Mamba: Linear-Time Sequence Modeling with Selective State Spaces

Albert Gu*[1] and Tri Dao*[2]

[1] Machine Learning Department, Carnegie Mellon University

[2] Department of Computer Science, Princeton University

agu@cs.cmu.edu, tri@tridao.me

Outlines

- Introduction to **State Space Models** (SSM)
  - Motivations
  - **S**tructured **S**tate **S**pace **S**equence Models (S4)
  - Comparison to Transformers
- Mamba: **Selective** SSM
  - Selection Mechanism
  - Efficient Implementation
- Experimental Results
- Discussions
  - Current Extensions
  - Future Directions

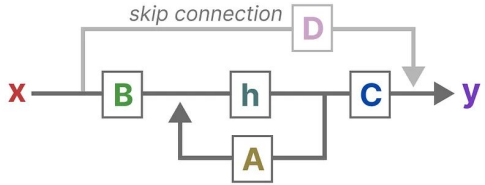
Introduction to State Space Models (SSM)

- Reference: *Modeling Sequences with Structured State Space*

- Originated from state space methods in control theory
  - Input: 1-dimensional function or sequence x(t)
  - Maintain latent state h(t) following

State equation $\mathbf{h}'(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{B}\mathbf{x}(t)$

Output equation $\mathbf{y}(t) = \mathbf{C}\mathbf{h}(t) + \mathbf{D}\mathbf{x}(t)$



where A, B, C & D are projection matrices. Such an SSM is represented by (A, B, C). D is often omitted as it is only a skip connection.

  - However, it is too general
    - Inefficient to compute, $O(N^2L)$ operations & $O(NL)$ space
    - Struggle to remember long dependencies, similar to the vanishing/exploding gradient problem of RNN

Structured State Space Models (S4)

- Discretization: for discrete inputs
  - e.g. zero-order hold (ZOH)

Discretized matrix **A**

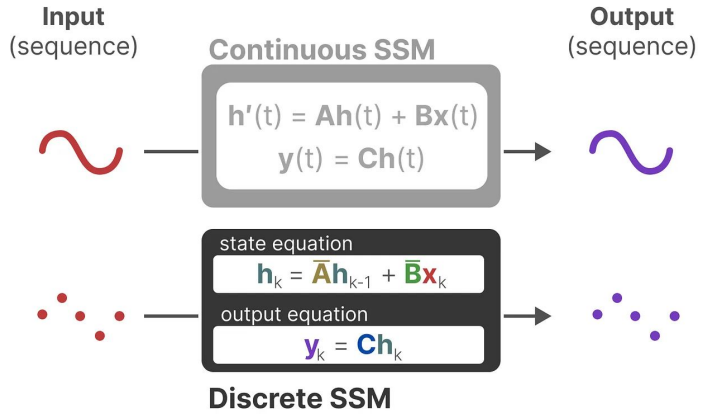
$$\overline{\mathbf{A}} = \exp(\Delta \mathbf{A})$$

Discretized matrix **B**

$$\overline{\mathbf{B}} = (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - I) \cdot \Delta \mathbf{B}$$

where Δ is the step size of sampling.

  - Now, the discrete SSM can be represented as (Δ, A, B, C), or ($\overline{A}$, $\overline{B}$, C)



Structured State Space Models (S4)

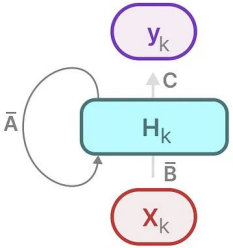
- Computation: Linear Time Invariant (LTI)
  - Linear recurrence

state equation

$$h_k = \bar{A}h_{k-1} + \bar{B}x_k$$

output equation

$$y_k = Ch_k$$



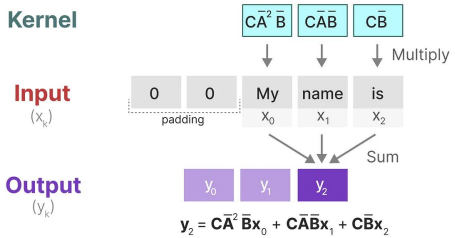
✓ *efficient inference*
✗ *parallelizable training*

  - Global convolution

$$\textit{kernel} \longrightarrow \bar{K} = (C\bar{B}, C\bar{A}\bar{B}, \ldots, C\bar{A}^{k}\bar{B}, \ldots)$$

$$y = x * \bar{K}$$

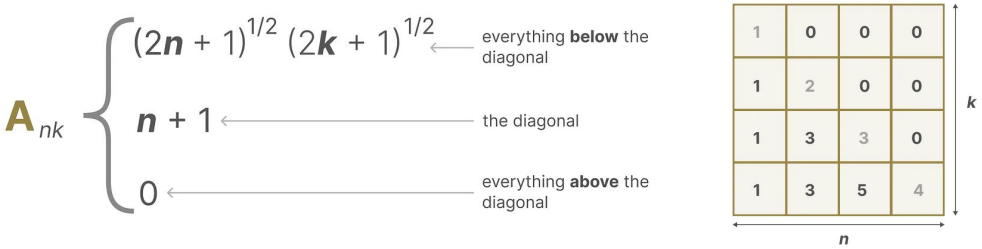
output input kernel



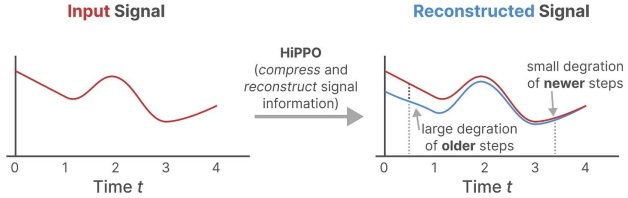
✗ *unbounded context*
✓ *parallelizable training*

