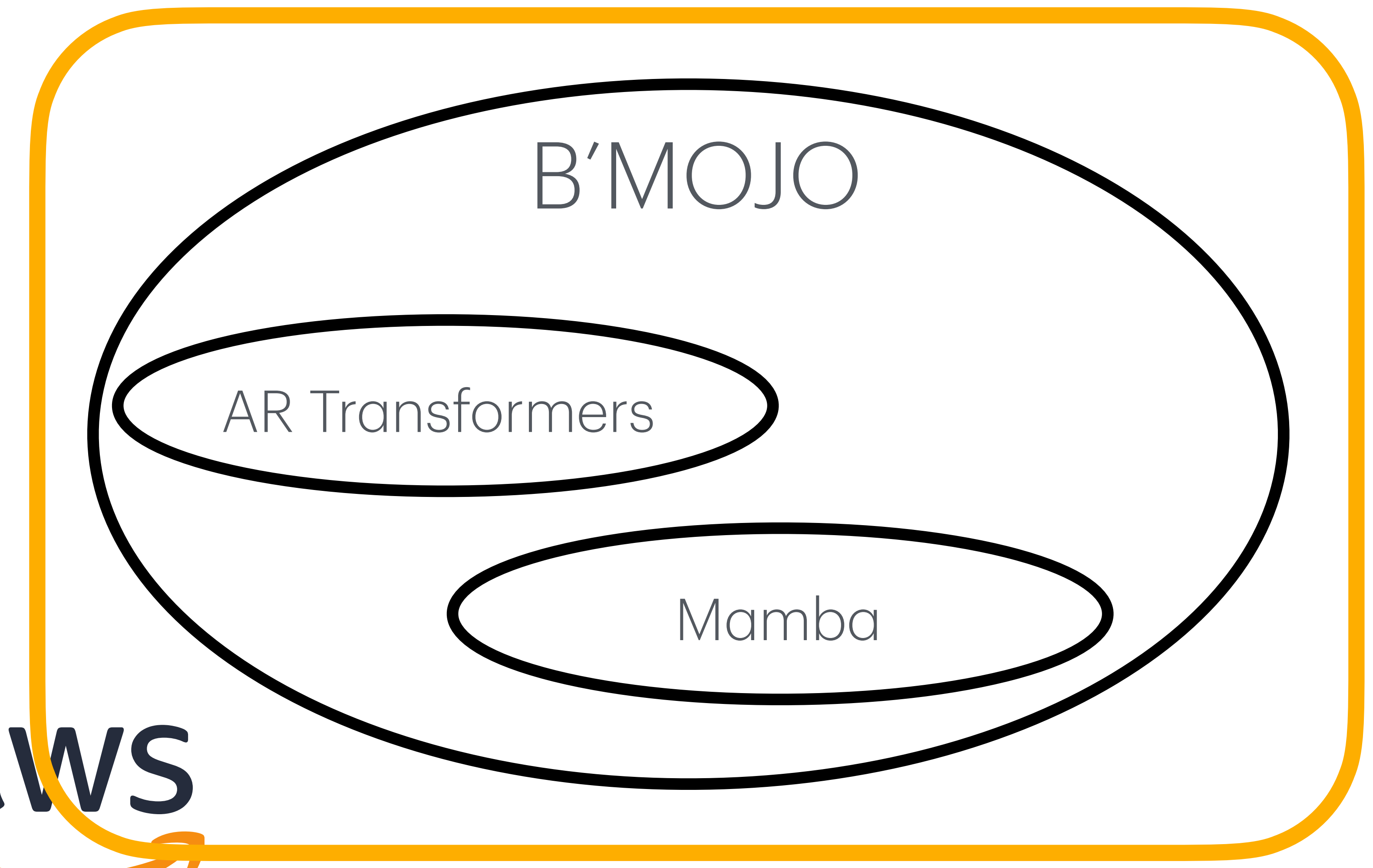
B'MOJO is strictly more expressive than SSMs

Idea: B'MOJO layers generalize Nilpotent dynamics (of AR Transformers) and fading diagonal dynamics (of Mamba/Linear Attention).

B'MOJO

$$\begin{cases} x_{t+1} = A(u_t)x_t + B(u_t)u_t \\ y_t = \text{softmax}(u_t, x_t) \end{cases}$$

$$A(u_t) = \begin{bmatrix} 0 & 1 & & \\ & & \ddots & \\ & & & 1 \\ a_0(u_t) & a_1(u_t) & \dots & a_{n-1}(u_t) \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b(u_t) \end{bmatrix}$$

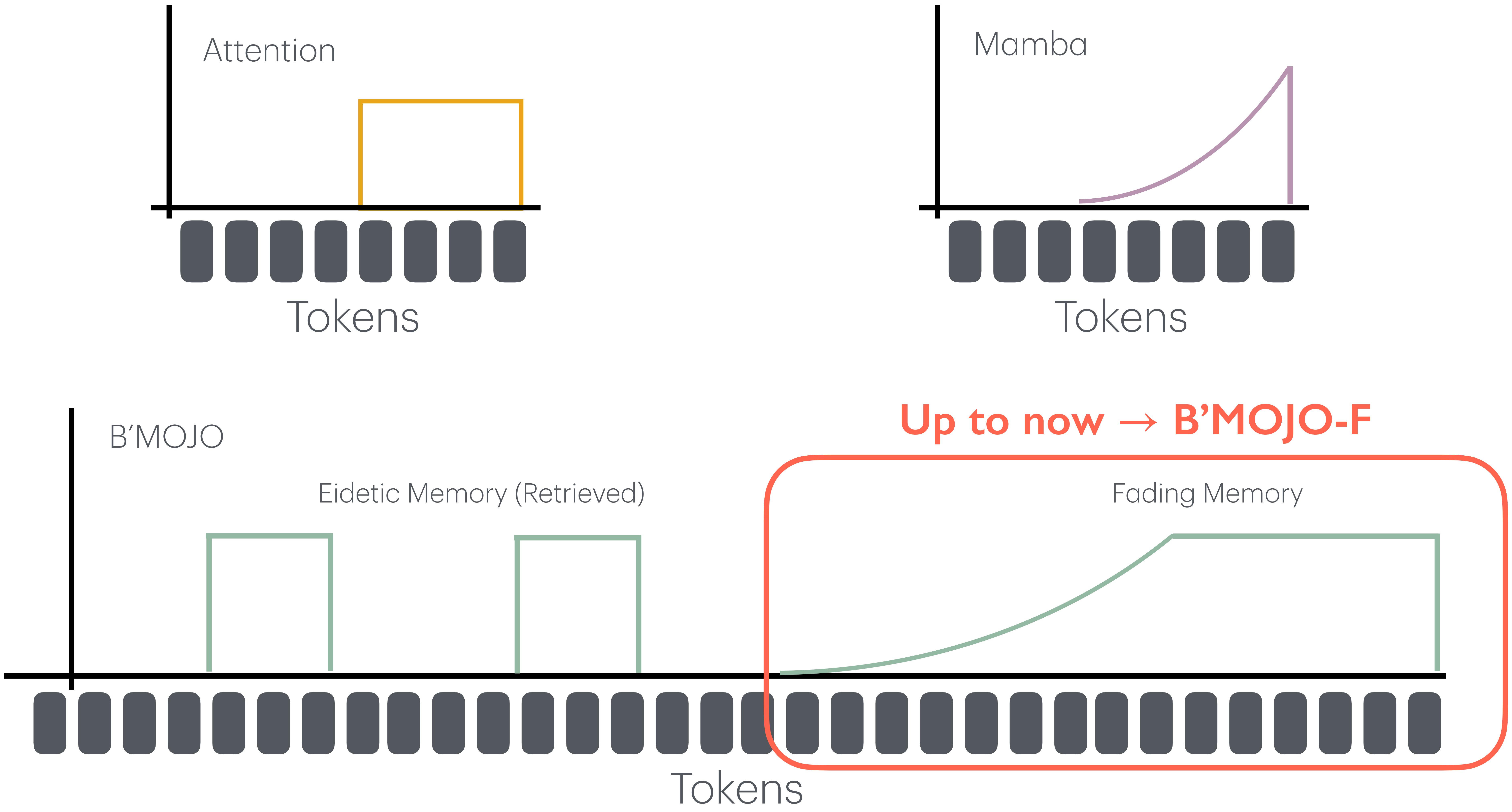


Mamba

$$\bar{A}(u_t) = \begin{bmatrix} \bar{a}_0(u_t) & & & \\ & \ddots & & \\ & & \bar{a}_{n-2}(u_t) & \\ & & & \bar{a}_{n-1}(u_t) \end{bmatrix} \qquad \bar{B}(u_t) = \begin{bmatrix} \bar{b}_0(u_t) \\ \vdots \\ \bar{b}_{n-2}(u_t) \\ \bar{b}_{n-1}(u_t) \end{bmatrix}$$

B`MOJO has a **non-diagonal input-dependent dynamics**, more expressive than Mamba.

B'MOJO's Associative Recall



Innovation Selection (Adaptive compression)

B'MOJO-F's memory is: $M_t := [\boxed{x^t_{-\infty}}, \boxed{u_{t-K-1}, \ldots, u_{t-K}}]$

Fading State Last K-tokens

Problem: Fading memory access to older information (through the fading state).

Idea: Store in the eidetic memory tokens that the state cannot easily predict (adaptive compression).

$$M_t \leftarrow \begin{cases} M_{t-1} \cup \{u_t, \epsilon_t\} & \textbf{if } \epsilon_t > \min_{\epsilon \in M_{t-1}}(\epsilon) \\ M_{t-1} & \textbf{otherwise} \end{cases}$$