Structured State Space Models (S4)

- By imposing *structure* on matrix A, the system can be solved using efficient algorithms
  - Popular choices: Diagonal, or High-order Polynomial Projection Operators (**HiPPO**)



  - HiPPO can reconstruct older signals, while keeping track of newer signals



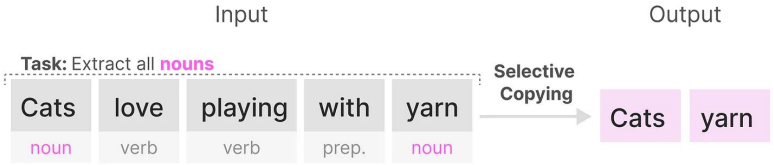
  - $O(N^2L)$ operations & $O(NL)$ space -> $\tilde{O}(N+L)$ operations and $O(N+L)$ space

Other Related Works

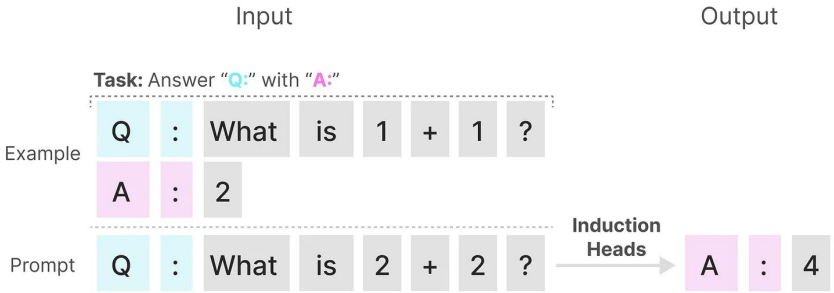
- Linear attention
- H3
- Hyena
- RetNet
- RWKV
- …

Mamba: Selective SSM

- Motivation: Selection as a Means of Compression
  - Selective Copying task



  - Induction Heads task



  - LTI fails as the constant ($\overline{A}$, $\overline{B}$, C) cannot let them select the correct information.

Mamba: Selective SSM

- Efficient Implementation of Selective SSMs

Algorithm 1 SSM (S4)

Input: $x : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
Output: $y : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
1: $\boldsymbol{A} : (\mathtt{D}, \mathtt{N}) \leftarrow$ Parameter
▷ Represents structured $N \times N$ matrix
2: $\boldsymbol{B} : (\mathtt{D}, \mathtt{N}) \leftarrow$ Parameter
3: $\boldsymbol{C} : (\mathtt{D}, \mathtt{N}) \leftarrow$ Parameter
4: $\Delta : (\mathtt{D}) \leftarrow \tau_\Delta(\text{Parameter})$
5: $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathtt{D}, \mathtt{N}) \leftarrow \text{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$
6: $y \leftarrow \text{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$
▷ Time-invariant: recurrence or convolution
7: **return** y

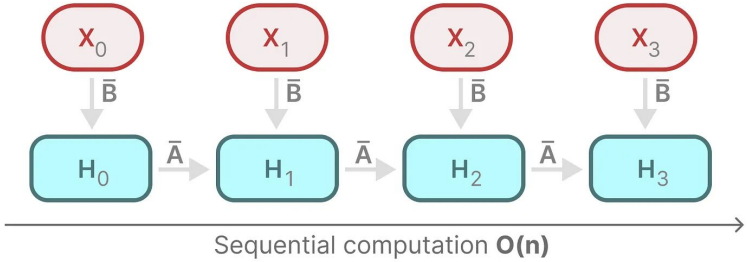
Algorithm 2 SSM + Selection (S6)

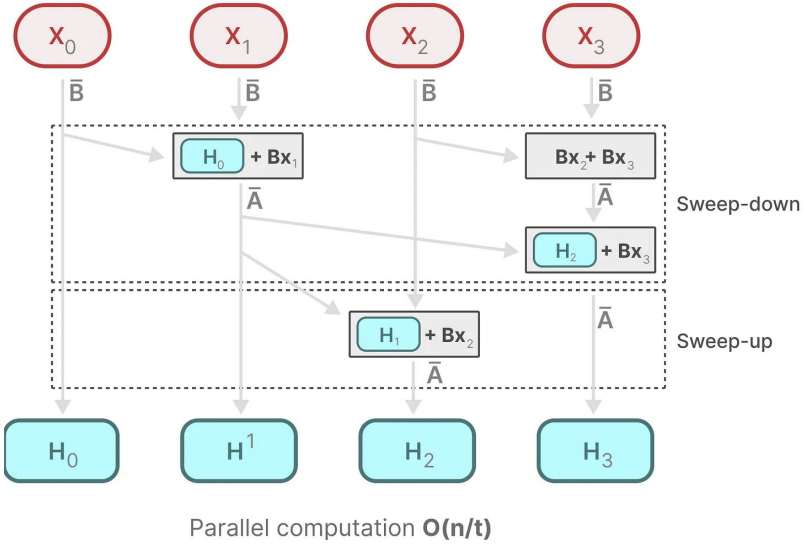
Input: $x : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
Output: $y : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
1: $\boldsymbol{A} : (\mathtt{D}, \mathtt{N}) \leftarrow$ Parameter
▷ Represents structured $N \times N$ matrix
2: $\boldsymbol{B} : (\mathtt{B}, \mathtt{L}, \mathtt{N}) \leftarrow s_B(x)$
3: $\boldsymbol{C} : (\mathtt{B}, \mathtt{L}, \mathtt{N}) \leftarrow s_C(x)$
4: $\Delta : (\mathtt{B}, \mathtt{L}, \mathtt{D}) \leftarrow \tau_\Delta(\text{Parameter} + s_\Delta(x))$
5: $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathtt{B}, \mathtt{L}, \mathtt{D}, \mathtt{N}) \leftarrow \text{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$
6: $y \leftarrow \text{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$
▷ Time-varying: recurrence (*scan*) only
7: **return** y

- Can no longer use convolution form

Mamba: Selective SSM

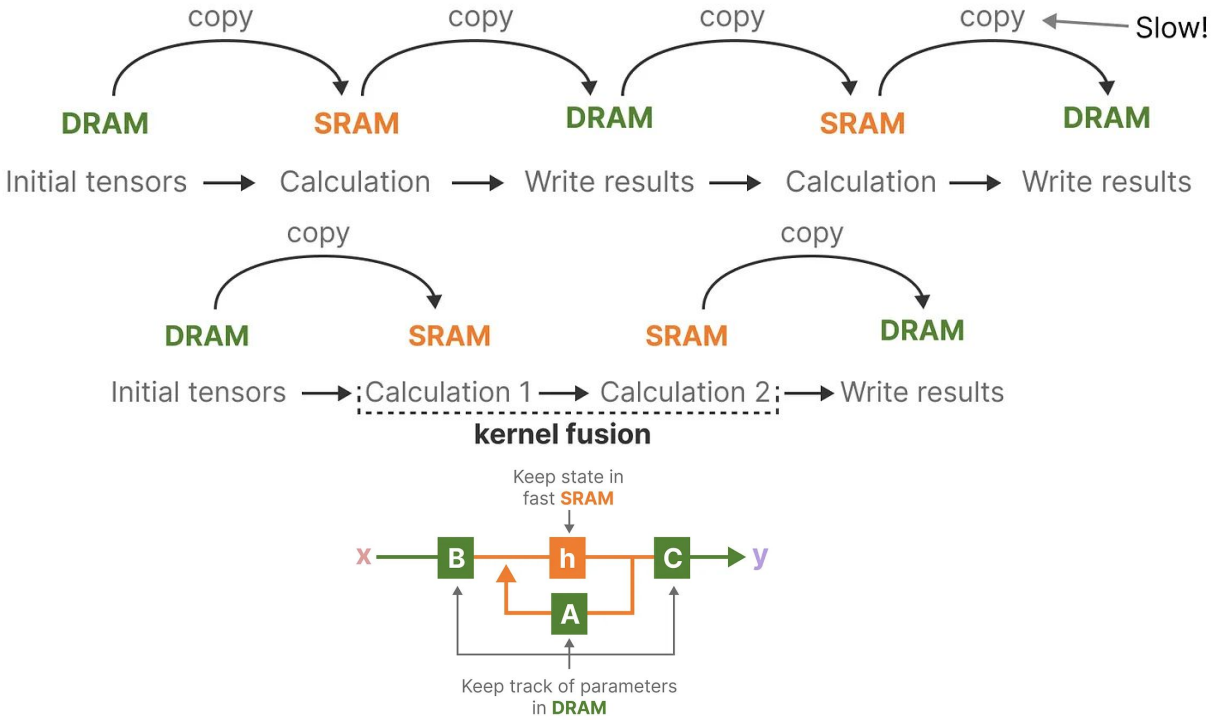
- Selective Scan: Dynamic Matrices + Parallel Scan