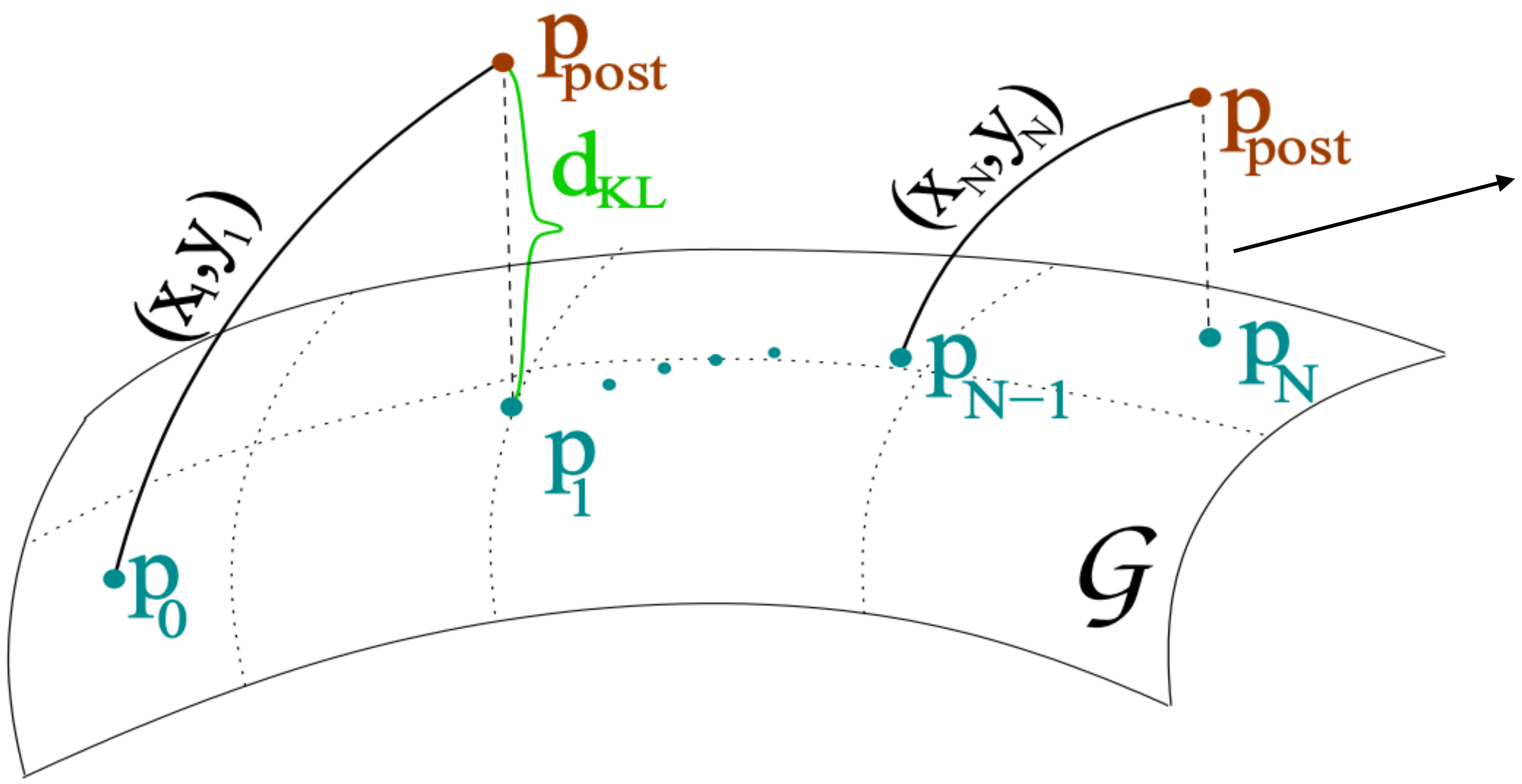
where $$\epsilon_t := ||\hat{y}_t(M_t) - y_t||^2$$

B'MOJO's memory is: $M_t := [\boxed{u_{i_1}, \ldots, u_{i_M}}, \boxed{x^t_{-\infty}}, \boxed{u_{t-K-1}, \ldots, u_{t-K}}]$

Eidetic Memory

Connection with Online Kernel Regression

Idea: Online update of the set of basis functions (kernel sections) $K(\cdot, x_i)$ when the prediction residual exceeds a threshold.

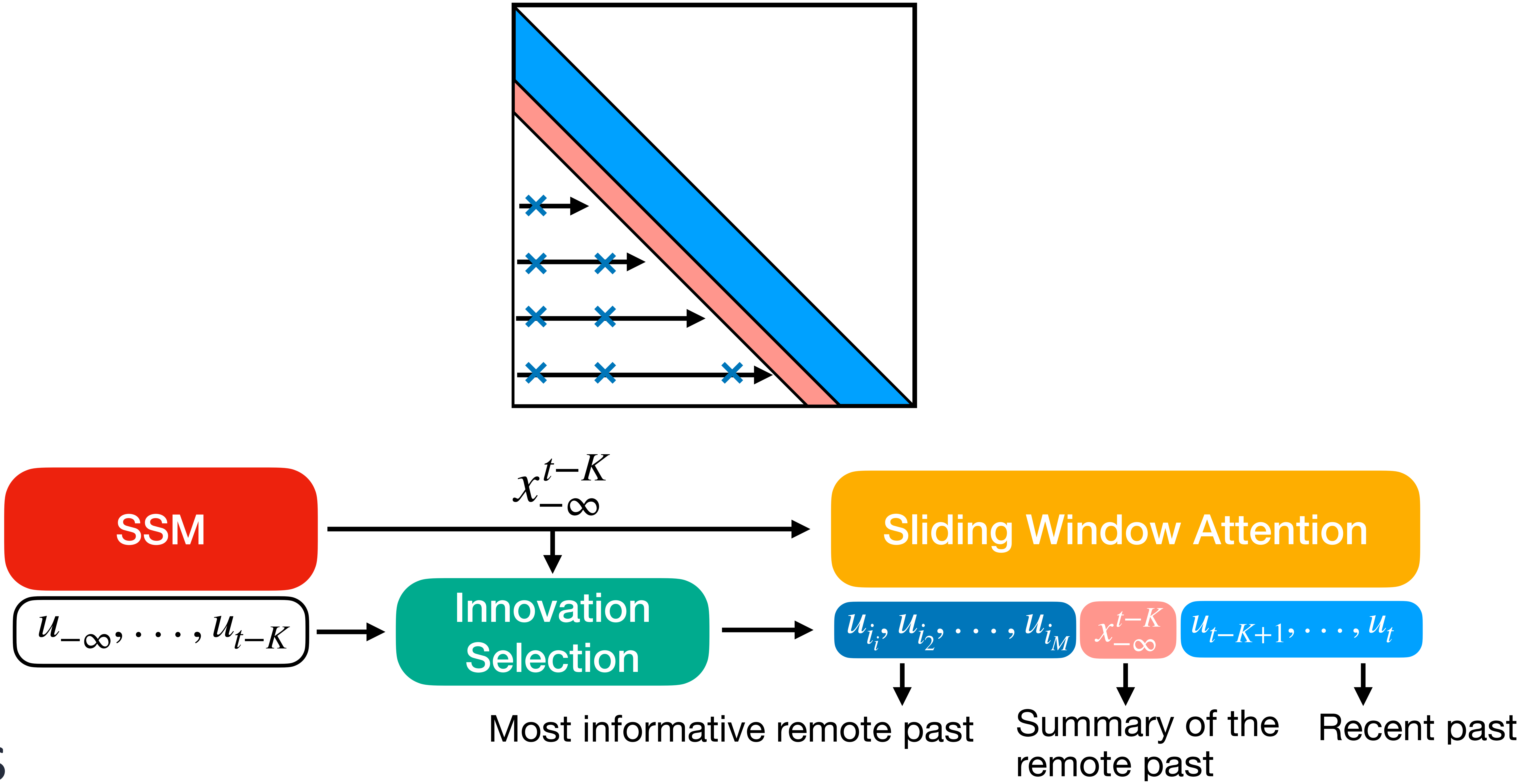


$$\left\| K_0(\cdot, x_{t+1}) - \sum_{i=1}^{t} \hat{e}_{t+1}(i) K_0(\cdot, x_i) \right\|^2$$

Figure 1: Visualisation of the online approximation of the untractable posterior process. The resulting approximate process from previous iteration is used as prior for the next one.

L. Csató and M. Opper, "Sparse Online Gaussian Processes", 2002

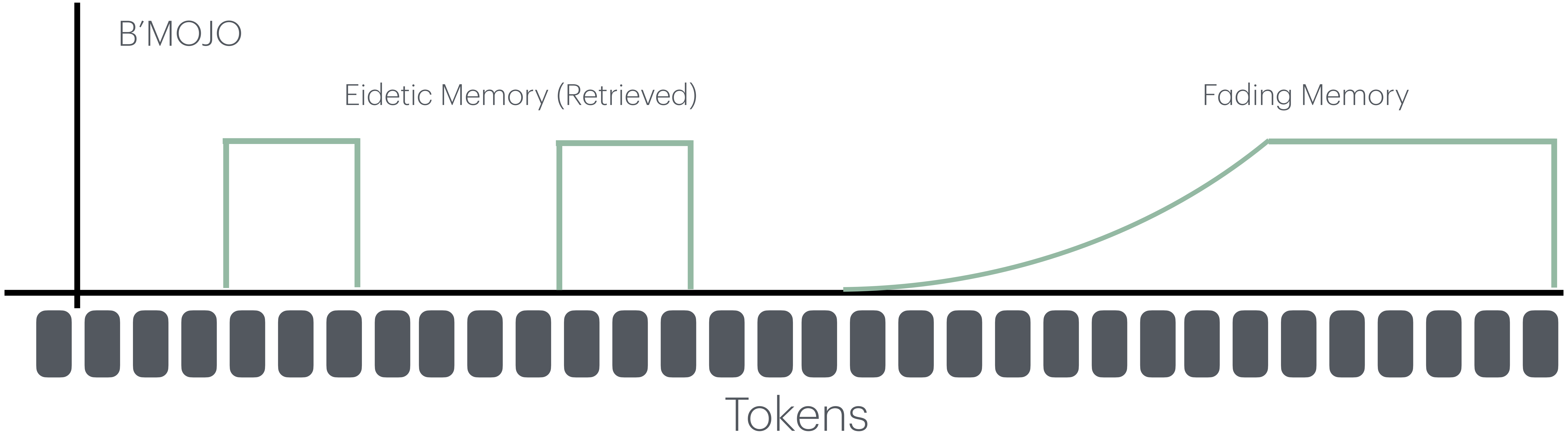
B'MOJO's Minimal Realization



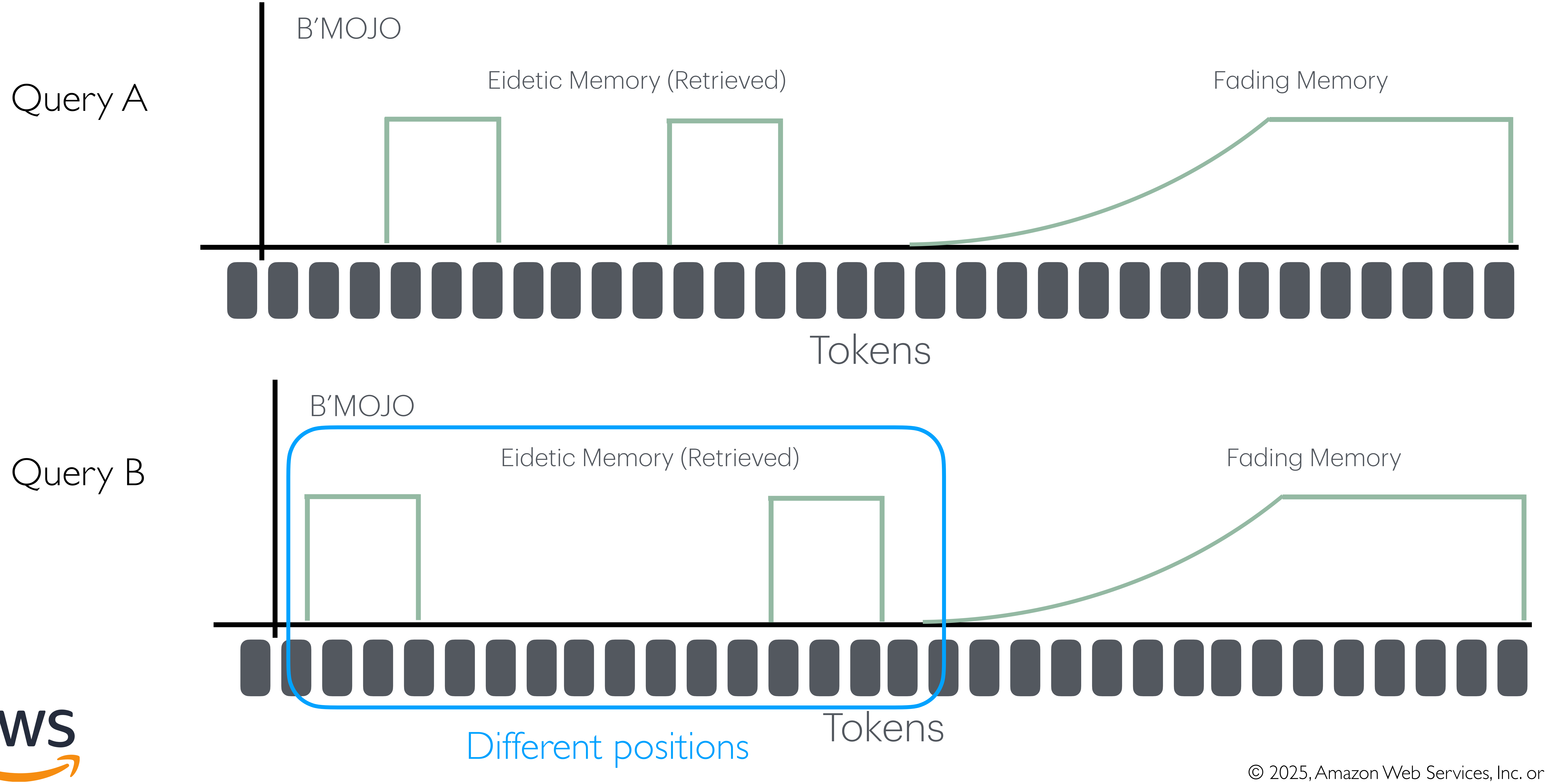
B'MOJO summary

B'MOJO's state can be dynamically allocated as required by the a priori unknown complexity of the observations

We augment B'MOJO's state with an eidetic memory implemented with *shifting registers* similar to Sliding Window Attention but based on an *Innovation Test* *(à la Box-Ljung)* rather than recency.