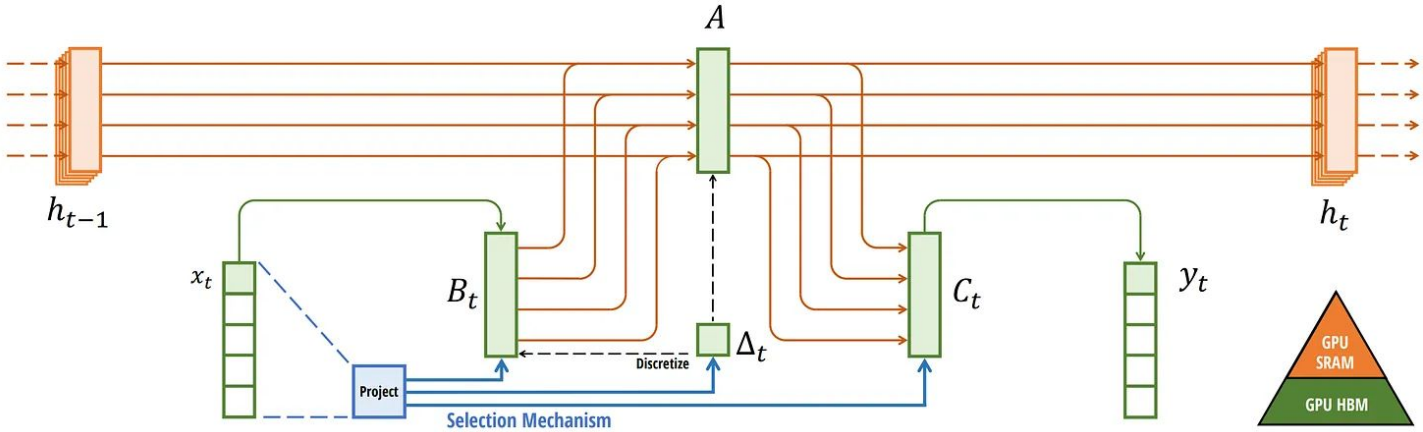
Mamba: Selective SSM

- Hardware Awareness



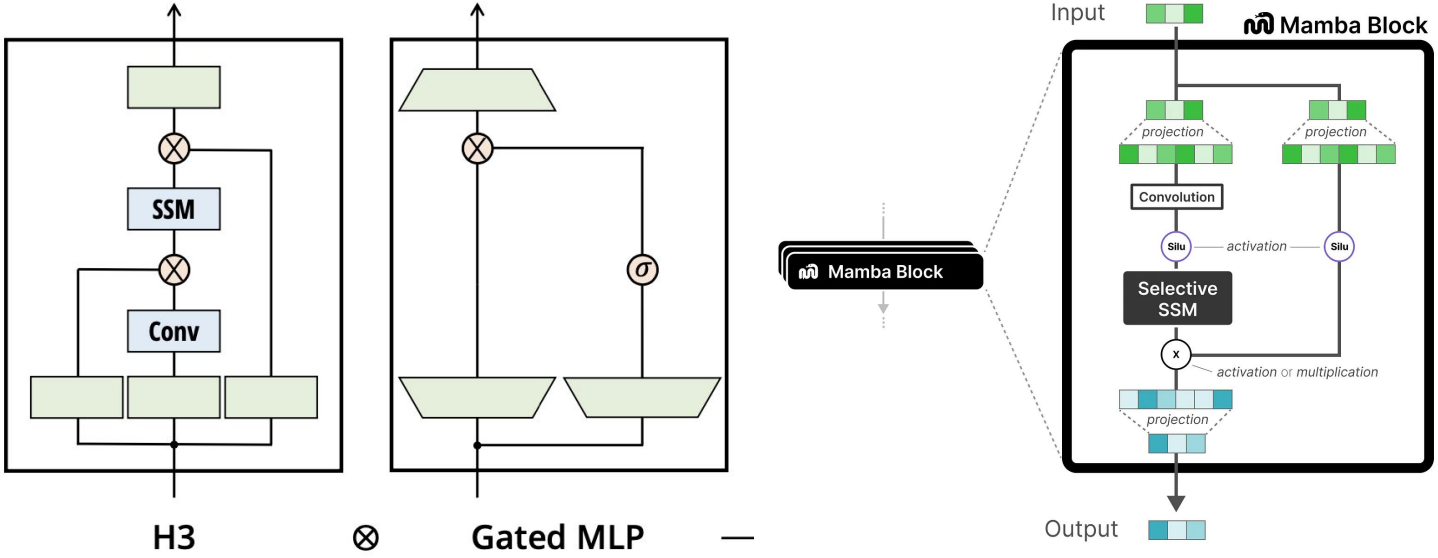
Mamba: Selective SSM

- Whole pipeline (S6)



Mamba: Selective SSM

- Mamba Block Design



Experimental Results

Model	Arch.	Layer	Acc.
S4	No gate	S4	18.3
-	No gate	S6	**97.0**
H3	H3	S4	57.0
Hyena	H3	Hyena	30.1
-	H3	S6	**99.7**
-	Mamba	S4	56.4
-	Mamba	Hyena	28.4
Mamba	Mamba	S6	**99.8**

Table 1: (**Selective Copying.**) Accuracy for combinations of architectures and inner sequence layers.

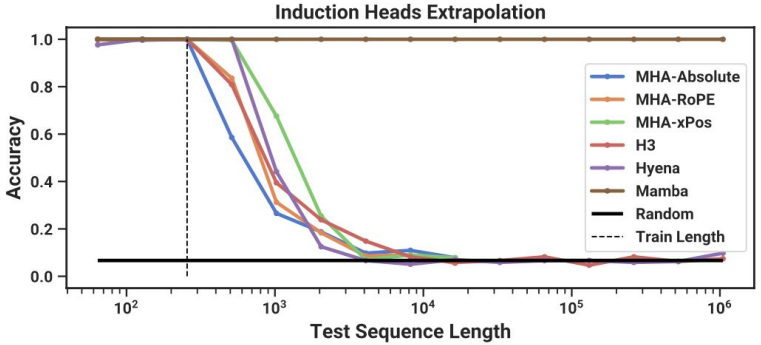


Table 2: (**Induction Heads.**) Models are trained on sequence length $2^8 = 256$, and tested on increasing sequence lengths of $2^6 = 64$ up to $2^{20} = 1048576$. Full numbers in Table 11.

Experimental Results

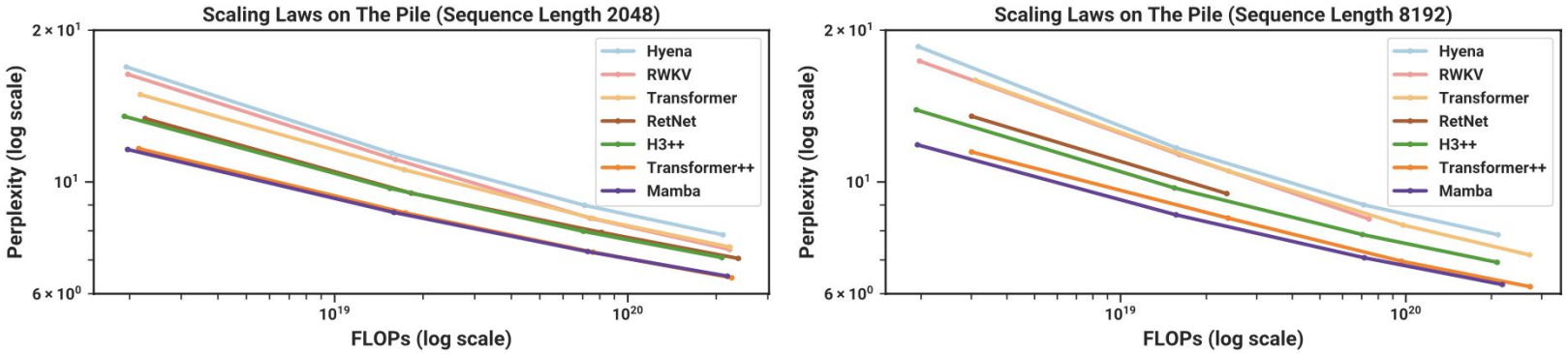


Figure 4: (**Scaling Laws.**) Models of size $\approx 125M$ to $\approx 1.3B$ parameters, trained on the Pile. Mamba scales better than all other attention-free models and is the first to match the performance of a very strong "Transformer++" recipe that has now become standard, particularly as the sequence length grows.

Experimental Results

Table 3: (**Zero-shot Evaluations**.) Best results for each size in bold. We compare against open source LMs with various tokenizers, trained for up to 300B tokens. Pile refers to the validation split, comparing only against models trained on the same dataset and tokenizer (GPT-NeoX-20B). For each model size, Mamba is best-in-class on every single evaluation result, and generally matches baselines at twice the model size.

Model	Token.	Pile ppl ↓	LAMBADA ppl ↓	LAMBADA acc ↑	HellaSwag acc ↑	PIQA acc ↑	Arc-E acc ↑	Arc-C acc ↑	WinoGrande acc ↑	Average acc ↑
Hybrid H3-130M	GPT2	—	89.48	25.77	31.7	64.2	44.4	24.2	50.6	40.1
Pythia-160M	NeoX	29.64	38.10	33.0	30.2	61.4	43.2	24.1	**51.9**	40.6
Mamba-130M	NeoX	**10.56**	**16.07**	**44.3**	**35.3**	**64.5**	**48.0**	**24.3**	**51.9**	**44.7**
Hybrid H3-360M	GPT2	—	12.58	48.0	41.5	68.1	51.4	24.7	54.1	48.0
Pythia-410M	NeoX	9.95	10.84	51.4	40.6	66.9	52.1	24.6	53.8	48.2
Mamba-370M	NeoX	**8.28**	**8.14**	**55.6**	**46.5**	**69.5**	**55.1**	**28.0**	**55.3**	**50.0**
Pythia-1B	NeoX	7.82	7.92	56.1	47.2	70.7	57.0	27.1	53.5	51.9
Mamba-790M	NeoX	**7.33**	**6.02**	**62.7**	**55.1**	**72.1**	**61.2**	**29.5**	**56.1**	**57.1**
GPT-Neo 1.3B	GPT2	—	7.50	57.2	48.9	71.1	56.2	25.9	54.9	52.4
Hybrid H3-1.3B	GPT2	—	11.25	49.6	52.6	71.3	59.2	28.1	56.9	53.0
OPT-1.3B	OPT	—	6.64	58.0	53.7	72.4	56.7	29.6	59.5	55.0
Pythia-1.4B	NeoX	7.51	6.08	61.7	52.1	71.0	60.5	28.5	57.2	55.2
RWKV-1.5B	NeoX	7.70	7.04	56.4	52.5	72.4	60.5	29.4	54.6	54.3
Mamba-1.4B	NeoX	**6.80**	**5.04**	**64.9**	**59.1**	**74.2**	**65.5**	**32.8**	**61.5**	**59.7**
GPT-Neo 2.7B	GPT2	—	5.63	62.2	55.8	72.1	61.1	30.2	57.6	56.5
Hybrid H3-2.7B	GPT2	—	7.92	55.7	59.7	73.3	65.6	32.3	61.4	58.0
OPT-2.7B	OPT	—	5.12	63.6	60.6	74.8	60.8	31.3	61.0	58.7
Pythia-2.8B	NeoX	6.73	5.04	64.7	59.3	74.0	64.1	32.9	59.7	59.1
RWKV-3B	NeoX	7.00	5.24	63.9	59.6	73.7	67.8	33.1	59.6	59.6
Mamba-2.8B	NeoX	**6.22**	**4.23**	**69.2**	**66.1**	**75.2**	**69.7**	**36.3**	**63.5**	**63.3**
GPT-J-6B	GPT2	–	4.10	68.3	66.3	75.4	67.0	36.6	64.1	63.0
OPT-6.7B	OPT	–	4.25	67.7	67.2	76.3	65.6	34.9	65.5	62.9
Pythia-6.9B	NeoX	6.51	4.45	67.1	64.0	75.2	67.3	35.5	61.3	61.7
RWKV-7.4B	NeoX	6.31	4.38	67.2	65.5	76.1	67.8	37.5	61.0	62.5

Experimental Results

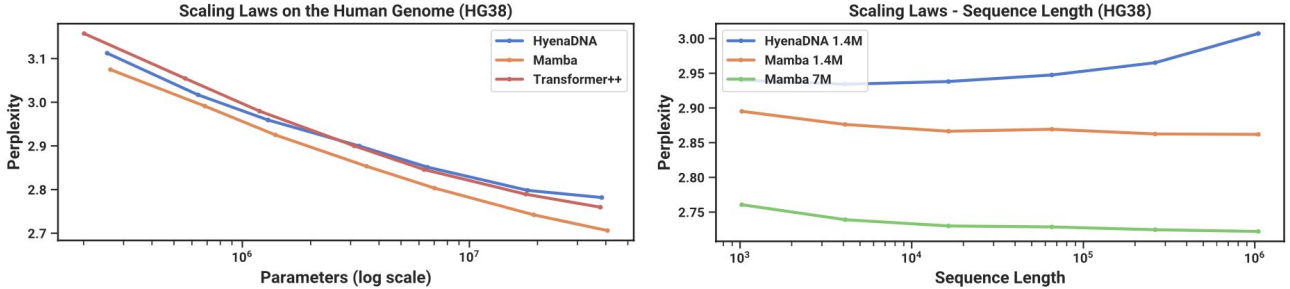


Figure 5: (**DNA Scaling Laws.**) Pretraining on the HG38 (human genome) dataset. (*Left*) Fixing short context length $2^{10} = 1024$ and increasing size from $\approx 200K$ to $\approx 40M$ parameters, Mamba scales better than baselines. (*Right*) Fixing model size and increasing sequence lengths while keeping tokens/batch and total training tokens fixed. Unlike baselines, the selection mechanism of Mamba facilitates better performance with increasing context length.