One problem...


Query A
B'MOJO
Eidetic Memory (Retrieved)
Fading Memory
Tokens
Query B
B'MOJO
Eidetic Memory (Retrieved)
Fading Memory
Tokens
Different positions

Can we further extend the selection span?

Expansion Span: Combining Fading Memory and Retrieval in Hybrid State Space Models

Elvis Nunez[†*] Luca Zancato[‡] Benjamin Bowman[‡]
Aditya Golatkar[‡] Wei Xia[‡] Stefano Soatto[‡]

[†]UCLA [‡]AWS AI Labs

elvis.nunez@ucla.edu, {zancato, bowmaben, agolatka, wxia, soattos}@amazon.com

[cs.CL] 17 Dec 2024

Abstract

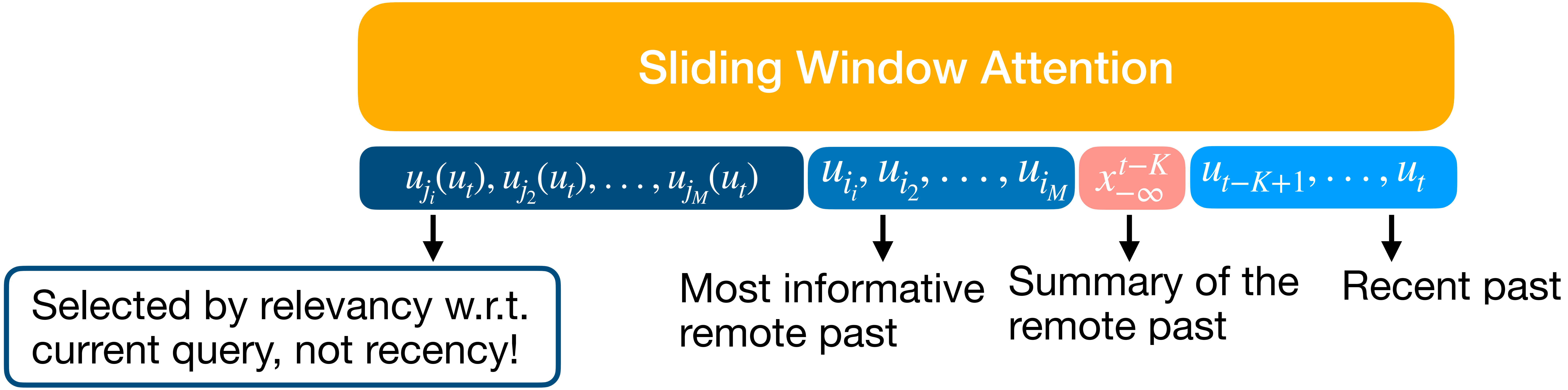
The 'state' of State Space Models (SSMs) represents their memory, which fades exponentially over an unbounded span. By contrast, Attention-based models have 'eidetic' (i.e., verbatim, or photographic) memory over a finite span (context size). Hybrid architectures combine State Space layers with Attention, but still cannot recall the distant past and can access only the most recent tokens eidetically. Unlike current methods of combining SSM and Attention layers, we allow the state to be allocated based on relevancy

1. Introduction

State Space Models are able to process sequences with an unbounded number of tokens by maintaining a fixed-size state. However, this state is lossy and information about early tokens 'fades' as more inputs are processed. In contrast, Transformer models have a state determined by the number of tokens in their input sequence and are able to access information from all past tokens in their context 'eidetically.' However, they do so at the cost of extra compute and memory. Recent Hybrid models [8, 12, 38] augment SSMs with Attention layers in an effort to counteract SSMs'

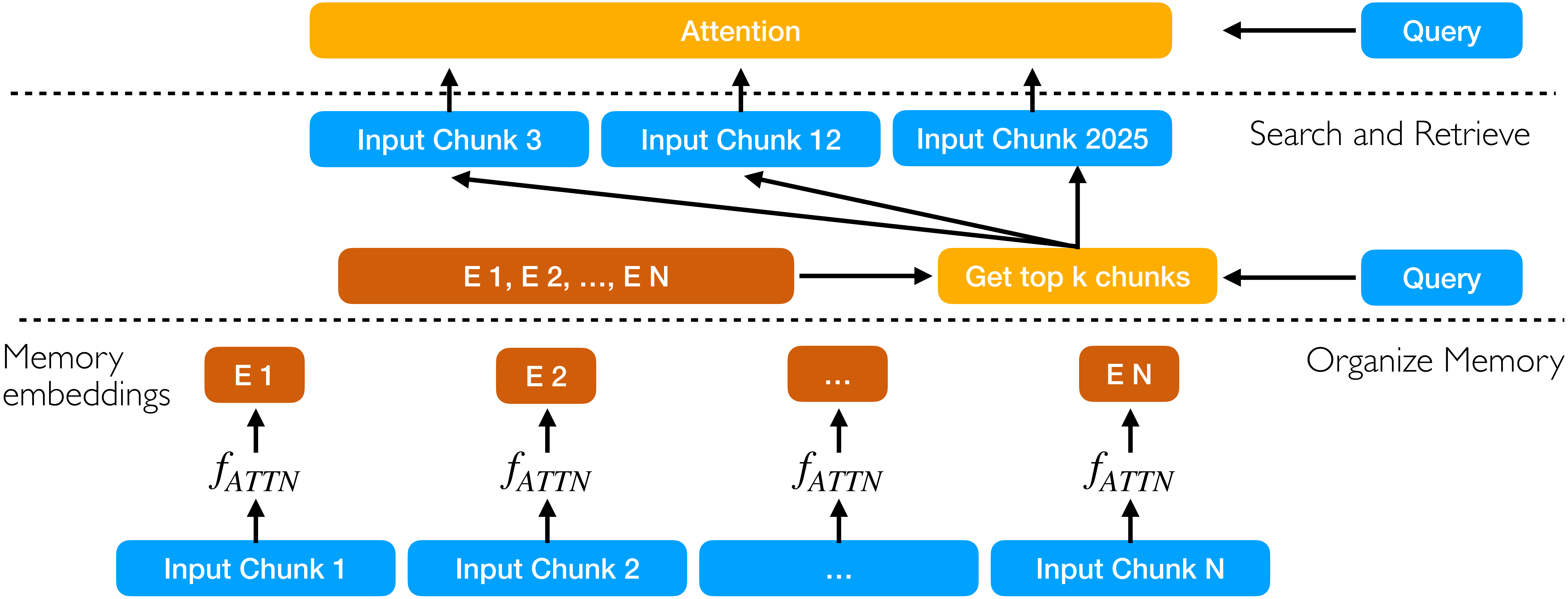
Can we further extend the selection span?

Question: Can we dynamically expand B'MOJO's Eidetic memory up to HW limitation?



We propose a Sparse Attention implementation (called *Span-Expanded Attention*) for Hybrid models.

Span-Expanded Attention



DeepSeek's Native Sparse Attention

deepseek

Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention

Jingyang Yuan[*1,2], Huazuo Gao[1], Damai Dai[1], Junyu Luo[2], Liang Zhao[1], Zhengyan Zhang[1], Zhenda Xie[1], Y. X. Wei[1], Lean Wang[1], Zhiping Xiao[3], Yuqing Wang[1], Chong Ruan[1], Ming Zhang[2], Wenfeng Liang[1], Wangding Zeng[1]

[1]DeepSeek-AI
[2]Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, PKU-Anker LLM Lab
[3]University of Washington
{yuanjy, mzhang_cs}@pku.edu.cn, {zengwangding, wenfeng.liang}@deepseek.com

27 Feb 2025

Abstract

Long-context modeling is crucial for next-generation language models, yet the high compu-

Expansion Span vs DeepSeek's Native Sparse Attention

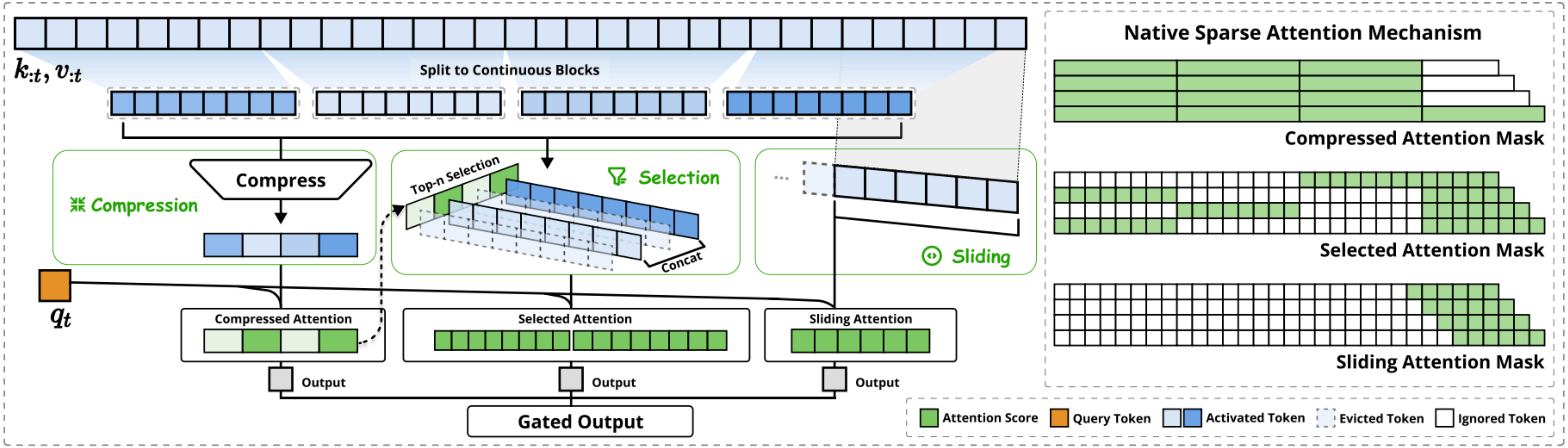


Figure 2 | Overview of NSA's architecture.