Experimental Results

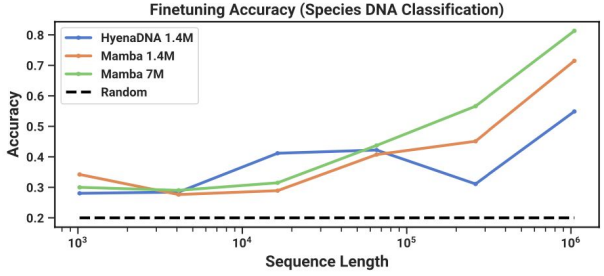


Figure 6: (**Great Apes DNA Classification**.) Accuracy after fine-tuning on sequences of length $2^{10} = 1024$ up to $2^{20} = 1048576$ using pretrained models of the same context length. Numerical results in Table 13.

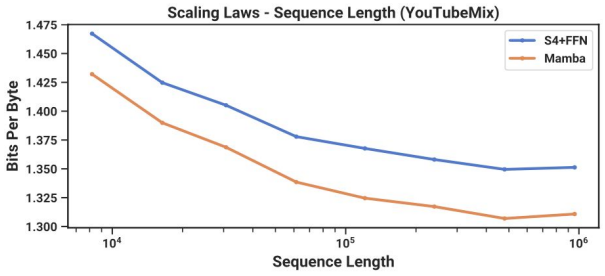


Figure 7: (**Audio Pretraining**.) Mamba improves performance over prior state-of-the-art (Sashimi) in autoregressive audio modeling, while improving up to minute-long context or million-length sequences (controlling for computation).

Experimental Results

Table 4: (**SC09**) Automated metrics for unconditional generation on a challenging dataset of fixed-length speech clips. (*Top to Bottom*) Autoregressive baselines, non-autoregressive baselines, Mamba, and dataset metrics.

Model	Params	NLL ↓	FID ↓	IS ↑	mIS ↑	AM ↓
SampleRNN	35.0M	2.042	8.96	1.71	3.02	1.76
WaveNet	4.2M	1.925	5.08	2.27	5.80	1.47
SaShiMi	5.8M	1.873	1.99	5.13	42.57	0.74
WaveGAN	19.1M	-	2.03	4.90	36.10	0.80
DiffWave	24.1M	-	1.92	5.26	51.21	0.68
+ SaShiMi	23.0M	-	1.42	5.94	69.17	0.59
Mamba	6.1M	**1.852**	0.94	6.26	88.54	0.52
Mamba	24.3M	1.860	**0.67**	**7.33**	**144.9**	**0.36**
Train	-	-	0.00	8.56	292.5	0.16
Test	-	-	0.02	8.33	257.6	0.19

Table 5: (**SC09 Model Ablations**) Models with 6M parameters. In SaShiMi's U-Net backbone, there are 8 center blocks operating on sequence length 1000, sandwiched on each side by 8 outer blocks on sequence length 4000, sandwiched by 8 outer blocks on sequence length 16000 (40 blocks total). The architecture of the 8 center blocks are ablated independently of the rest. Note that Transformers (MHA+MLP) were not tested in the more important outer blocks because of efficiency constraints.

Outer	Center	NLL ↓	FID ↓	IS ↑	mIS ↑	AM ↓
S4+MLP	MHA+MLP	1.859	1.45	5.06	47.03	0.70
S4+MLP	S4+MLP	1.867	1.43	5.42	53.54	0.65
S4+MLP	Mamba	1.859	1.42	5.71	56.51	0.64
Mamba	MHA+MLP	**1.850**	1.37	5.63	58.23	0.62
Mamba	S4+MLP	1.853	1.07	6.05	73.34	0.55
Mamba	Mamba	1.852	**0.94**	**6.26**	**88.54**	**0.52**

Experimental Results

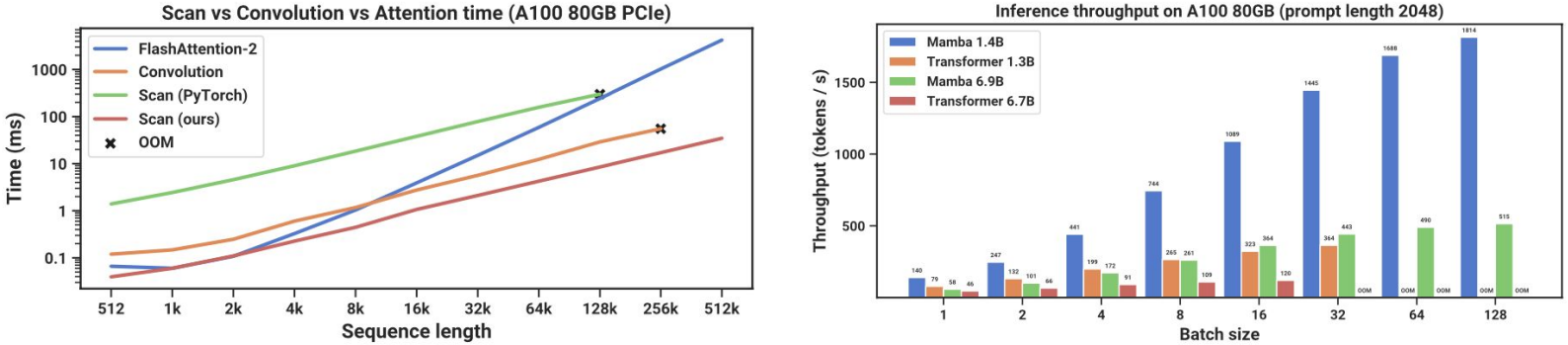


Figure 8: (**Efficiency Benchmarks**.) (*Left*) Training: our efficient scan is 40× faster than a standard implementation. (*Right*) Inference: as a recurrent model, Mamba can achieve 5× higher throughput than Transformers.

Experimental Results

Table 6: (**Ablations: Architecture and SSM layer**.) The Mamba block performs similarly to H3 while being simpler. In the inner layer, there is little difference among different parameterizations of LTI models, while selective SSMs (S6) provide a large improvement. More specifically, the S4 (real) variant is S4D-Real and the S4 (complex) variant is S4D-Lin.

Model	Arch.	SSM Layer	Perplexity
Hyena	H3	Hyena	10.24
H3	H3	S4 (complex)	10.30
-	H3	S4 (real)	10.34
-	H3	S6	**8.95**

Model	Arch.	SSM Layer	Perplexity
-	Mamba	Hyena	10.75
-	Mamba	S4 (complex)	10.54
-	Mamba	S4 (real)	10.56
Mamba	Mamba	S6	**8.69**

Experimental Results

Table 7: (**Ablations: Selective parameters**.) Δ is the most important parameter (Theorem 1), but using multiple selective parameters together synergizes.

Selective Δ	Selective B	Selective C	Perplexity
✗	✗	✗	10.93
✗	✓	✗	10.15
✗	✗	✓	9.98
✓	✗	✗	9.81
✓	✓	✓	8.71

Table 8: (**Ablations: Parameterization of A.**) The more standard initializations based on S4D-Lin (Gu, Gupta, et al. 2022) perform worse than S4D-Real or a random initialization, when the SSM is selective.

A_n Initialization	Field	Perplexity
$A_n = -\frac{1}{2} + ni$	Complex	9.16
$A_n = -1/2$	Real	8.85
$A_n = -(n+1)$	Real	8.71
$A_n \sim \exp(\mathcal{N}(0,1))$	Real	8.71

Experimental Results

Table 9: (**Ablations: Expressivity of Δ.**) The selection mechanism of Δ constructs it with a projection of the input. Projecting it even to dim. 1 provides a large increase in performance; increasing it further provides further improvements at the cost of a modest increase in parameters. State size fixed to $N = 16$.

Size of Δ proj.	Params (M)	Perplexity
-	358.9	9.12
1	359.1	8.97
2	359.3	8.97
4	359.7	8.91
8	360.5	8.83
16	362.1	8.84
32	365.2	8.80
64	371.5	8.71

Table 10: (**Ablations: SSM state dimension.**) (*Top*) Constant $\boldsymbol{B}$ and $\boldsymbol{C}$ (*Bottom*) Selective $\boldsymbol{B}$ and $\boldsymbol{C}$. Increasing the SSM state dimension N, which can be viewed as an expansion factor on the dimension of the recurrent state, can significantly improve performance for a negligible cost in parameters/FLOPs, but only when $\boldsymbol{B}$ and $\boldsymbol{C}$ are also selective. Size of Δ projection fixed to 64.

State dimension N	Params (M)	Perplexity
1	367.1	9.88
2	367.4	9.86
4	368.0	9.82
8	369.1	9.82
16	371.5	9.81
1	367.1	9.73
2	367.4	9.40
4	368.0	9.09
8	369.1	8.84
16	371.5	8.71

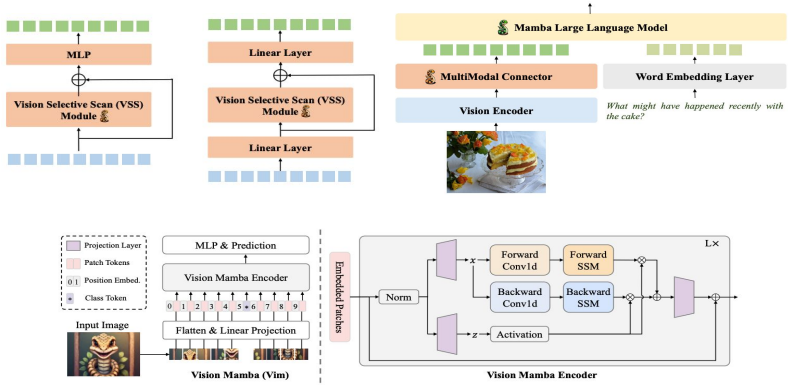
Discussions

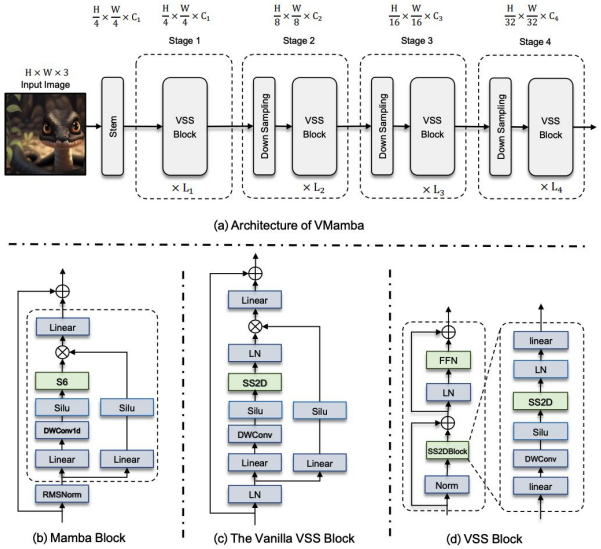
- Strength
  - The paper introduces a key mechanism by parameterizing the SSM parameters based on the input, allowing the model to filter out irrelevant information and remember relevant information indefinitely.
  - The results as compared to Pythia, and Transforms on many benchmarks are impressive.
  - The paper is written in a clear and understandable manner, with a well-defined approach and simple yet effective improvement strategies.