J. Yuan et al., "Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention", 2025 February

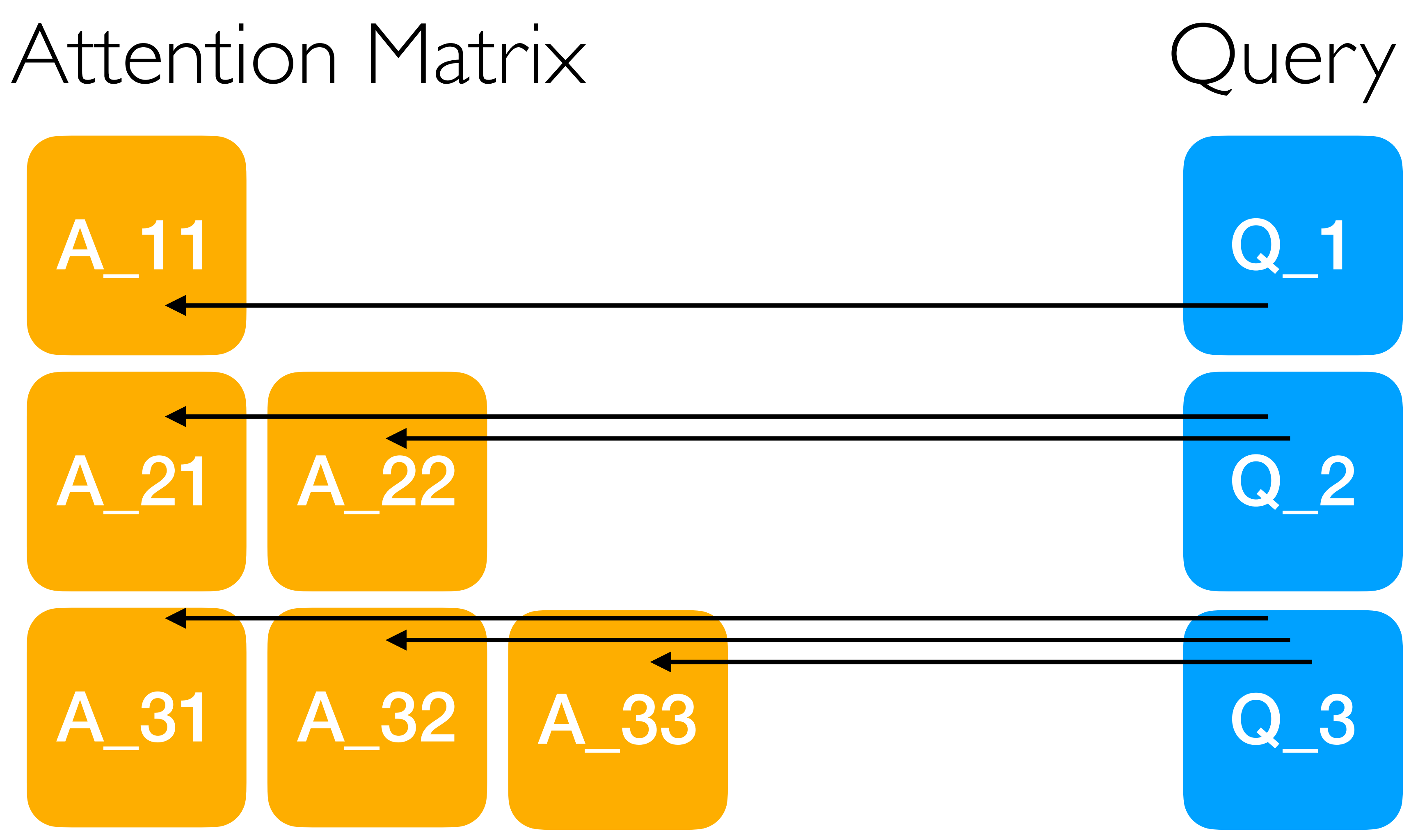
Expansion Span vs DeepSeek's Native Sparse Attention

	Ours	NSA
KV Compression	$A_j^{\text{Mem}} = \text{softmax}\left(\frac{Q_j^{\text{Mem}}(K_j^{\text{Mem}})^T}{\sqrt{d_{\text{model}}}}\right) V_j^{\text{Mem}} \in \mathbb{R}^{S \times d_{\text{model}}}$ $c_j = \frac{1}{S}\sum_{t=1}^{S}(A_j^{\text{Mem}})_t \in \mathbb{R}^{d_{\text{model}}}$	$\tilde{K}_t^{\text{cmp}} = f_K^{\text{cmp}}(\mathbf{k}_{:t}) = \left\{\varphi(\mathbf{k}_{id+1:id+l}) \,\middle\|\, 1 \leqslant i \leqslant \left\lfloor \frac{t-l}{d} \right\rfloor\right\}$
Importance Scores	$R_{ij} = \sum^{M}(Q_i c_j)_t$ $\tilde{R}_i = \text{softmax}\left(\frac{1}{\sqrt{d_{\text{model}}}}(R_i + \mathcal{M}_i)\right) \in \mathbb{R}^U$	$\mathbf{p}_t^{\text{cmp}} = \text{Softmax}\left(\mathbf{q}_t^T \tilde{K}_t^{\text{cmp}}\right)$
Rank and Retrieve	$\tilde{K} = \text{Concatenate}(K^{\text{Mem}}_{\phi_i(U)_1}, \ldots, K^{\text{Mem}}_{\phi_i(U)_k}, K_i)$ $\tilde{V} = \text{Concatenate}(V^{\text{Mem}}_{\phi_i(U)_1}, \ldots, V^{\text{Mem}}_{\phi_i(U)_k}, V_i)$	$\mathcal{I}_t = \{i \mid \text{rank}(\mathbf{p}_t^{\text{slc}'}[i]) \leqslant n\}$ $\tilde{K}_t^{\text{slc}} = \text{Cat}\left[\{\mathbf{k}_{il'+1:(i+1)l'} \mid i \in \mathcal{I}_t\}\right]$
Attention	$A_i^{\text{SE-Attn}} = \mathit{Attention}(Q_i, \tilde{K}_i, \tilde{V}_i)$ $o^{\text{SE-Attn}} = \text{Concatenate}(A_1^{\text{SE-Attn}}, A_2^{\text{SE-Attn}}, \ldots, A_T^{\text{SE-Attn}})$	$\mathbf{o}_t^* = \sum_{c \in C} g_t^c \cdot \text{Attn}(\mathbf{q}_t, \tilde{K}_t^c, \tilde{V}_t^c)$ $(\tilde{K}_t^{\text{cmp}}, \tilde{V}_t^{\text{cmp}}; \tilde{K}_t^{\text{slc}}, \tilde{V}_t^{\text{slc}};$ and $\tilde{K}_t^{\text{win}}, \tilde{V}_t^{\text{win}})$

Hardware aware implementation

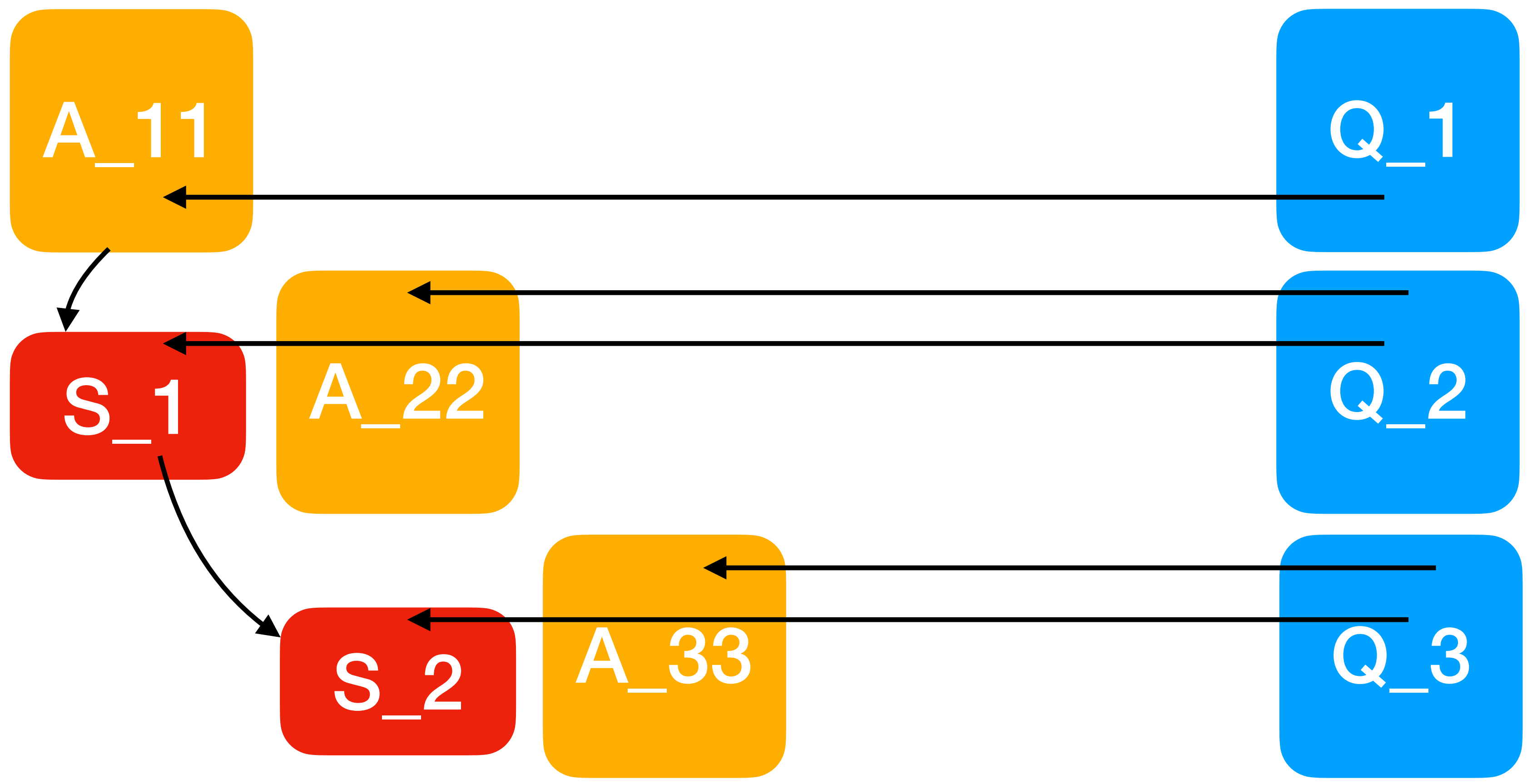
Efficient chunked implementations

FlashAttention



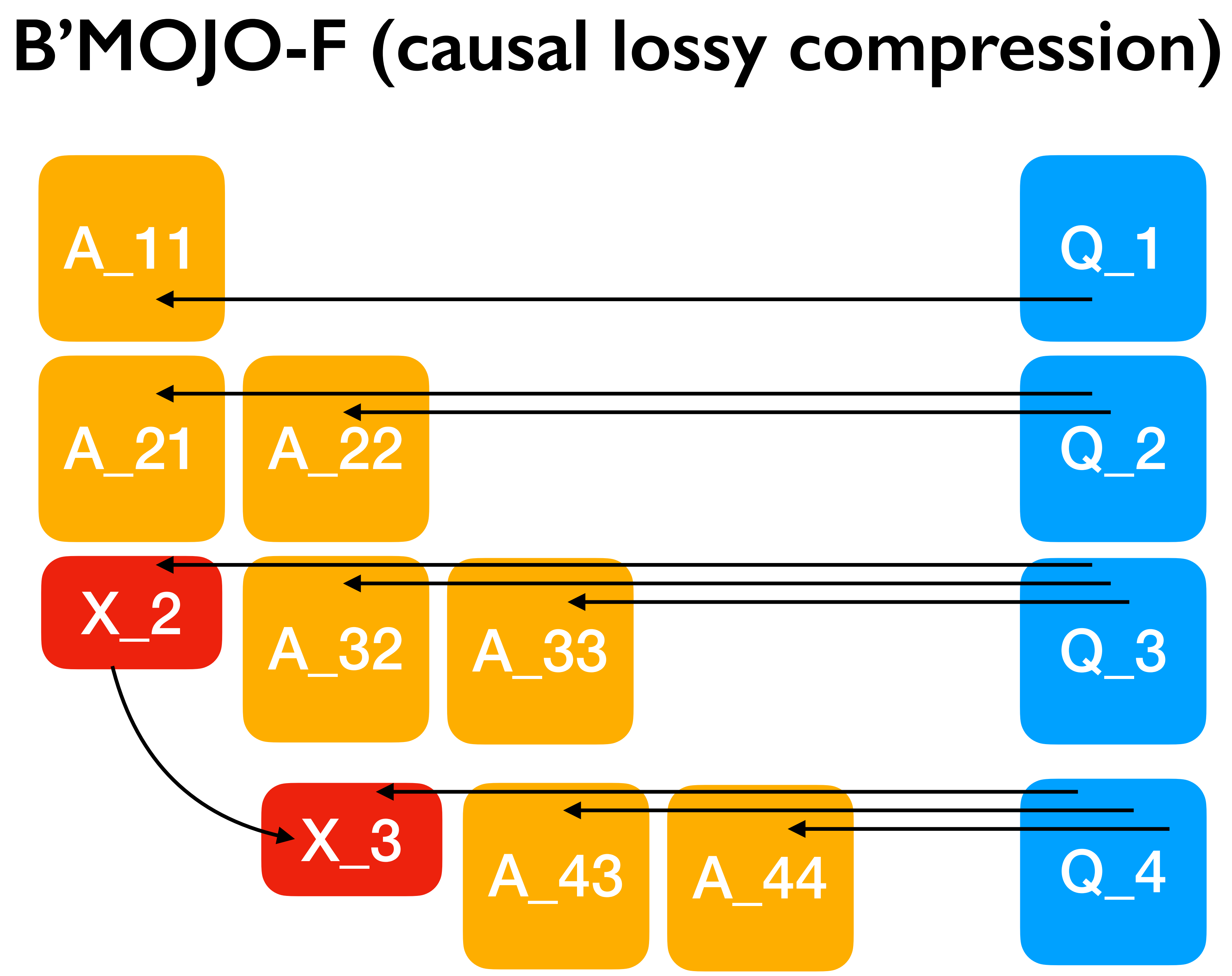
$O(N^2)$ memory reads, 2 for loops

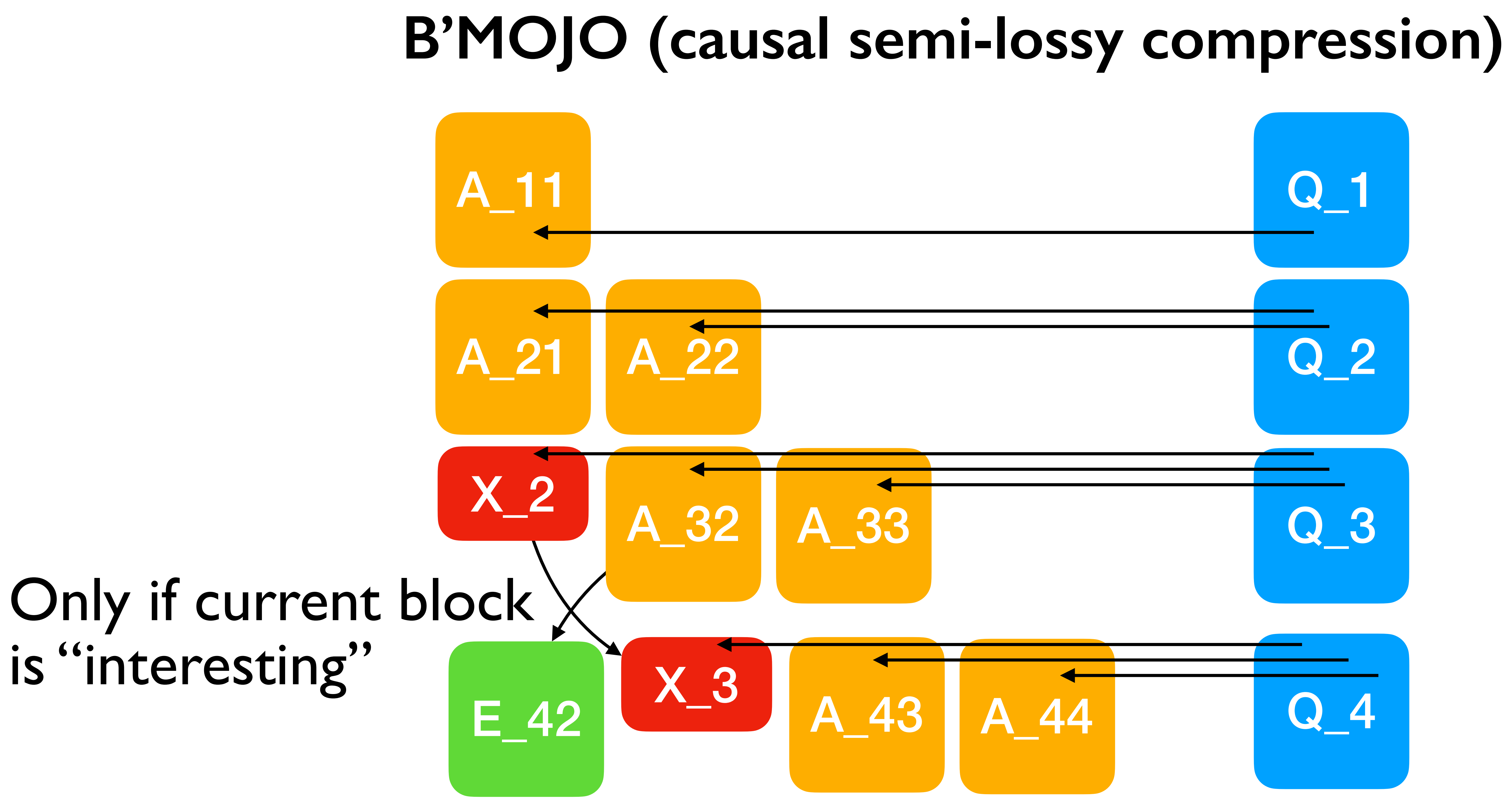
Mamba/Linear Attention



$O(N)$ memory reads, 1 for loop

Can we avoid costly tiles if they don't add new info?



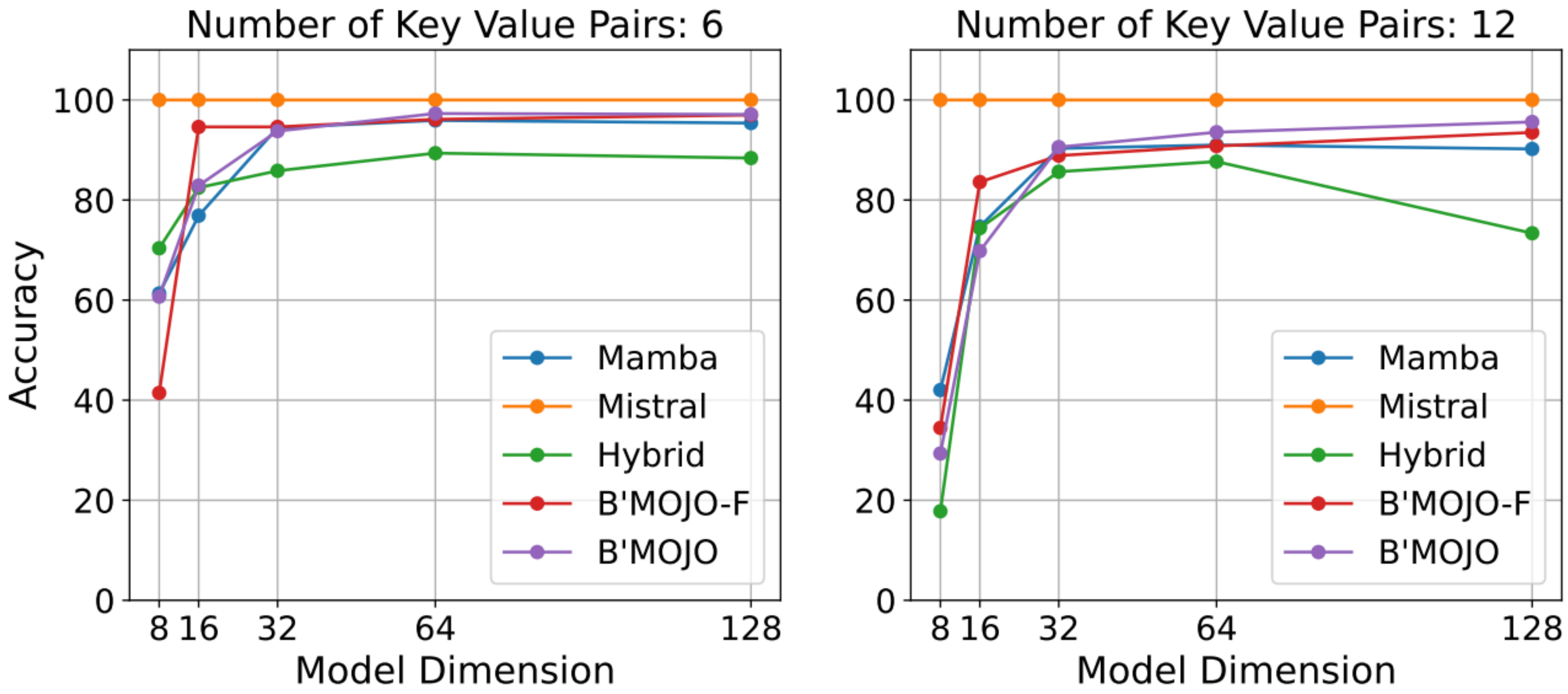


$O(N)$ memory reads, 2 for loops (smaller than FlashAttention)

Experimental Results

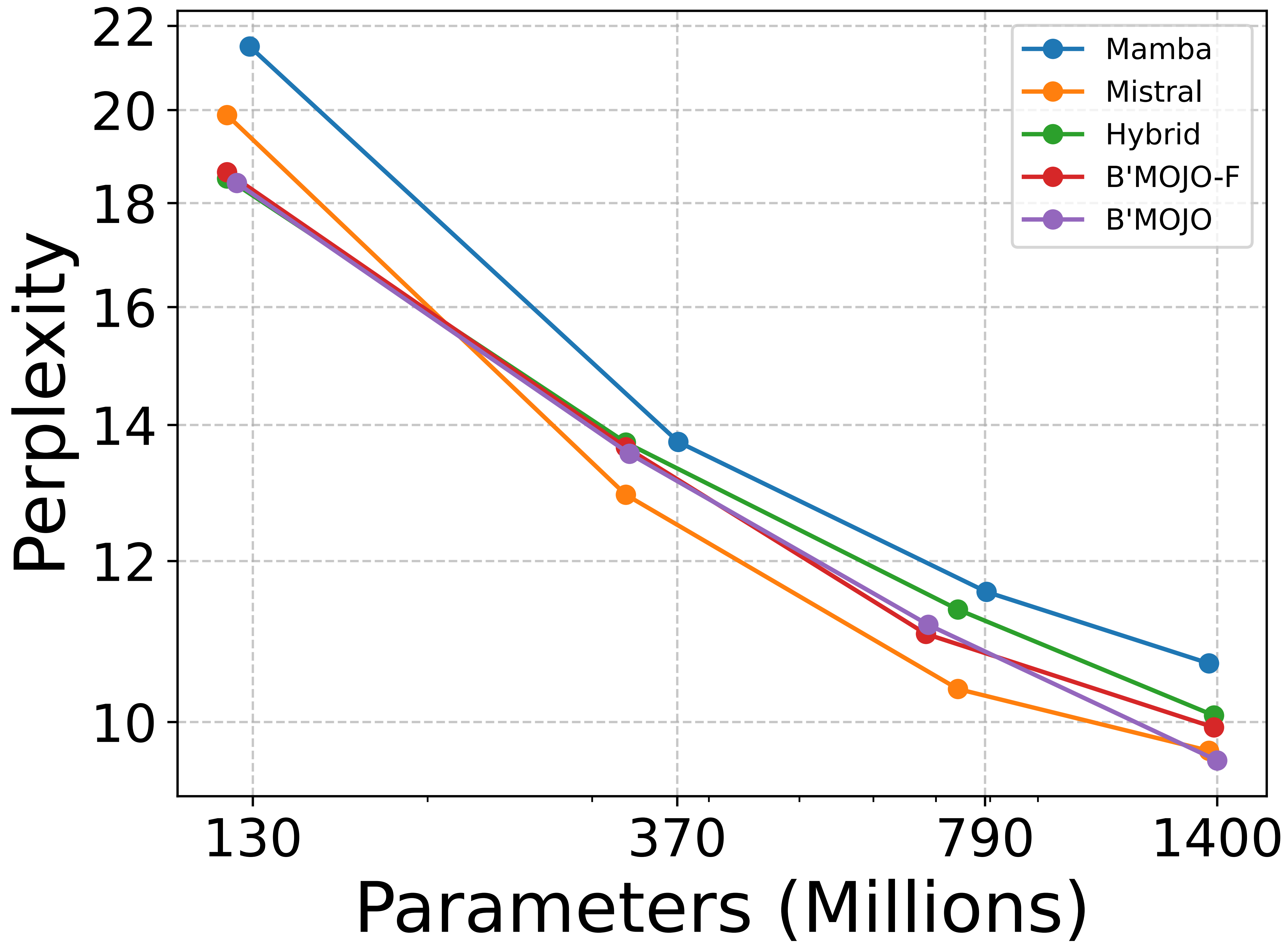
B’MOJO’s in-context associative recall: MQAR

MQAR Input: “A 1 B 3 C 2 E 5 ” Query: “A ? E ? C?” Expected output: “1 5 2”



B’MOJO’s Eidetic memory stores key-values pairs for future recall!

B’MOJO’s scaling laws



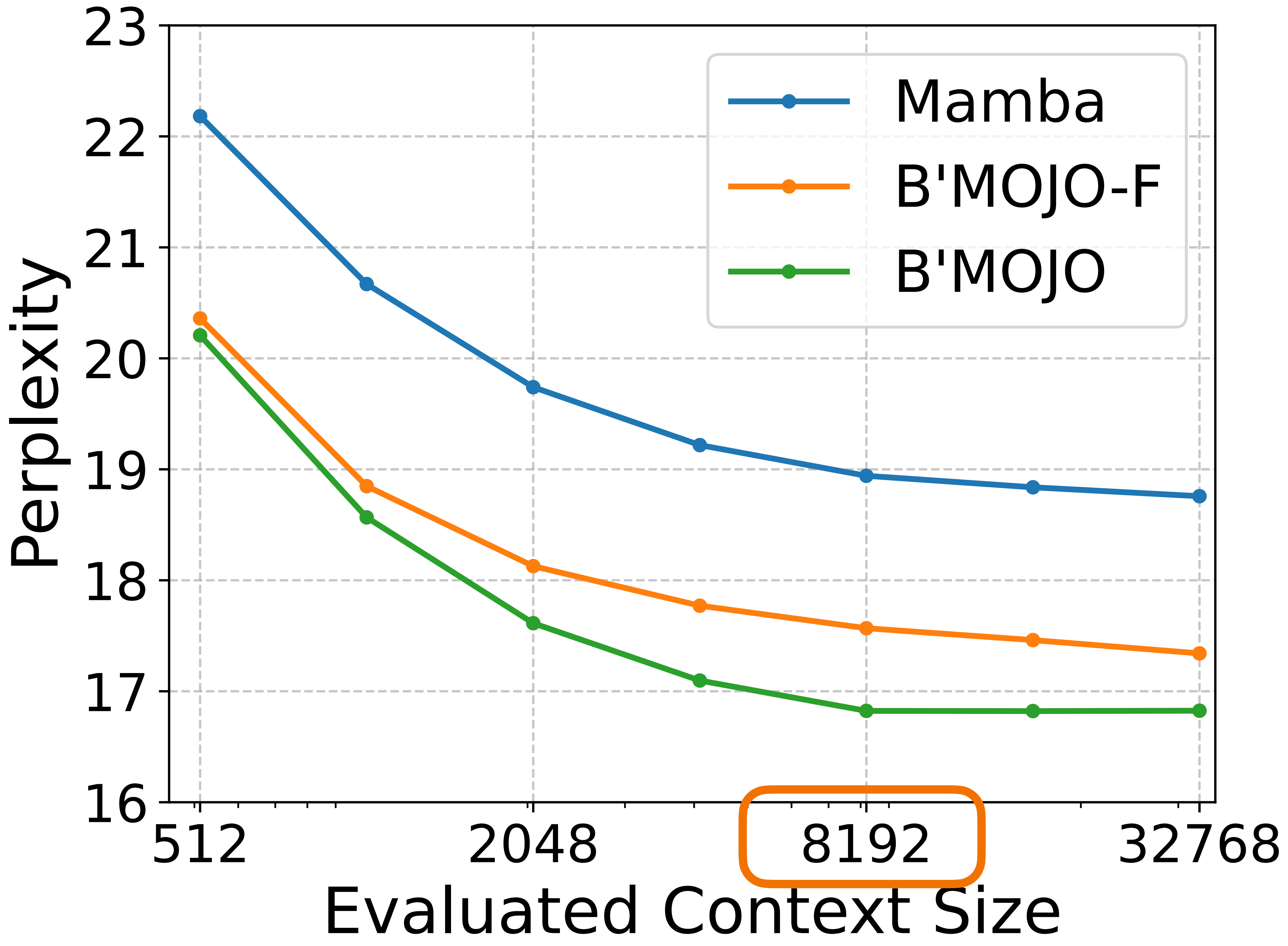
B’MOJO language modeling scaling law. B’MOJO exhibits a non-saturating scaling law.

Zero shot evaluation

		Pre-training Log-Perplexity	LAMBADA [34]	HellaSwag [51]	Short Context (acc ↑) PIQA [4]	ARC-E [8]	ARC-C [8]	WinoGrande [36]	Avg.
370M	Mistral (Full-Attention)	2.56	31.6	33.8	64.0	44.9	23.5	50.4	41.4
	Mamba (SSM)	2.62	31.4	33.4	63.5	45.0	22.3	51.7	41.2
	Hybrid (Sliding Attention + SSM)	2.69	26.3	31.3	61.1	42.7	22.4	51.9	39.3
	BMoJo (Fading)	2.68	29.6	33.2	63.7	43.1	23.0	51.8	40.7
	BMoJo (Fading + Eidetic)	2.67	28.6	33.3	63.9	44.3	22.1	50.7	40.5
1.4B	Mistral (Full-Attention)	2.27	50.1	50.7	70.4	58.2	27.5	54.4	51.9
	Mamba (SSM)	2.37	43.9	45.0	70.3	52.4	28.0	51.9	48.6
	Hybrid (Sliding Attention + SSM)	2.42	37.6	38.8	66.1	48.4	25.4	52.6	44.8
	BMoJo (Fading)	2.27	45.4	46.0	70.0	52.3	26.6	53.3	48.9
	BMoJo (Fading + Eidetic)	2.26	44.8	46.8	69.9	54.7	26.6	52.1	49.1

Zero-shot evaluation. B'MOJO outperforms our pre-trained Mamba and Hybrid models on common-sense reasoning and question-answering.

B'MOJO's length extrapolation



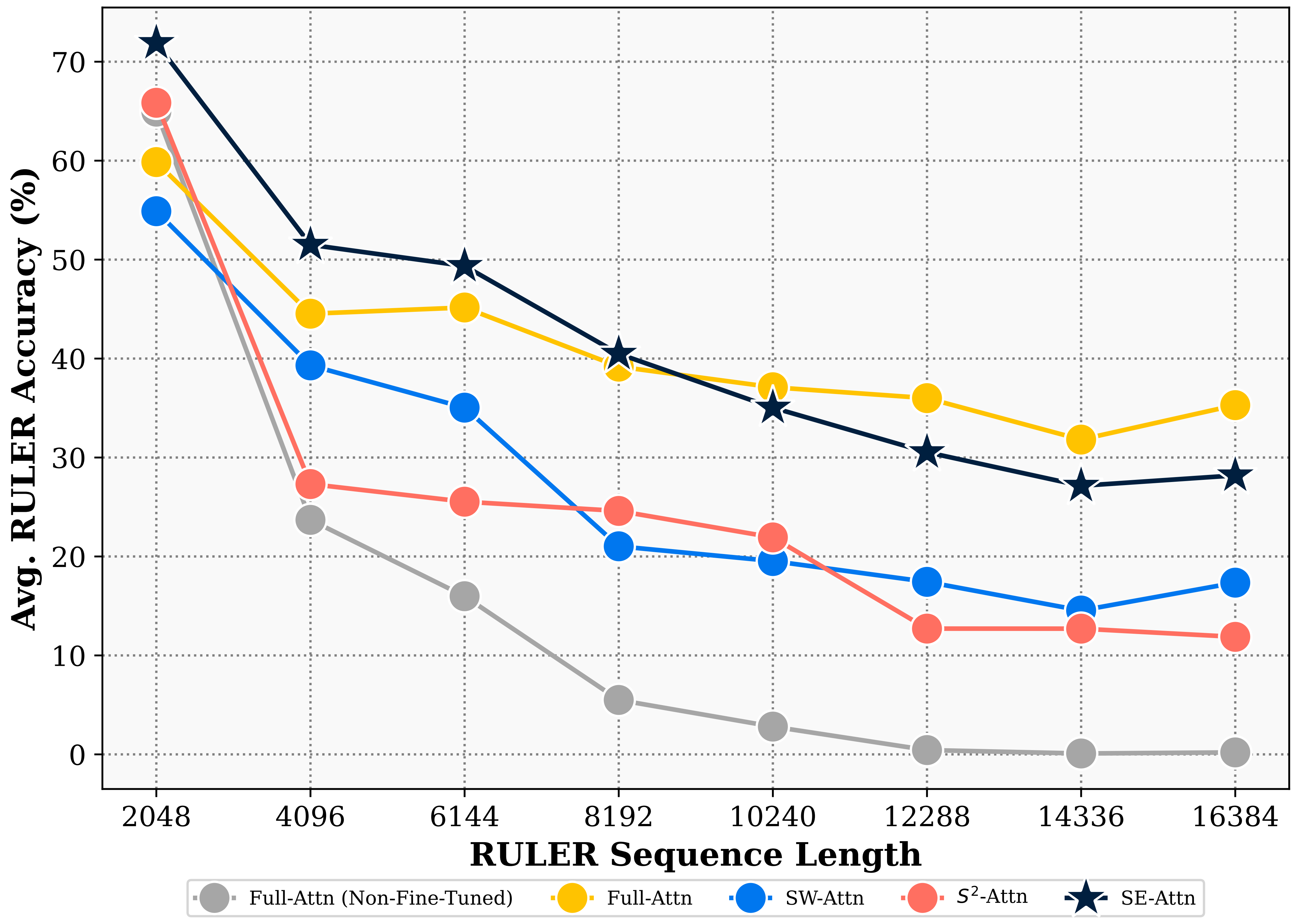
Length generalization. B'MOJO generalizes to longer sequences at inference time (up to 4x the ones seen during training). *B'MOJO does not need positional embeddings!*

Context Length	Model	S-NIAH	MK-NIAH	MV-NIAH	MQ-NIAH	Average
2048	Transformer	100	95	62	61	79
	Mamba	100	32	29	28	47
	B'MOJO-F	90	36	35	31	48
	B'MOJO	90	45	37	33	51
4096	Transformer	0	0	0	0	0
	Mamba	9	12	5	7	8
	B'MOJO-F	10	16	5	8	10
	B'MOJO	22	21	17	17	19

Length generalization on RULER. We trained our models on 2k context length and evaluated on RULER up to 4k sequences.

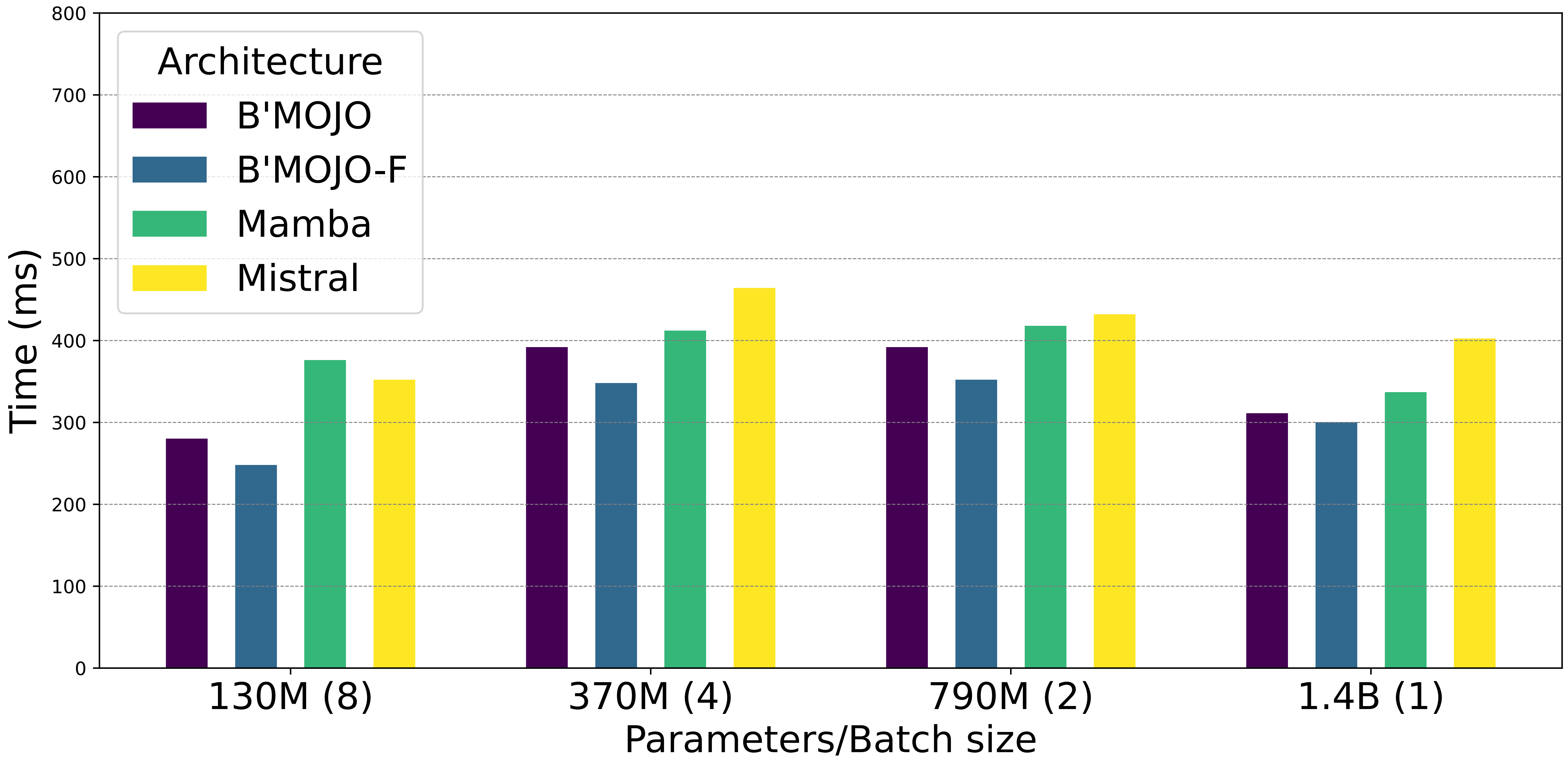
Span-Expanded Attention long context results

Attention	Eval Context Size (PG-19 PPL ↓) 2048	8192	16384	32768	Long Context Tasks (↑) SWDE	SQA	SNQA	Avg.
Non-fine-tuned	10.72	14.99	19.35	26.37	85.60	15.18	3.65	34.81
Full-Attn	10.99	10.28	10.39	11.14	85.24	26.99	19.75	43.99
SW-Attn	10.98	10.80	11.82	13.45	84.61	24.85	15.41	41.63
S^2-Attn	10.87	12.89	14.67	16.37	86.41	17.44	8.53	37.46
SE-Attn	10.99	10.45	11.14	12.64	85.96	26.70	18.04	43.57



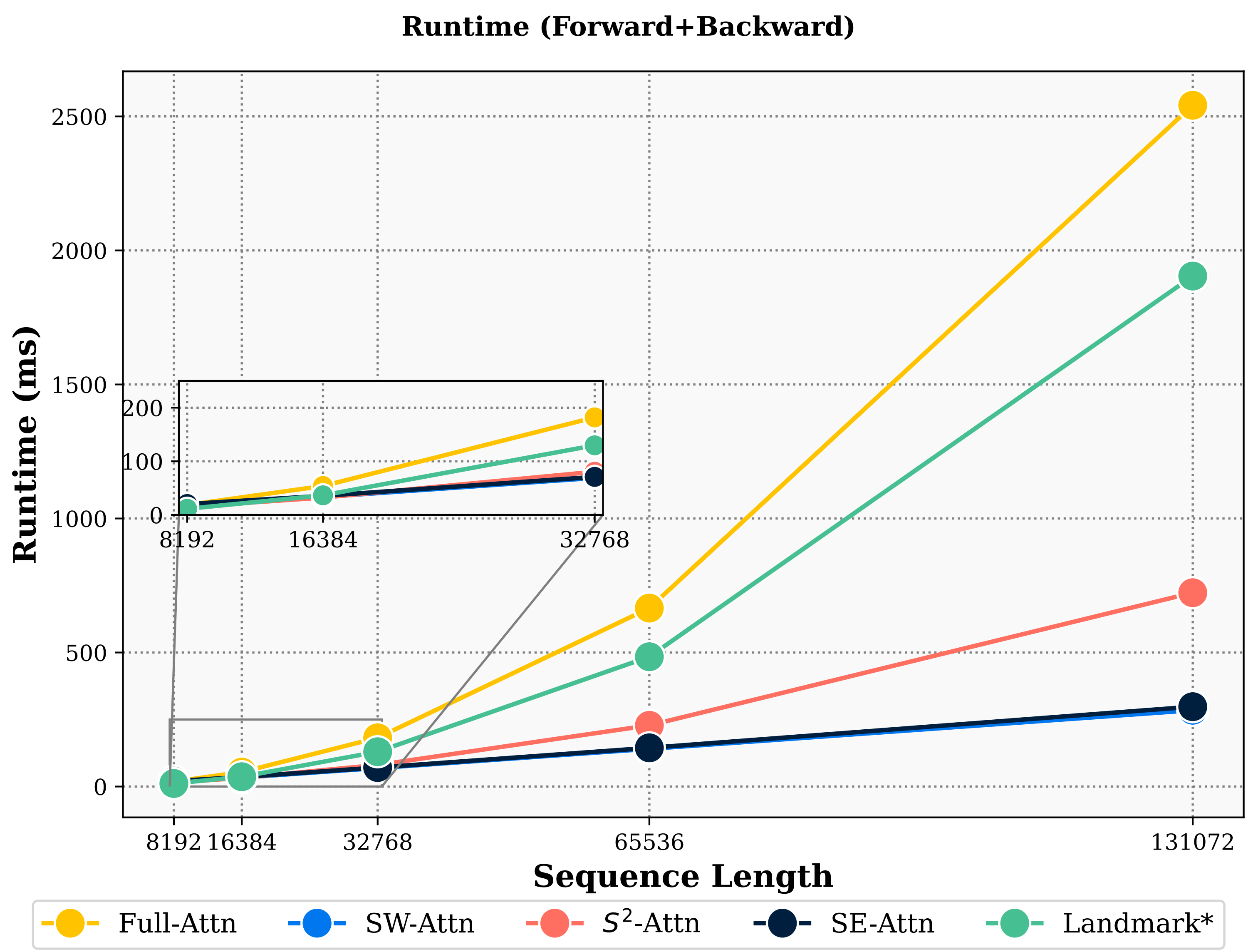
SE-Attention on long context. SE-Attention compares favorably with Full Attention on long context benchmarks (and length extrapolates on PG-19).

B'MOJO's Hardware Efficient Implementation

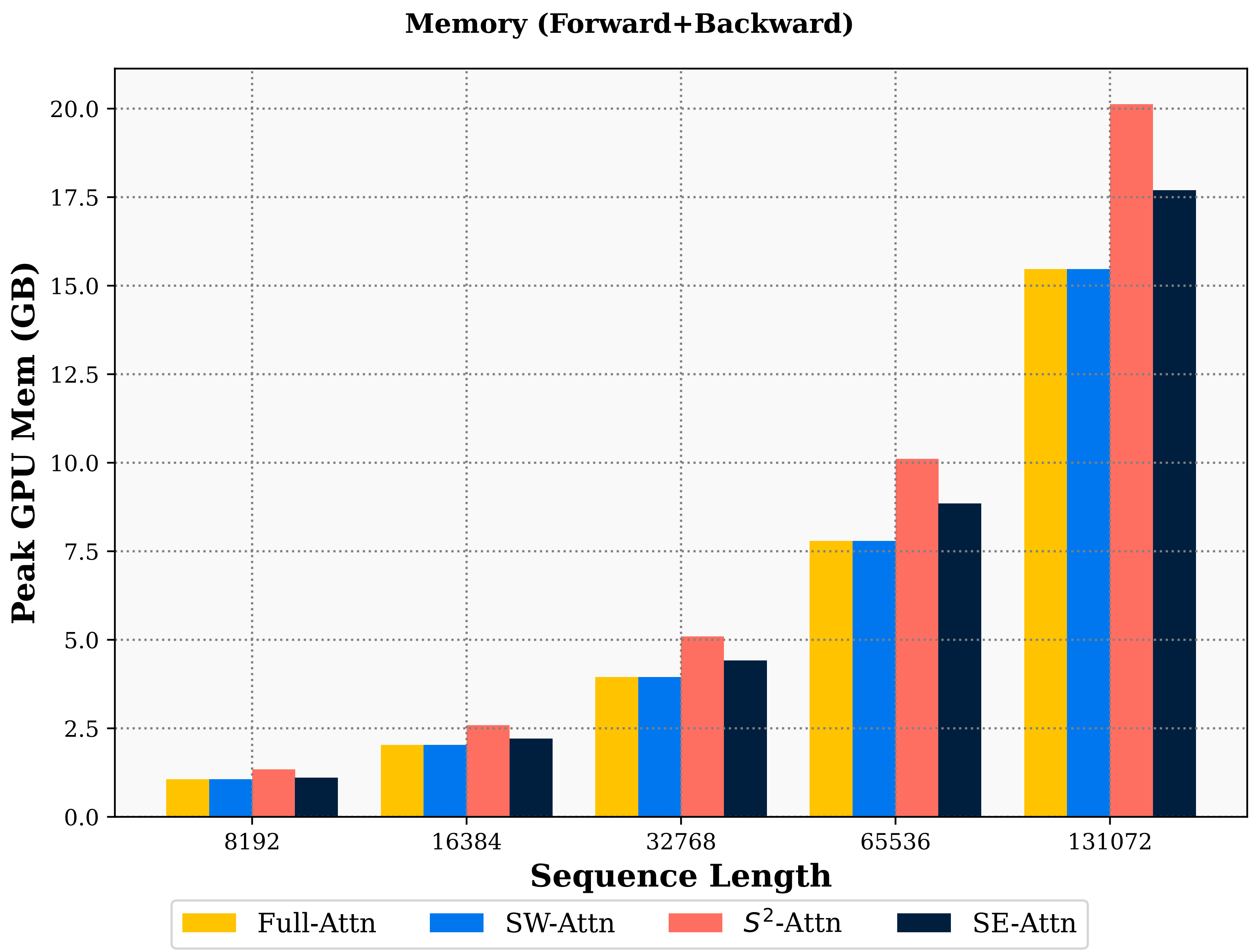


Time in ms to process sequences of 2k tokens. B'MOJO is faster than other efficient implementations of Mamba and Transformers at all scales.

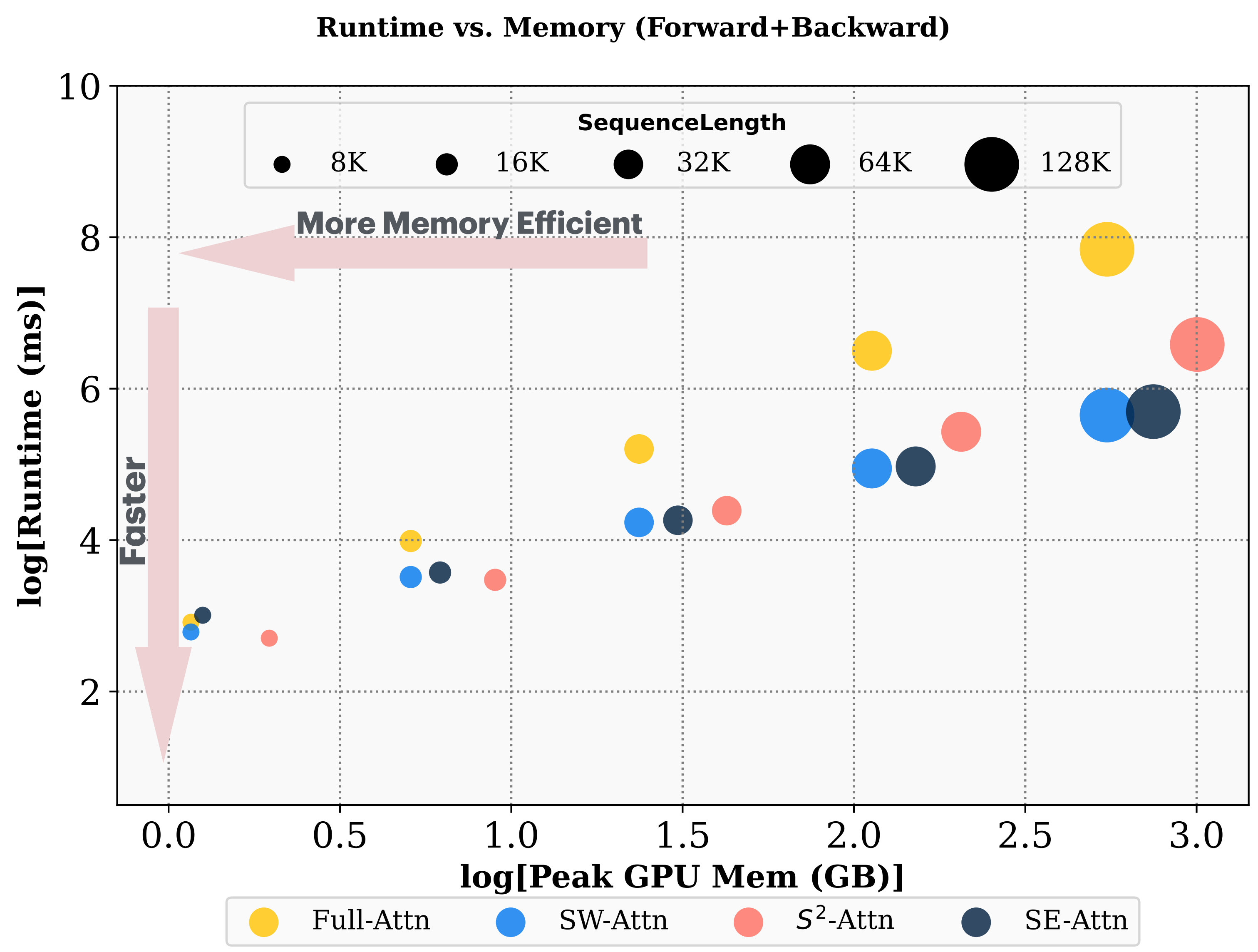
Span-Expanded Attention's efficiency



Runtime



Peak GPU Memory



Memory/Time Trade-off

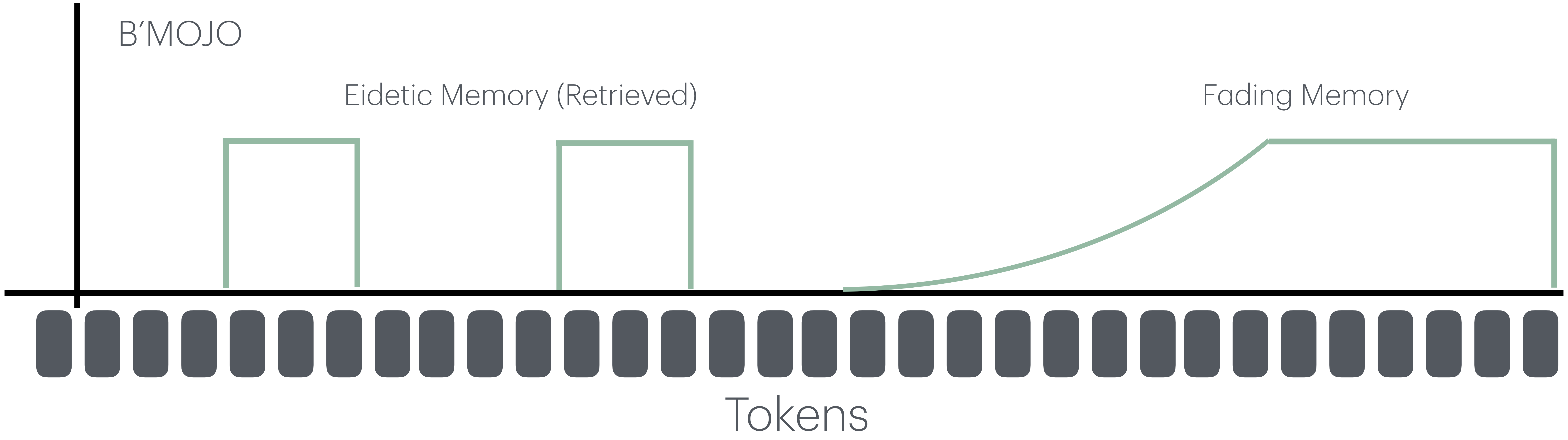
While increasing *peak memory usage* by15/20% compared to Full Attention/SWA **Span-Expanded Attention** *is up to 5x faster* than FlashAttention.

Summary

Realization of Memory hierarchies in Modern Sequence Models

Proposed B'MOJO an unified sequence layer that generalizes modern Hybrid models (SSMs + Attention)

B'MOJO has non-diagonal and input dependent recurrence (more expressive than Attention/Mamba)



Complemented efficient fading memory with:

- Causal Innovation Selection Mechanism
- Span-Expanded Attention, a Native-RAG method for Hybrid models

B'MOJO: Hybrid State Space Realizations of Foundation Models with Eidetic and Fading Memory

Luca Zancato* Arjun Seshadri Yonatan Dukler Aditya Golatkar Yantao Shen
Benjamin Bowman Matthew Trager Alessandro Achille Stefano Soatto
AWS AI Labs

Expansion Span: Combining Fading Memory and Retrieval in Hybrid State Space Models

Elvis Nunez[†*] Luca Zancato[‡] Benjamin Bowman[‡]
Aditya Golatkar[‡] Wei Xia[‡] Stefano Soatto[‡]
[†]UCLA [‡]AWS AI Labs
elvis.nunez@ucla.edu, {zancato, bowmaben, agolatka, wxia, soattos}@amazon.com

17 Dec 2024

Abstract

The 'state' of State Space Models (SSMs) represents their memory, which fades exponentially over an unbounded span. By contrast, Attention-based models have 'eidetic'

1. Introduction

State Space Models are able to process sequences with an unbounded number of tokens by maintaining a fixed-size state. However, this state is lossy and information about early tokens 'fades' as more inputs are processed. In con-

MARCONI: PREFIX CACHING FOR THE ERA OF HYBRID LLMS

Rui Pan [1*] Zhuang Wang [2] Zhen Jia [2] Can Karakus [2] Luca Zancato [2] Tri Dao [1] Yida Wang [2] Ravi Netravali [1]

If you are interested in Speculative Decoding efficient inference with Hybrid models

Thanks!

References

L. Zancato et al., "*B'MOJO: Hybrid State Space Realizations of Foundation Models with Eidetic and Fading Memory*", NeurIPS 2024

E. Nunez et al., "*Expansion Span: Combining Fading Memory and Retrieval in Hybrid State Space Models*", arXiv:2412.13328

R. Pan et al., "*Marconi: Prefix caching for the era of hybrid llms*", MLSys2025

Y. Wu et al., "*Snakes and Ladders: Accelerating SSM Inference with Speculative Decoding*", NeurIPS 2025 (Efficient NLP Workshop)

TY Liu et al., "*PICASO: Permutation-Invariant Context Composition with State Space Models*", ICLR 2025

S. Soatto et al., "*Taming ai bots: Controllability of neural states in large language models*", arXiv:2305.18449

L. Zancato et al., "*Train/test-time adaptation with retrieval*", CVPR 2023

TY Liu et al., "*Meaning representations from trajectories in autoregressive models*", ICLR 2024

R. Waleffe et al., "*An Empirical Study of Mamba-based Language Models*", arXiv:2406.07887

L. Csató and M. Opper, "*Sparse Online Gaussian Processes*", 2002

DeepSeek-AI Team, "*DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*", arXiv:2501.12948

J. Yuan et al., "*Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention*", arXiv:2502.11089v1