- Weakness
  - Might need a more concise experiment design to validate its capability as an alternative backbone for LLMs.

Discussions

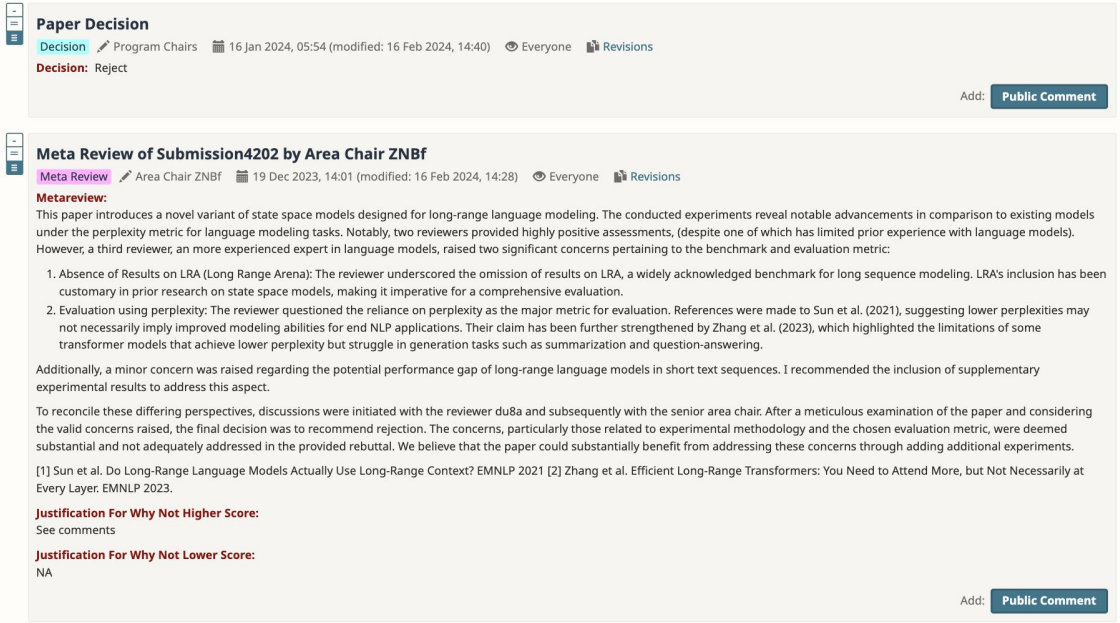
- Many works have explored this framework on distinctive modalities
  - Visual representation: Vision mamba, Vmamba, Localmamba, Hsimamba, …
  - Biomedical image segmentation: U-mamba, Segmamba, Vm-unet, …
  - Video representation: Videomamba
  - Motion generation: Motion mamba
  - Multimodality: VL-Mamba
  - Checkout **Awesome-Mamba-Papers** on Github for more!





Discussions

- However, it is rejected by ICLR 2024
- TL;DR
  - Absence of results on Long Range Arena (LRA)
  - Evaluation using perplexity
  - Lack of evaluation on short sequences



Paper Decision

Decision · Program Chairs · 16 Jan 2024, 05:54 (modified: 16 Feb 2024, 14:40) · Everyone · Revisions

Decision: Reject

Add: Public Comment

Meta Review of Submission4202 by Area Chair ZNBf

Meta Review · Area Chair ZNBf · 19 Dec 2023, 14:01 (modified: 16 Feb 2024, 14:28) · Everyone · Revisions

Metareview:

This paper introduces a novel variant of state space models designed for long-range language modeling. The conducted experiments reveal notable advancements in comparison to existing models under the perplexity metric for language modeling tasks. Notably, two reviewers provided highly positive assessments, (despite one of which has limited prior experience with language models). However, a third reviewer, an more experienced expert in language models, raised two significant concerns pertaining to the benchmark and evaluation metric:

1. Absence of Results on LRA (Long Range Arena): The reviewer underscored the omission of results on LRA, a widely acknowledged benchmark for long sequence modeling. LRA's inclusion has been customary in prior research on state space models, making it imperative for a comprehensive evaluation.
2. Evaluation using perplexity: The reviewer questioned the reliance on perplexity as the major metric for evaluation. References were made to Sun et al. (2021), suggesting lower perplexities may not necessarily imply improved modeling abilities for end NLP applications. Their claim has been further strengthened by Zhang et al. (2023), which highlighted the limitations of some transformer models that achieve lower perplexity but struggle in generation tasks such as summarization and question-answering.

Additionally, a minor concern was raised regarding the potential performance gap of long-range language models in short text sequences. I recommended the inclusion of supplementary experimental results to address this aspect.

To reconcile these differing perspectives, discussions were initiated with the reviewer du8a and subsequently with the senior area chair. After a meticulous examination of the paper and considering the valid concerns raised, the final decision was to recommend rejection. The concerns, particularly those related to experimental methodology and the chosen evaluation metric, were deemed substantial and not adequately addressed in the provided rebuttal. We believe that the paper could substantially benefit from addressing these concerns through adding additional experiments.

[1] Sun et al. Do Long-Range Language Models Actually Use Long-Range Context? EMNLP 2021 [2] Zhang et al. Efficient Long-Range Transformers: You Need to Attend More, but Not Necessarily at Every Layer. EMNLP 2023.

Justification For Why Not Higher Score:

See comments

Justification For Why Not Lower Score:

NA

Add: Public Comment

Thank you!
Any Questions?

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN