THE UNIVERSITY
OF BRITISH COLUMBIA

Mamba

Fatemeh Khashei, Yongxing Zhang

January 2026

Mamba: Linear Time-Sequence Modeling with Selective State Spaces

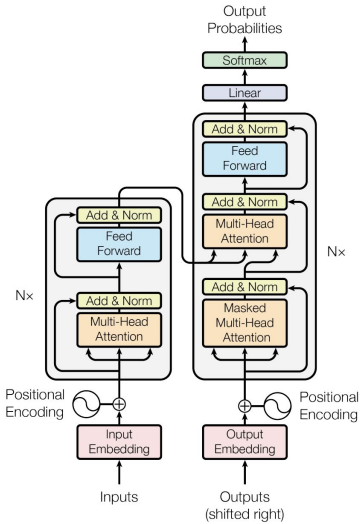
***Albert Gu**, Machine Learning Department, Carnegie Mellon University*

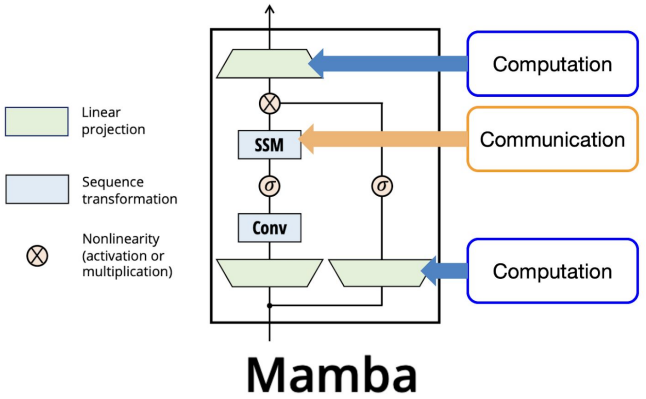
***Tri Dao**, Department of Computer Science, Princeton University*

COLM 2024

Transformer

- Transformer architecture [*] has been used in nearly all the LLMs that are being used today
- New architectures are being developed: Mamba, a State Space Model





[*] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. NIPS 2017.

Transformer

	Training	Inference
Transformers	**Fast!** (parallelizable)	**Slow...** (scales **quadratically** with sequence length)

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

RNNs

	Training	Inference
Transformers	**Fast!** (parallelizable)	**Slow...** (scales **quadratically** with sequence length)
RNNs	**Slow...** (not parallelizable)	**Fast!** (scales **linearly** with sequence length)

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Best of Two Worlds?

	Training	Inference
Transformers	**Fast!** (parallelizable)	**Slow...** (scales **quadratically** with sequence length)
RNNs	**Slow...** (not parallelizable)	**Fast!** (scales **linearly** with sequence length)
?	**Fast!** (parallelizable)	**Fast!** (scales **linearly** with sequence length)

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

State Space Model (SSM)

- SSMs [*] Traditionally used in control theory to model a dynamic system via state variables

- Models used to describe these state representations and make predictions of what their next state could be depending on some input

[*] Albert Gu, Karan Goel, Christopher Ré. 2021. Efficiently Modeling Long Sequences with Structured State Spaces. *ICLR*.

State Space Model (SSM)

At time t

- x(t) input sequence - Continuous
- h(t) current state
- y(t) output sequence - Continuous

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

State Space Model (SSM)

- SSMs assume that dynamic systems, such as an object moving in 3D space, can be predicted from its state at time t through two equations.

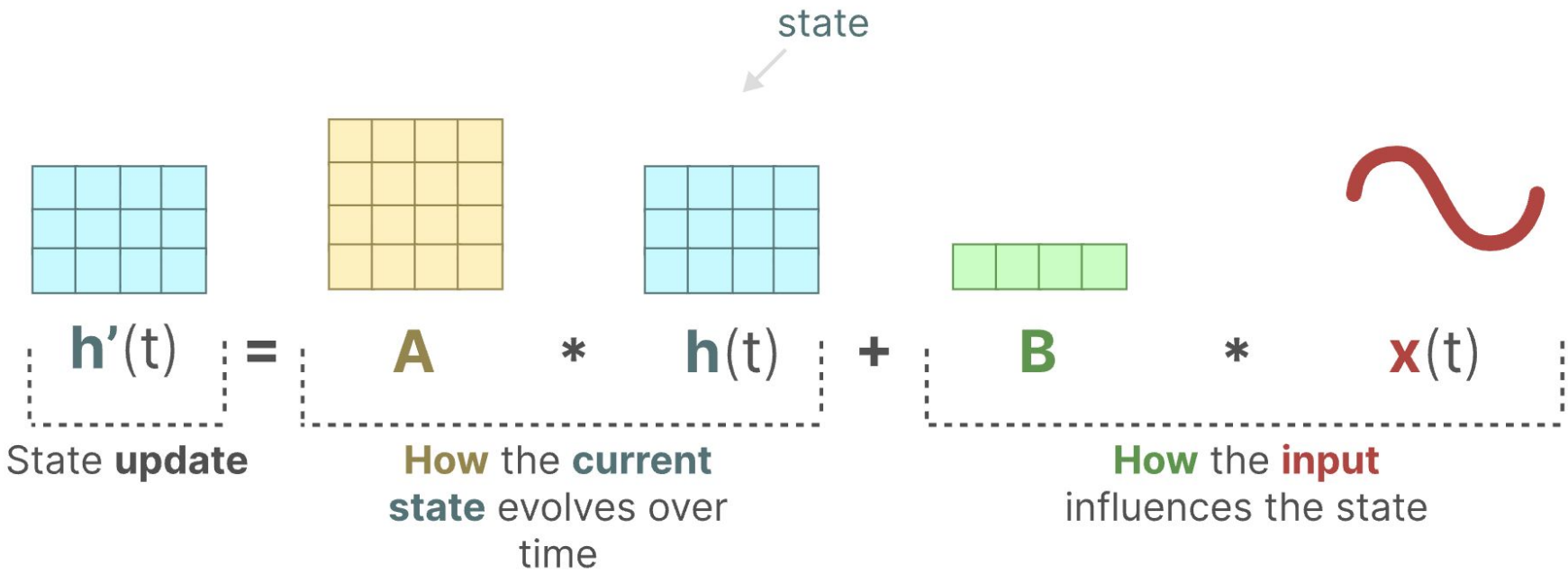
State equation $$\mathbf{h'}(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{B}\mathbf{x}(t)$$

  - h’(t): the way the state is changing at time t

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

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

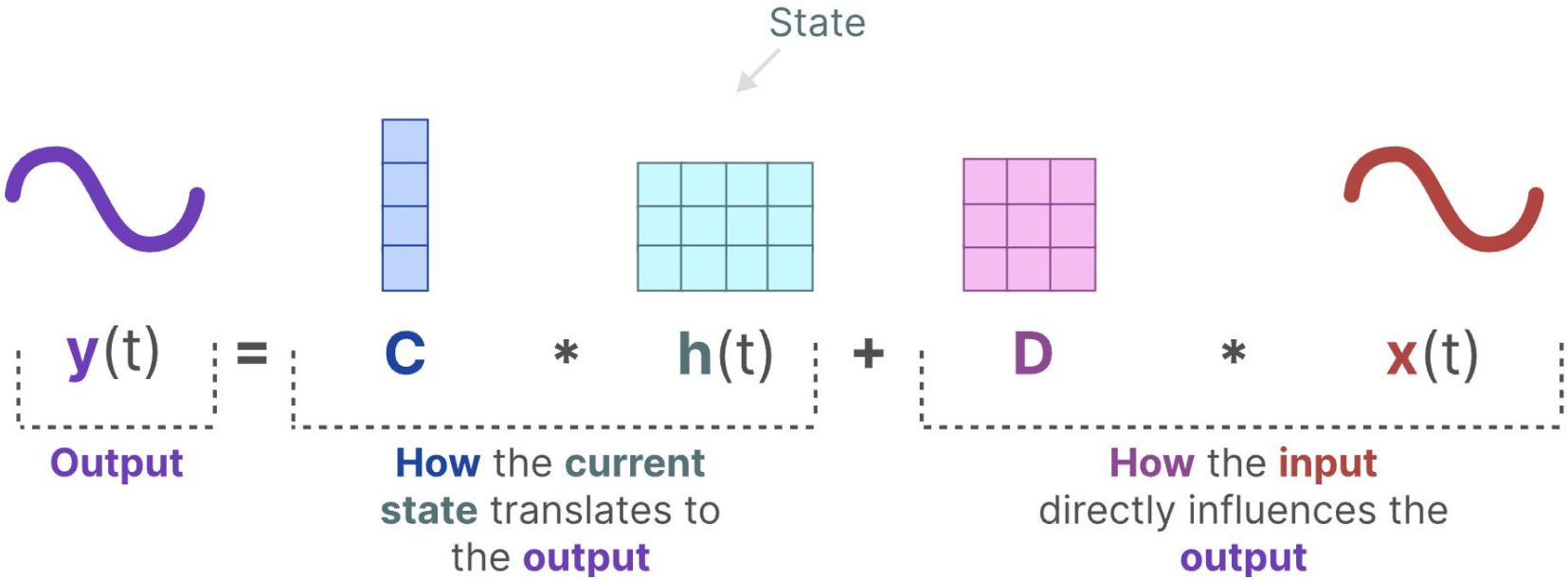
The State Equation



A Matrix how the current state **h**, influences the future state
B Matrix how a given input affects each state variable

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

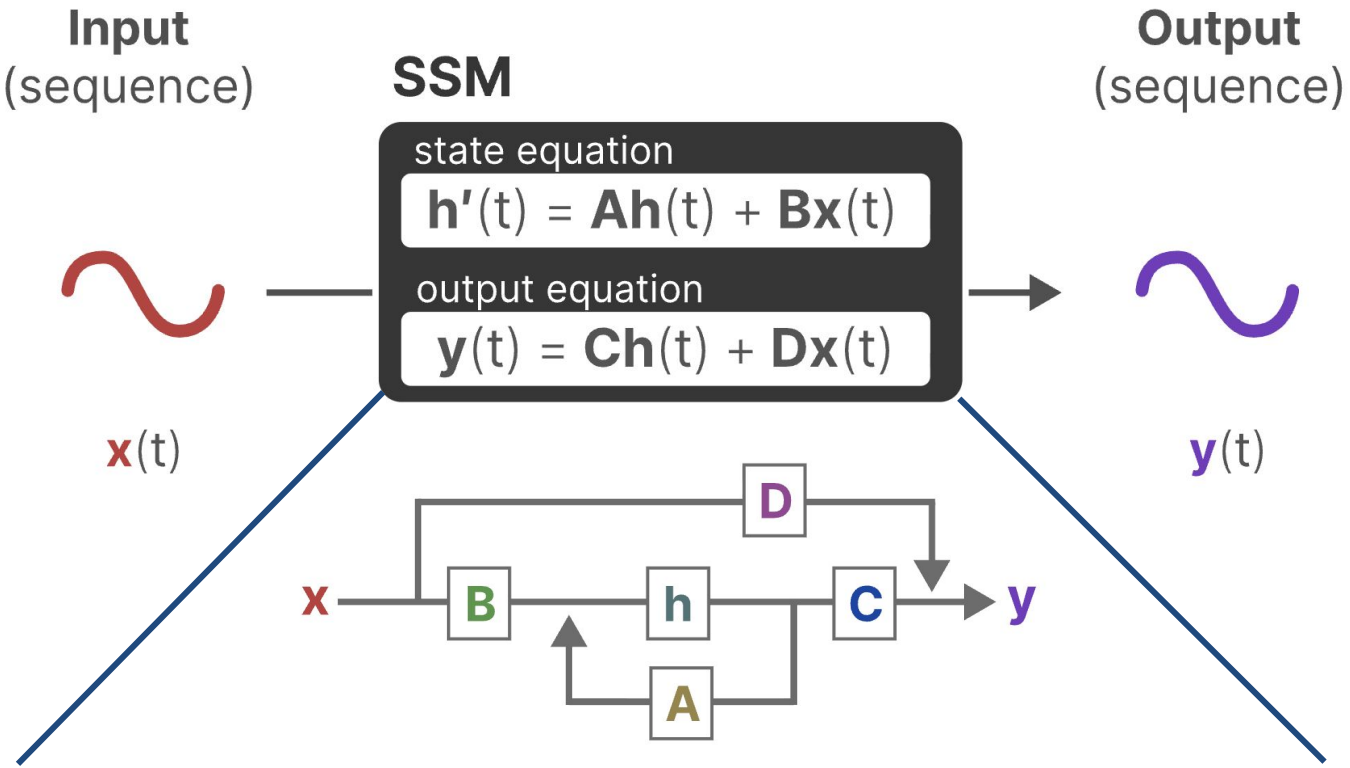
The Output Equation



C Matrix the relationship between the internal state variables and the output y
D Matrix how the input directly influences the observed system output (often omitted)

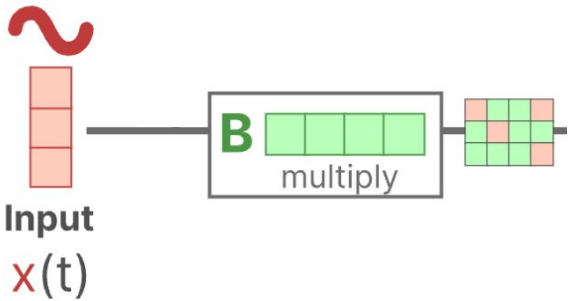
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

SSM Overview



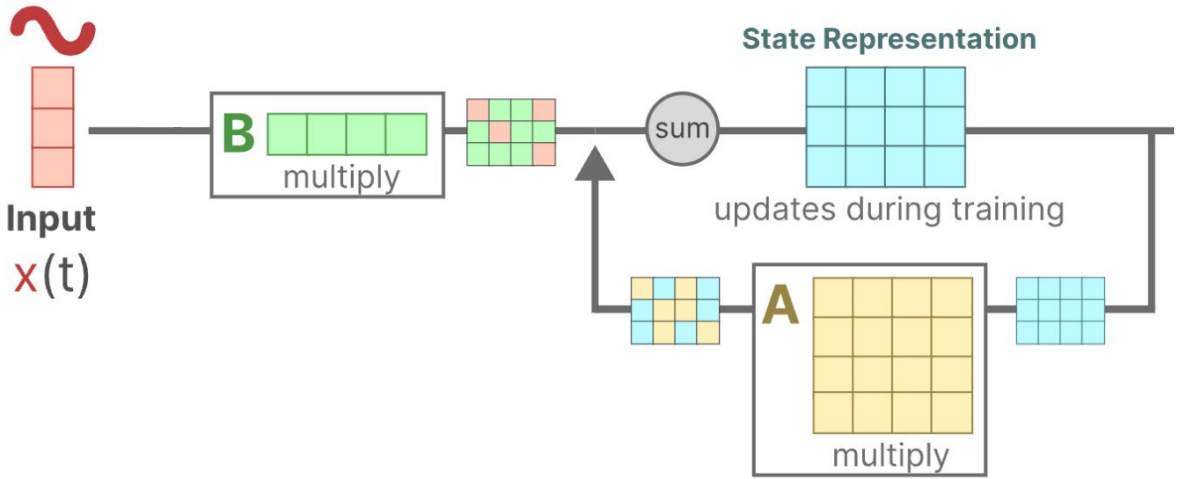
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

SSM Step-By-Step



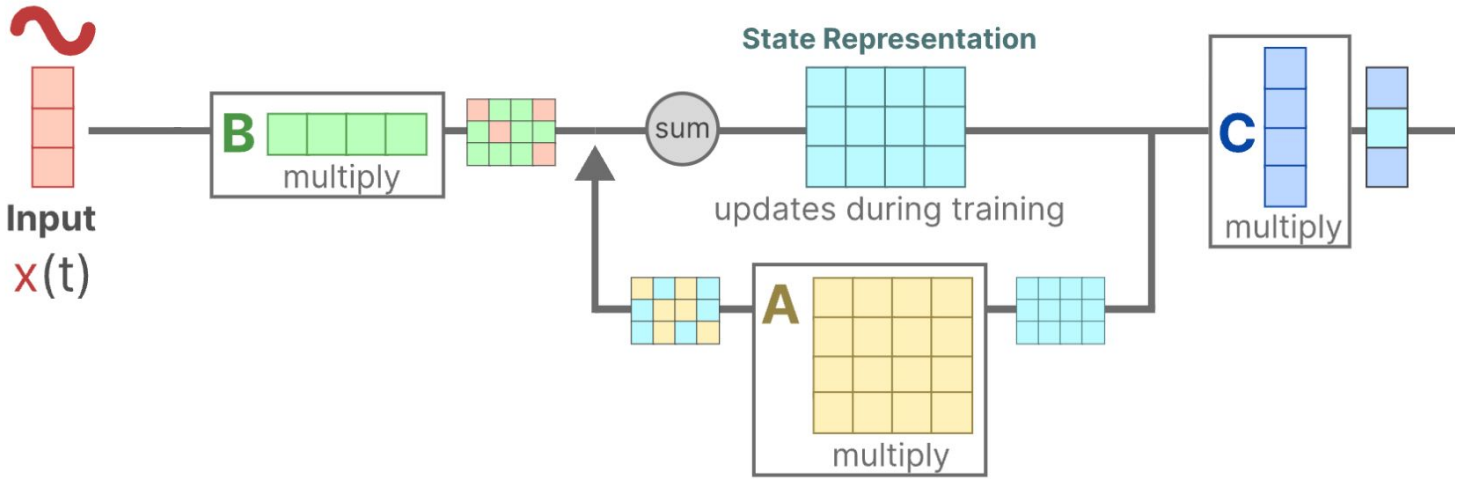
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

SSM Step-By-Step



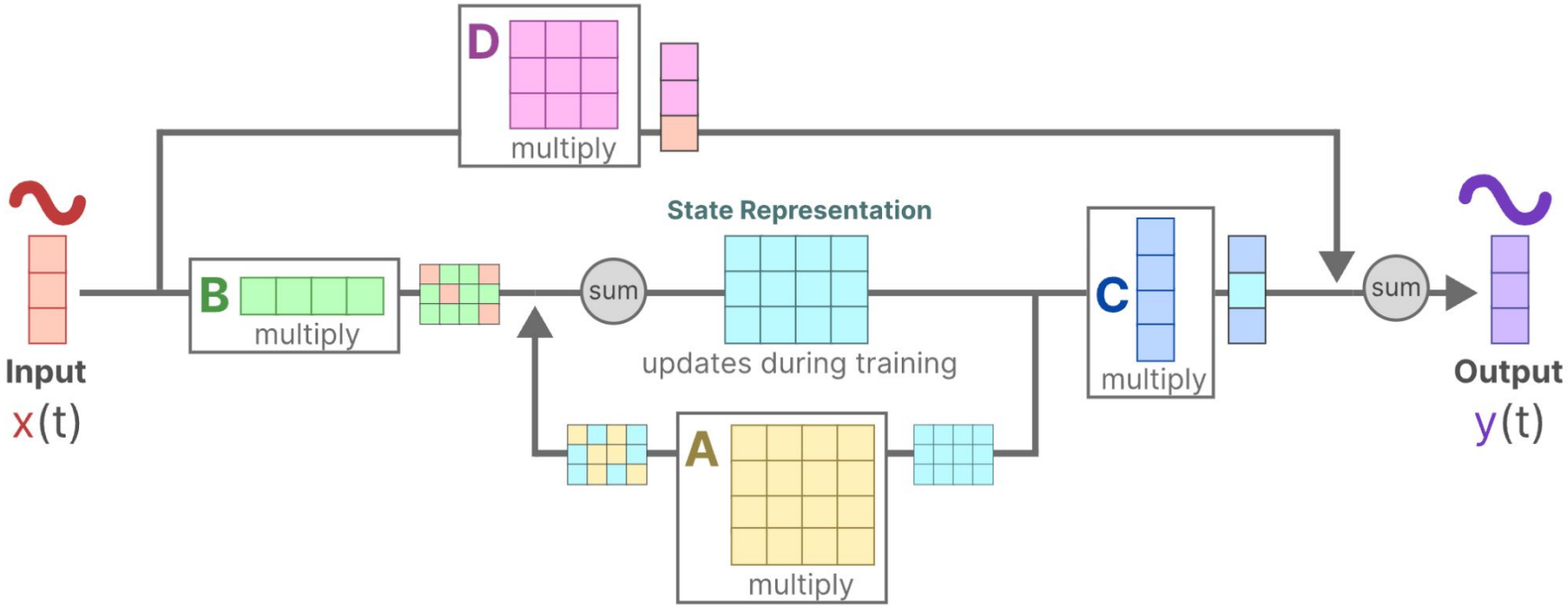
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

SSM Step-By-Step



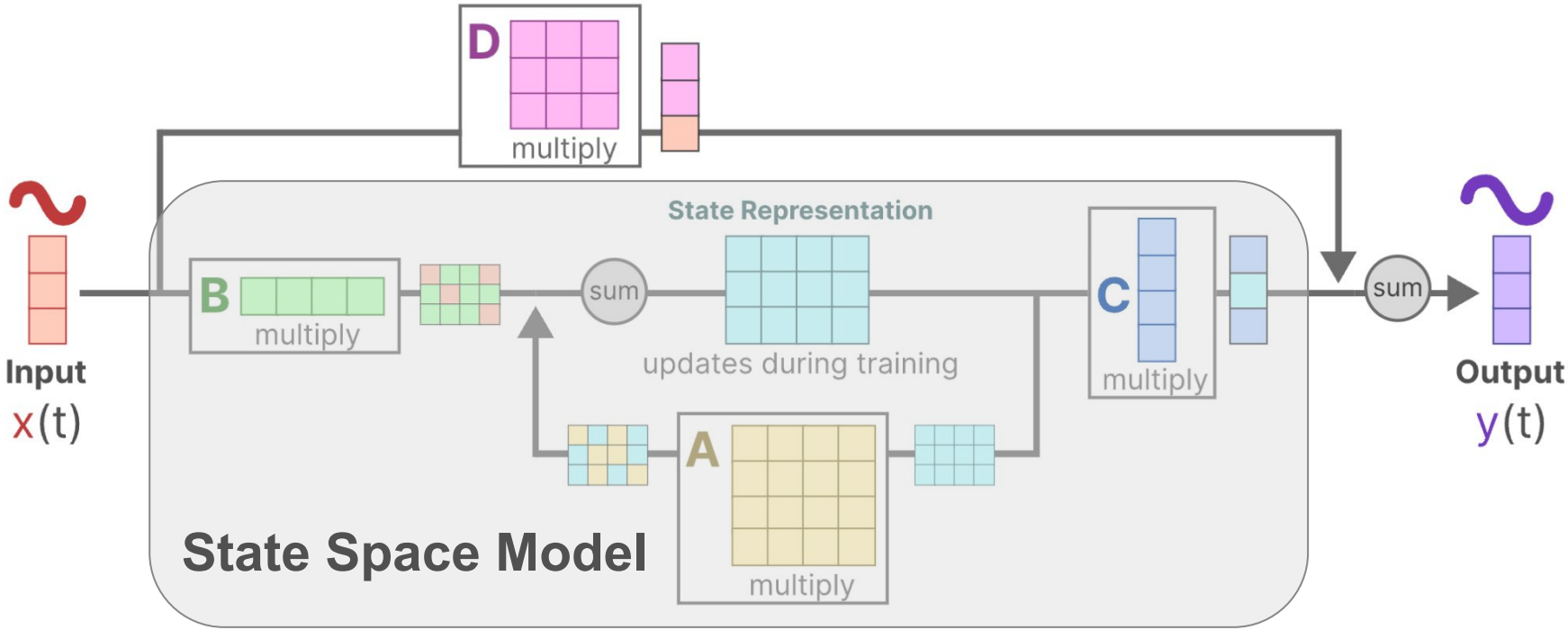
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

SSM Step-By-Step



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

SSM Step-By-Step

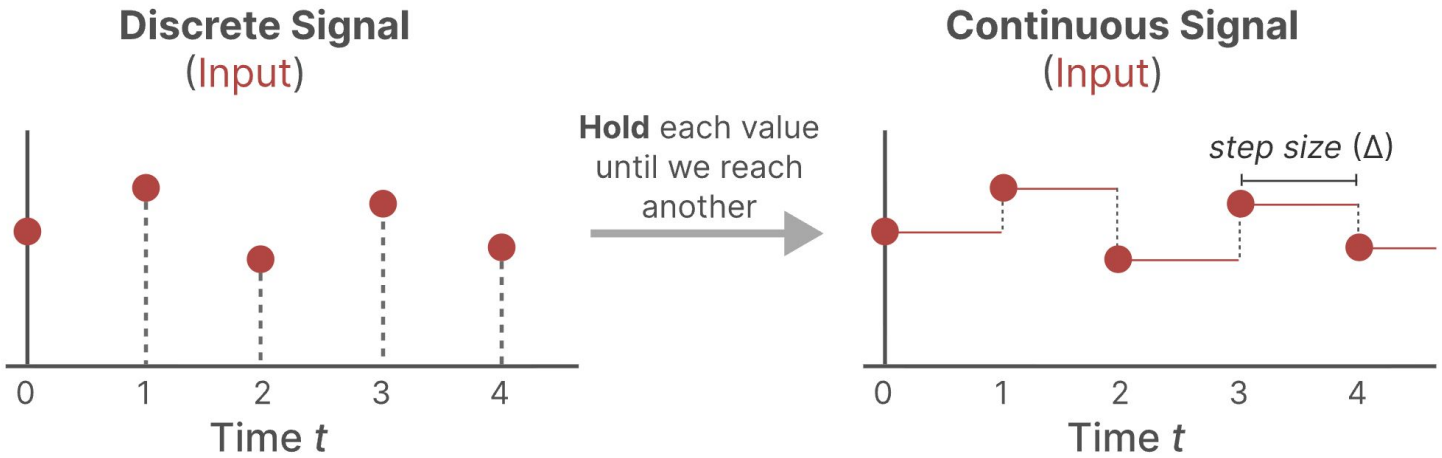


[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Discretization

- Finding h(t) is challenging if you have continuous signal
- What if input is discrete (e.g. text sequence)

- Discrete input signal -> continuous input signal: Zero order hold technique

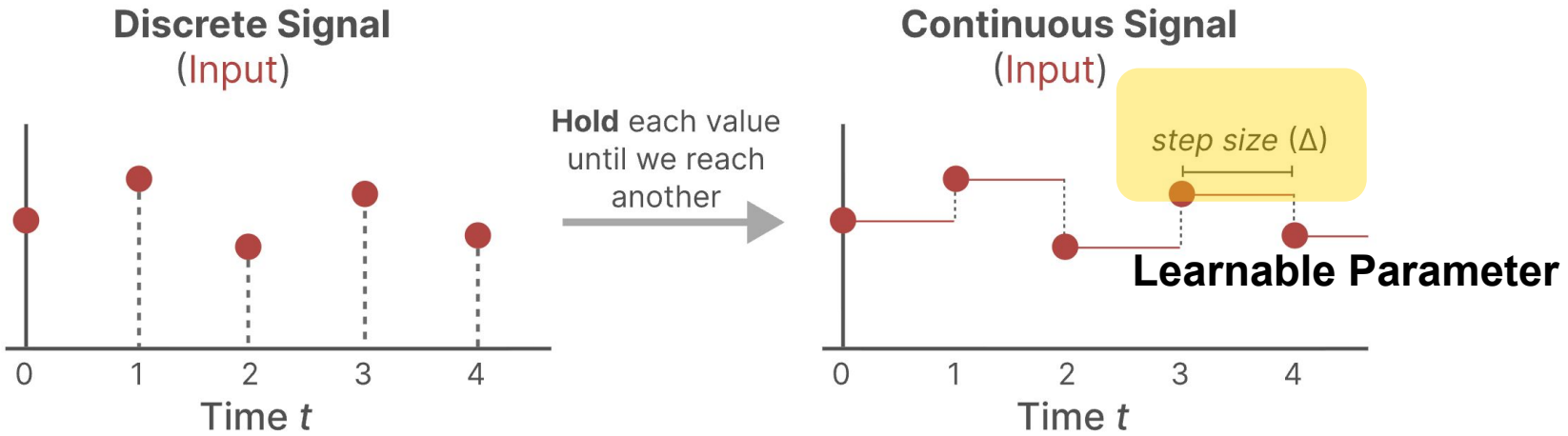


[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Discretization

- Finding h(t) is challenging if you have continuous signal
- What if input is discrete (e.g. text sequence)

- Discrete input signal -> continuous input signal: Zero order hold technique

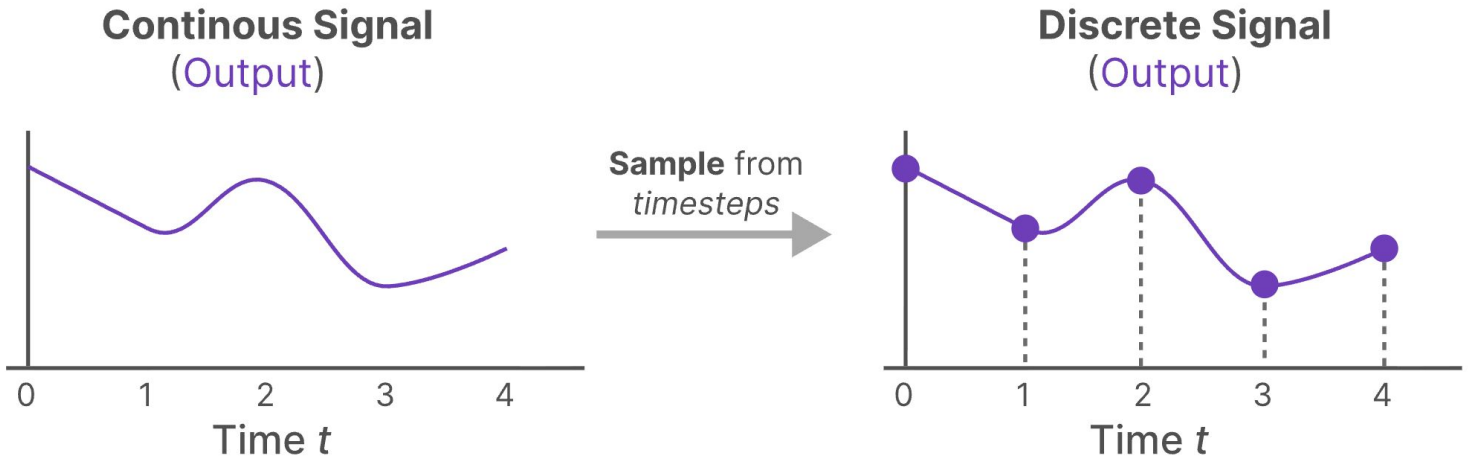


[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Discretization

- Finding h(t) is challenging if you have continuous signal
- What if input is discrete (e.g. text sequence)

- Continuous output signal -> discrete input signal



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Discretization: Mathematically

Discretized matrix **A**

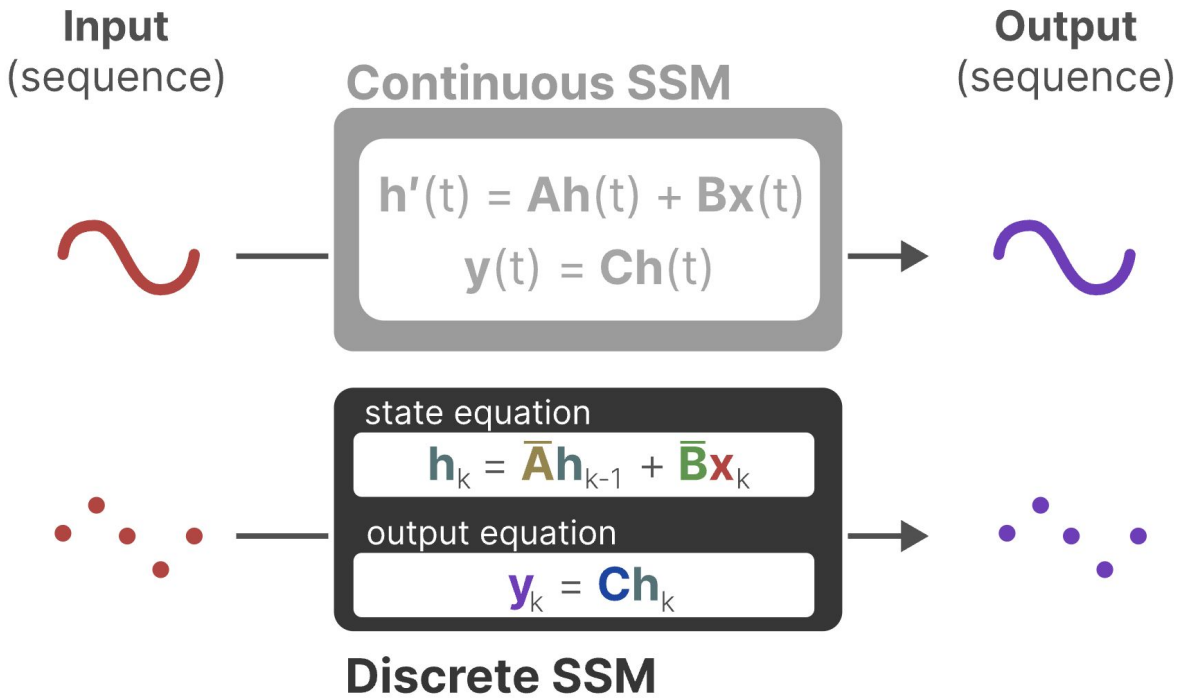
$$\overline{\mathbf{A}} = \exp(\Delta \mathbf{A})$$

Discretized matrix **B**

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

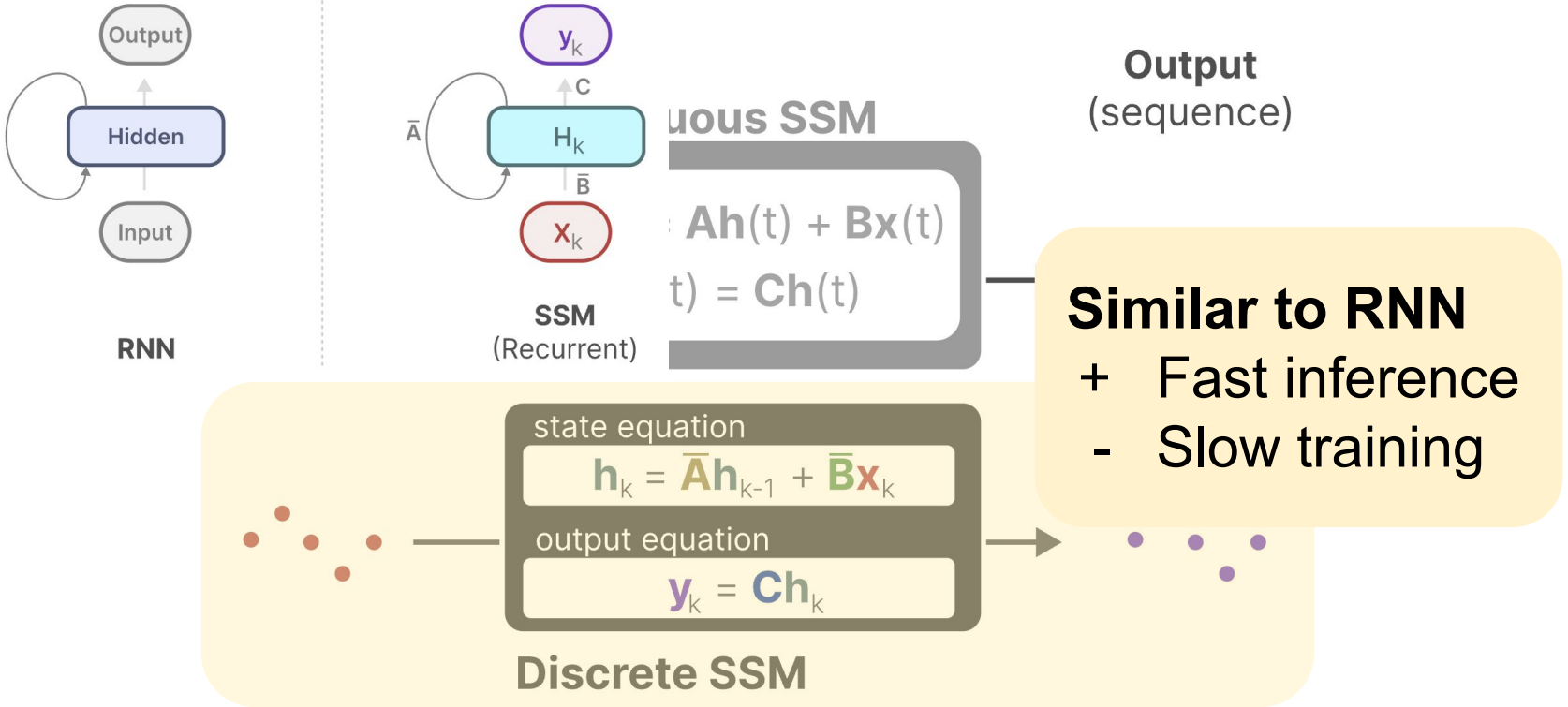
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Continuous and Discrete SSM



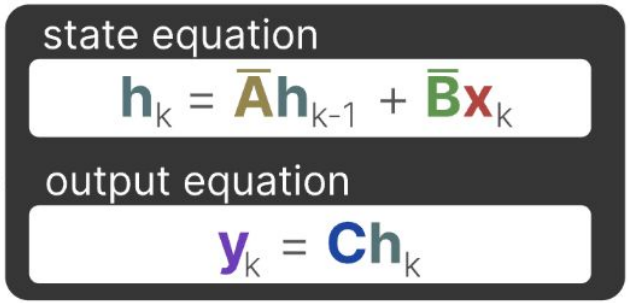
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Recurrent Representation



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Convolution Representation



State equation

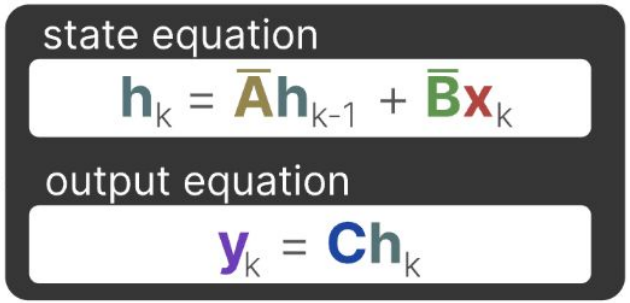
Step 0: $h_0 = Bx_0$
Step 1: $h_1 = \bar{A}h_0 + Bx_1 = \bar{A}Bx_0 + Bx_1$
Step 2: $h_2 = \bar{A}h_1 + Bx_2 = \bar{A}^2Bx_0 + \bar{A}Bx_1 + Bx_2$

Output equation

Step 0: $y_0 = Ch_0 = CBx_0$
Step 1: $y_1 = Ch_1 = C\bar{A}Bx_0 + CBx_1$
Step 2: $y_2 = Ch_2 = C\bar{A}^2Bx_0 + C\bar{A}Bx_1 + CBx_2$

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Convolution Representation



State equation

Step 0: $h_0 = Bx_0$
Step 1: $h_1 = \bar{A}h_0 + Bx_1 = \bar{A}Bx_0 + Bx_1$
Step 2: $h_2 = \bar{A}h_1 + Bx_2 = \bar{A}^2Bx_0 + \bar{A}Bx_1 + Bx_2$

Output equation

Step 0: $y_0 = Ch_0 = CBx_0$
Step 1: $y_1 = Ch_1 = C\bar{A}Bx_0 + CBx_1$
Step 2: $y_2 = Ch_2 = C\bar{A}^2Bx_0 + C\bar{A}Bx_1 + CBx_2$

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

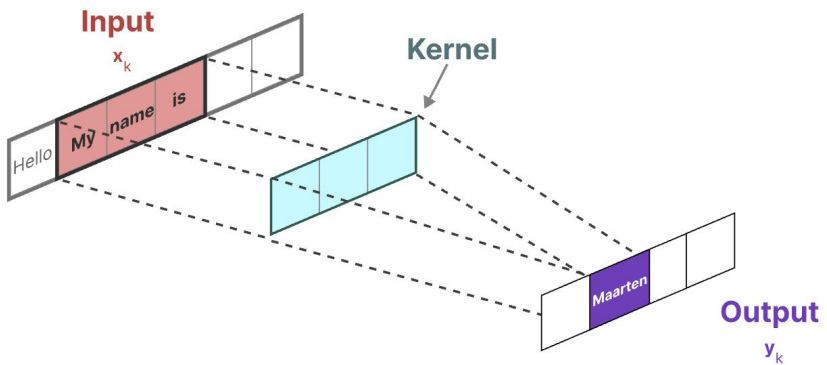
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Convolution Representation

$$\text{kernel} \longrightarrow \overline{\mathbf{K}} = (\overline{\mathbf{C}}\,\overline{\mathbf{B}}, \overline{\mathbf{C}}\,\overline{\mathbf{A}\mathbf{B}}, \ldots, \overline{\mathbf{C}}\,\overline{\mathbf{A}}^{k}\,\overline{\mathbf{B}}, \ldots)$$

$$\mathbf{y} = \mathbf{x} * \overline{\mathbf{K}}$$

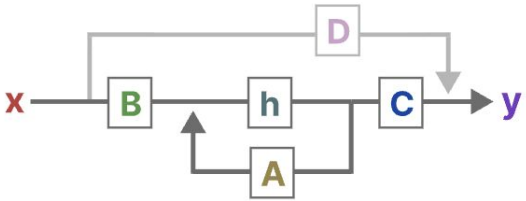
output input kernel



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

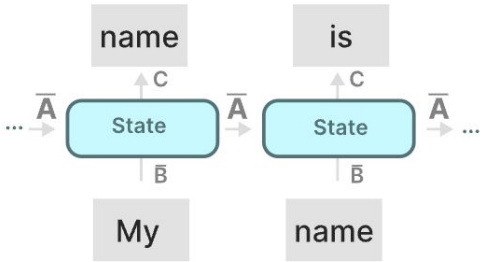
SSM Three Representations

Continuous-time



Discretize

Recurrent

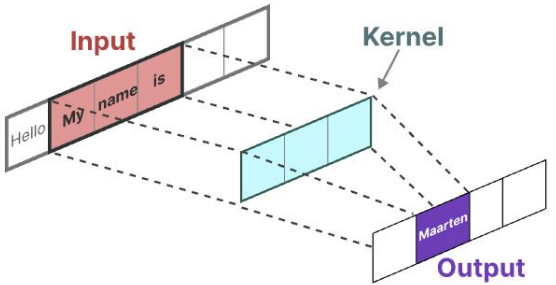


✓ *efficient inference*

✗ *parallelizable training*

or

Convolutional



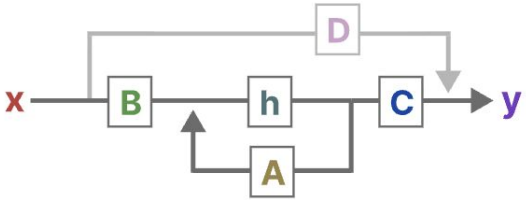
✗ *unbounded context*

✓ *parallelizable training*

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

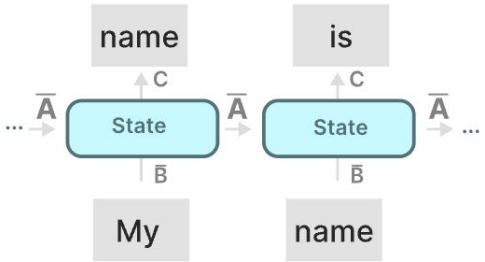
SSM Three Representations

Continuous-time



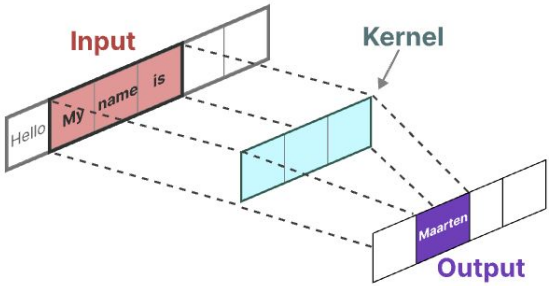
Discretize

Recurrent



or

Convolutional



✓ *efficient inference*

✗ *parallelizable training*

Inference Mode

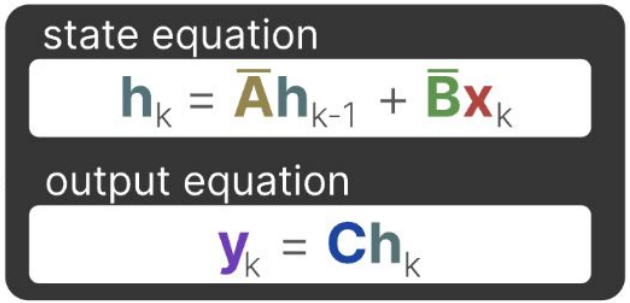
✗ *unbounded context*

✓ *parallelizable training*

Training Mode

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models. https://www.maartengrootendorst.com/blog/mamba/

Linear Time Invariant (LTI)



- SSM parameters are fixed for all timestamps
- A, B, C matrices are the same for every generated token by SSM
  - Static
  - Not content-aware

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

'A' Matrix

- Captures information about previous state to generate the new state

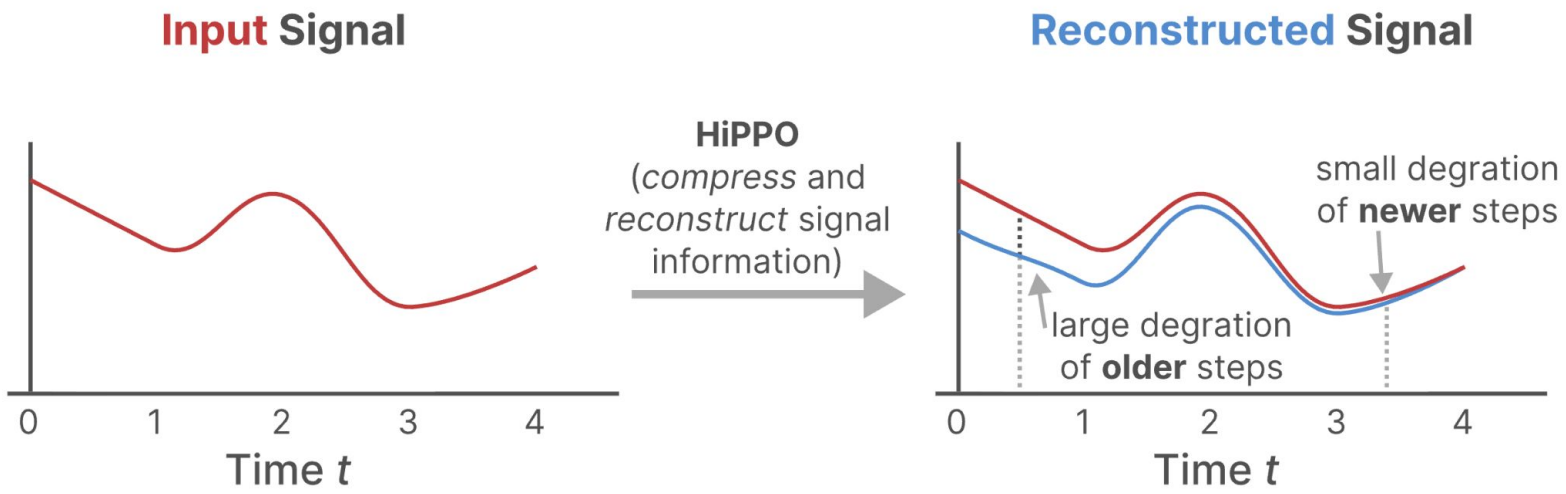
- Depending on the value of matrix A:
  - Remembering only a few previous tokens
    or
  - Capturing every token we have seen thus far

- how can we create matrix A in a way that retains a large memory (context size)?
  - **HiPPO**: High-order Polynomial Projection Operators

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

HiPPO

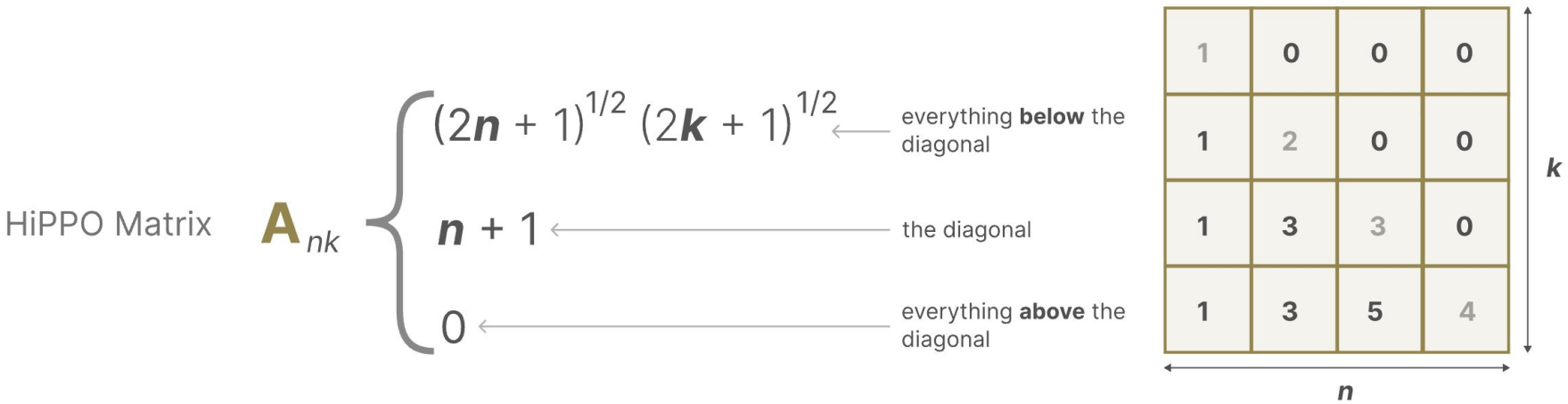
- Compress all input signals it has seen thus far into a vector of coefficients



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

HiPPO

- Capture recent tokens well
- Decays older tokens



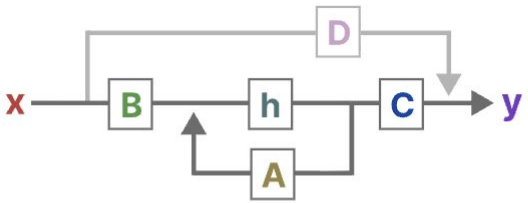
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

The S4 SSM

Structured State Spaces for Sequences (S4)

=

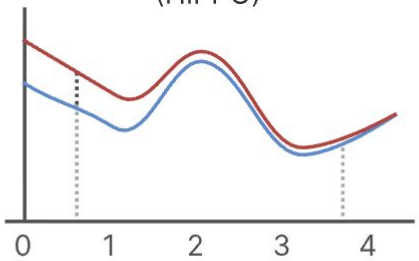
Continuous State Space



+

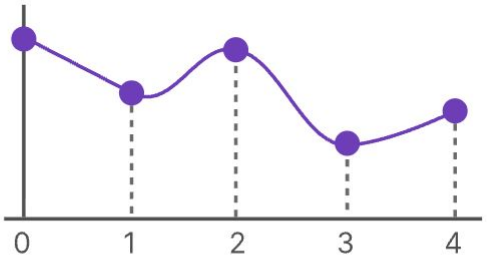
Long-Range Dependencies

(HiPPO)



+

Discrete Representations



Training mode (convolutional)

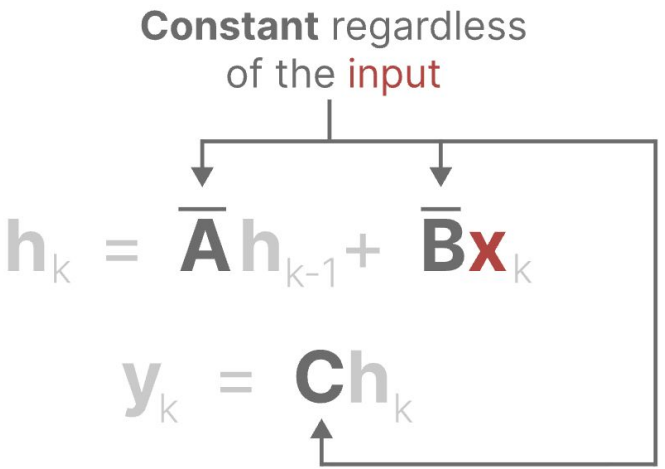
Inference mode (recurrence)

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

The S4 SSM

+ Handle long sequences

- Perform poorly on certain tasks that require the ability to focus on or ignore particular inputs
  - Selective copying
  - Induction heads



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Mamba: Dynamic Matrices

- Compress data selectively into the state
  - Parameters need to be dependent on input

- Mamba makes matrices B, C, and step size Δ dependent on input
  - For every input token, different B and C
  - Context-aware
  - Smaller Δ, ignores the word using context more
  - Bigger Δ, focuses on the word more than context

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Mamba: Dynamic Matrices

- Matrix A remains static
  - State is static
  - The way it is influenced (B and C) is dynamic

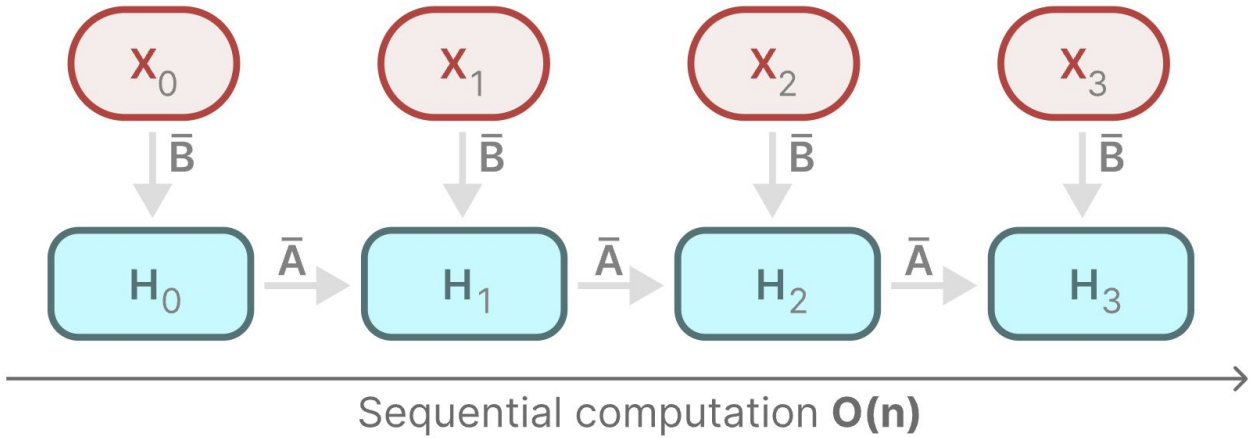
- Cannot use convolution representation
  - Matrices are dynamic
  - Convolution representation assumes a fixed kernel

- Only the recurrent representation
  - Lose the parallelization

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Mamba: Parallel Scan

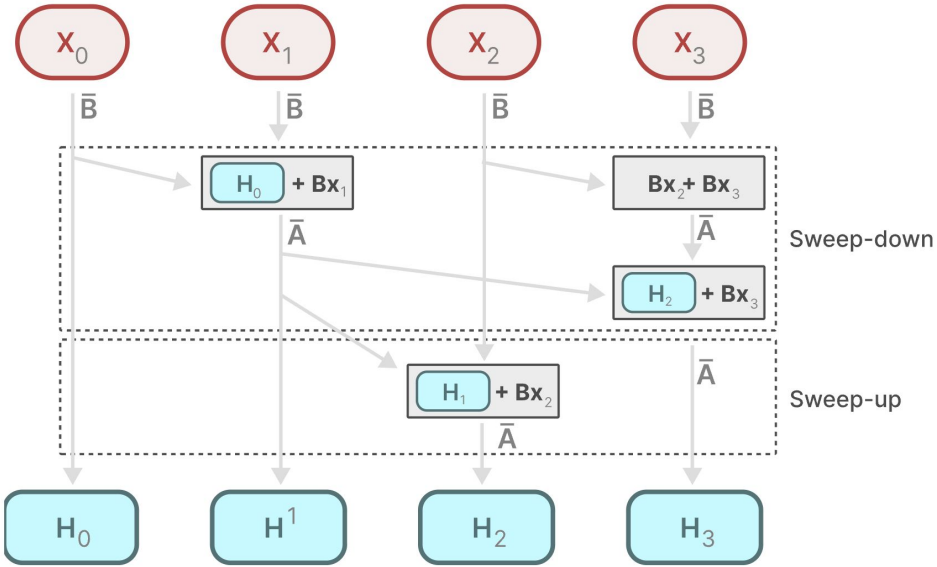
- Recurrent computations (scan operation)



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Mamba: Parallel Scan

- Calculate sequences in parts and iteratively combine



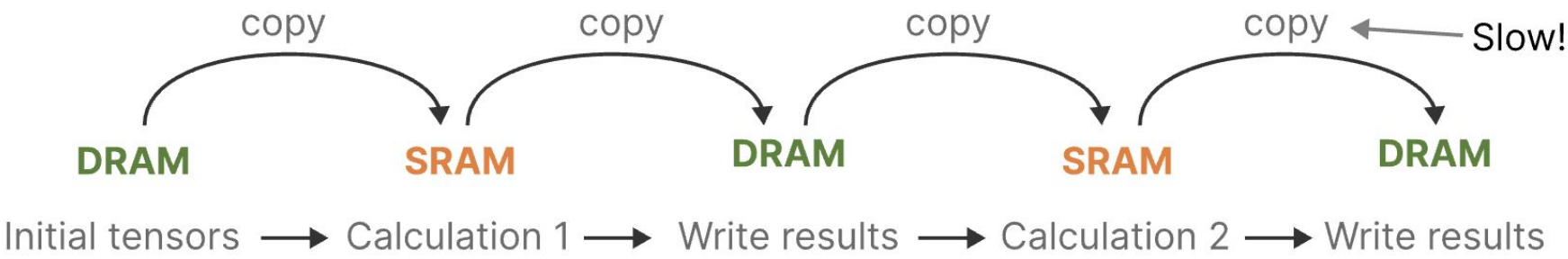
Parallel computation **O(n/t)**

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Mamba: Hardware

- SRAM: small but highly efficient
- DRAM: large but less efficient

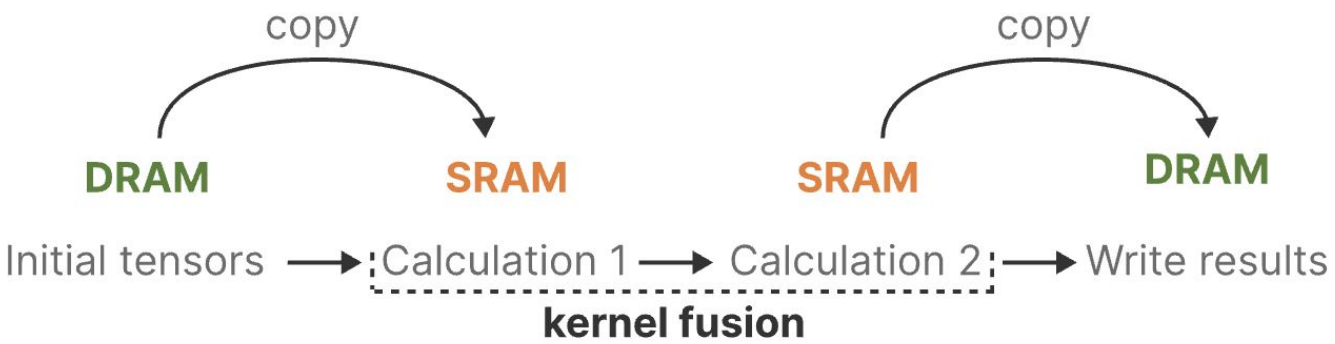
- Transfers between SRAM and DRAM are slow



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

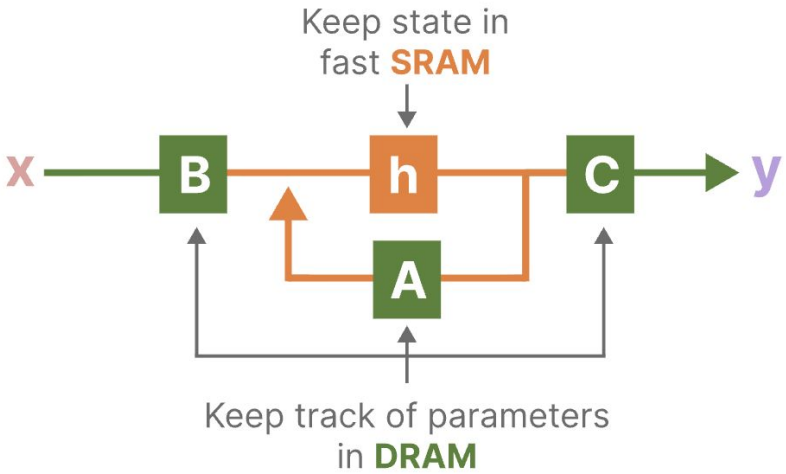
Mamba: Hardware

- Limit the number of SRAM-DRAM transfers
- Kernel fusion: continuously perform computations until it's done



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

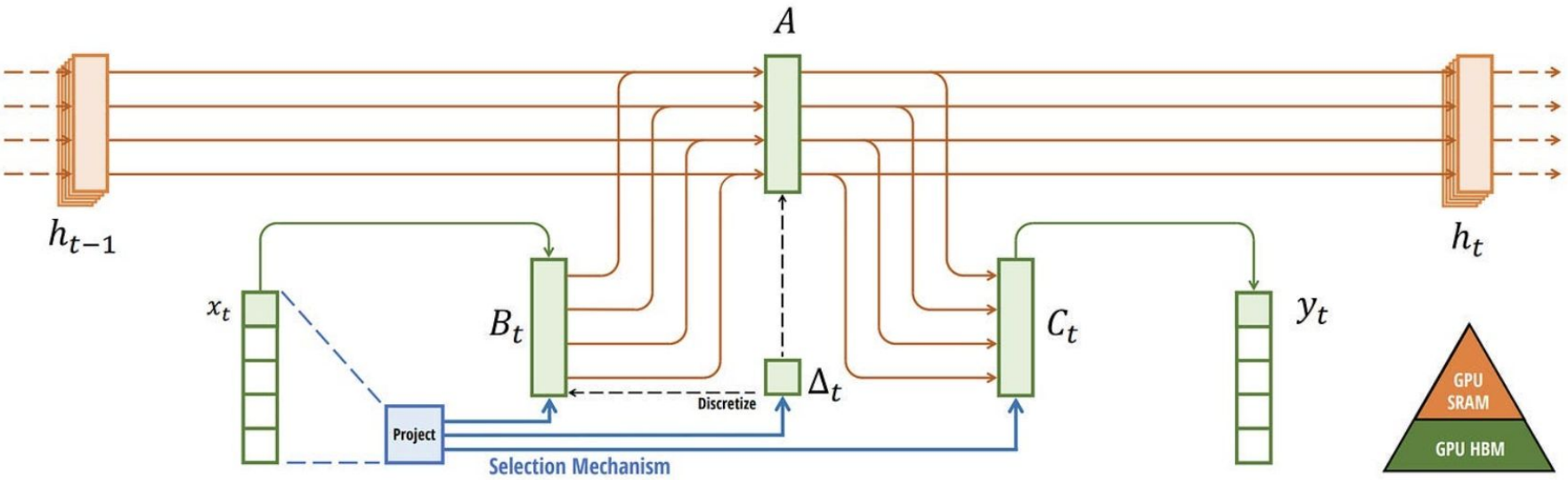
Mamba: Hardware



- Intermediate states are not saved but recomputed whenever necessary
- Seems inefficient but much less costly than reading intermediate states from DRAM

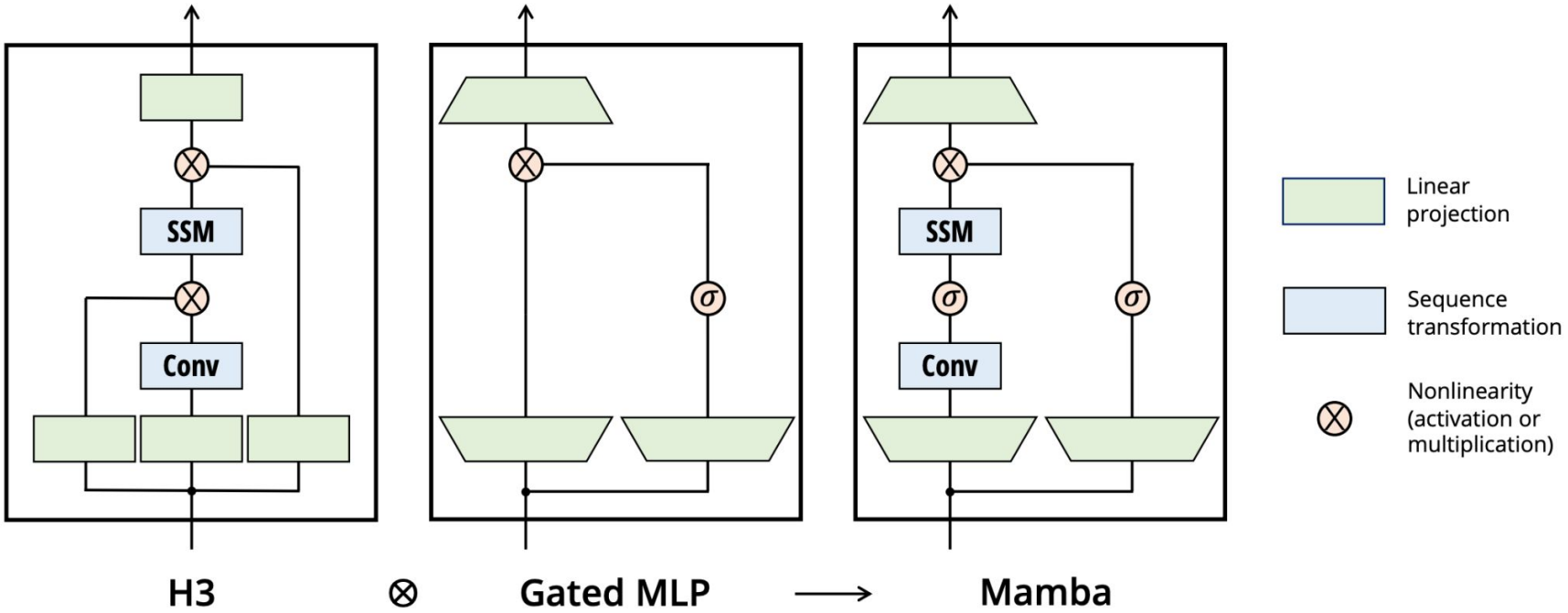
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Mamba: Selective SSM / S6 Model



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

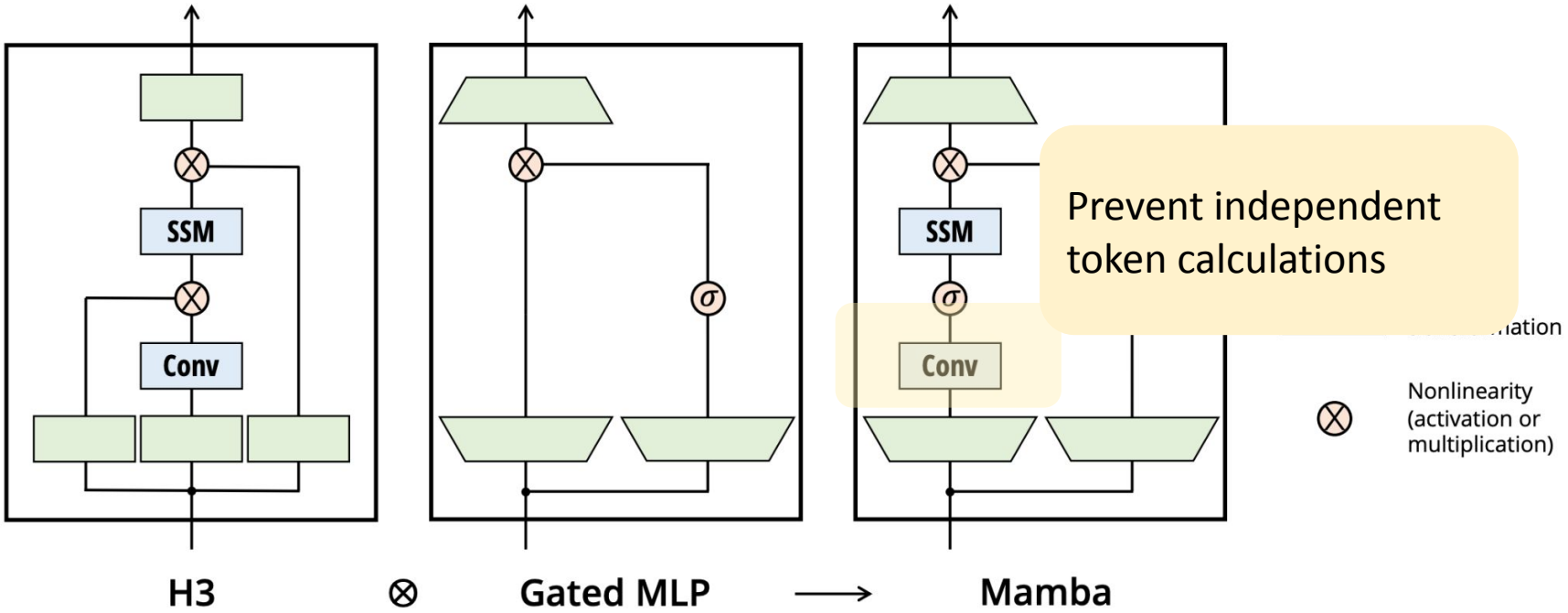
Mamba: Mamba Block



Basis of most SSM architectures

[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Mamba: Mamba Block



Basis of most SSM architectures

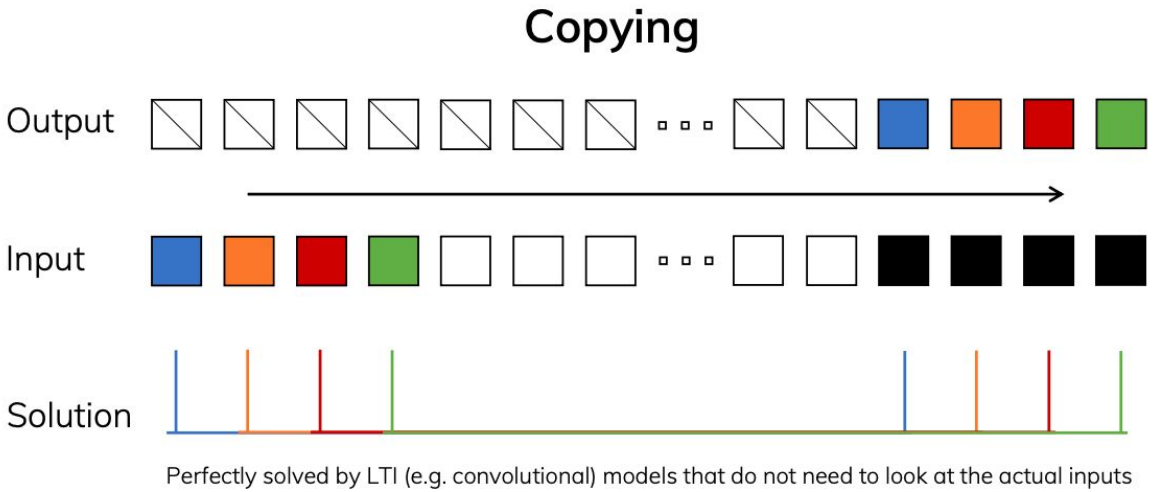
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Mamba

	Training	Inference
Transformers	**Fast!** (parallelizable)	**Slow...** (scales **quadratically** with sequence length)
RNNs	**Slow...** (not parallelizable)	**Fast!** (scales **linearly** with sequence length)
Mamba	**Fast!** (parallelizable)	**Fast!** (scales **linearly** with sequence length)

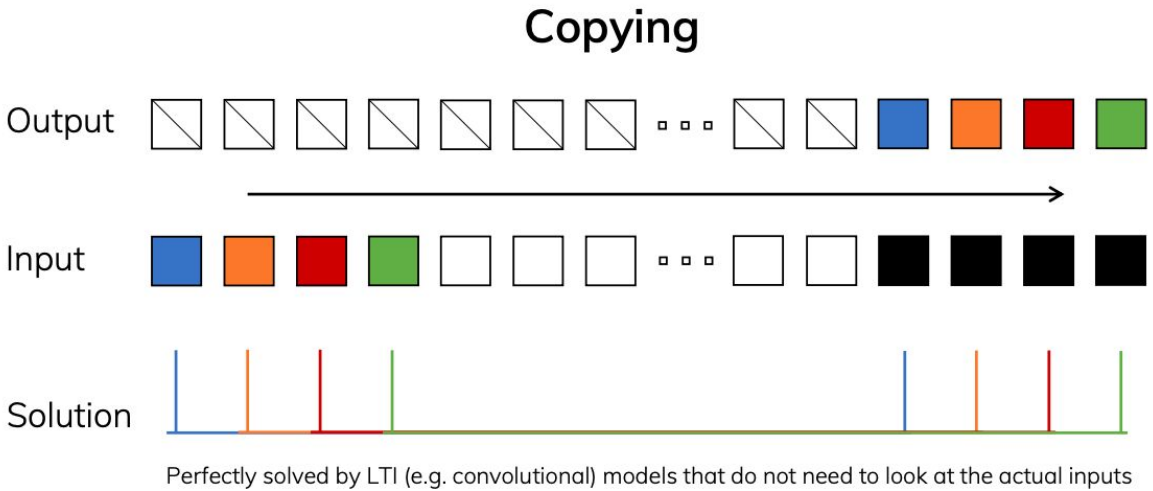
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Experiments: Synthetic Tasks



Perfectly solved by **Linear Time Invariant (LTI)** SSMs!

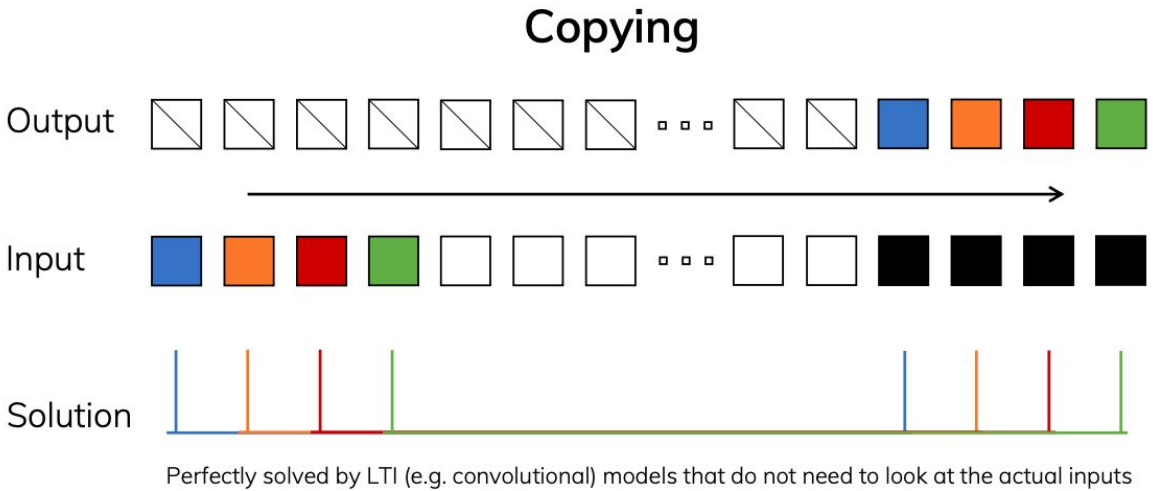
Experiments: Synthetic Tasks



Perfectly solved by **Linear Time Invariant (LTI)** SSMs!

$$y = K * x = \sum_{k=0}^{t} K_k x_{t-k}$$

Experiments: Synthetic Tasks

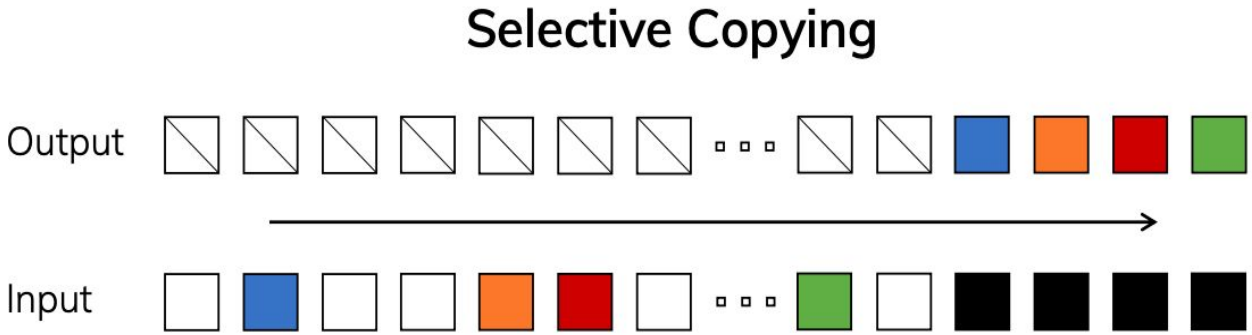


Perfectly solved by **Linear Time Invariant (LTI)** SSMs!

$$y = K * x = \sum_{k=0}^{t} K_k x_{t-k}$$

Different kernel for different time/position

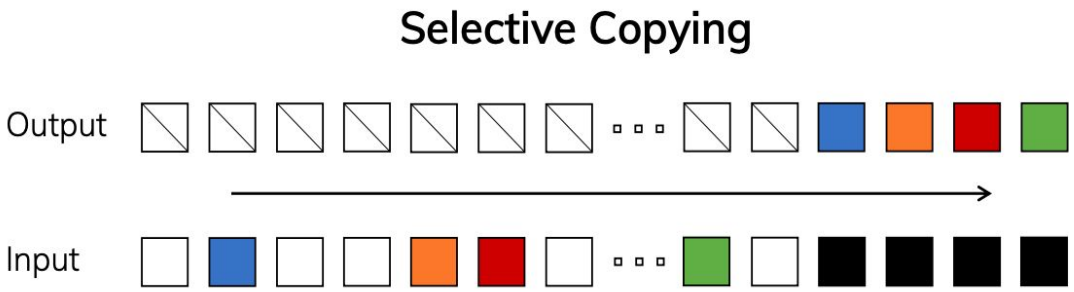
Experiments: Synthetic Tasks



Can’t just memorize the timesteps!

- Need to actually look at the **content** of the input sequence
- SSM + **Selection** (i.e. data-dependent dynamics)

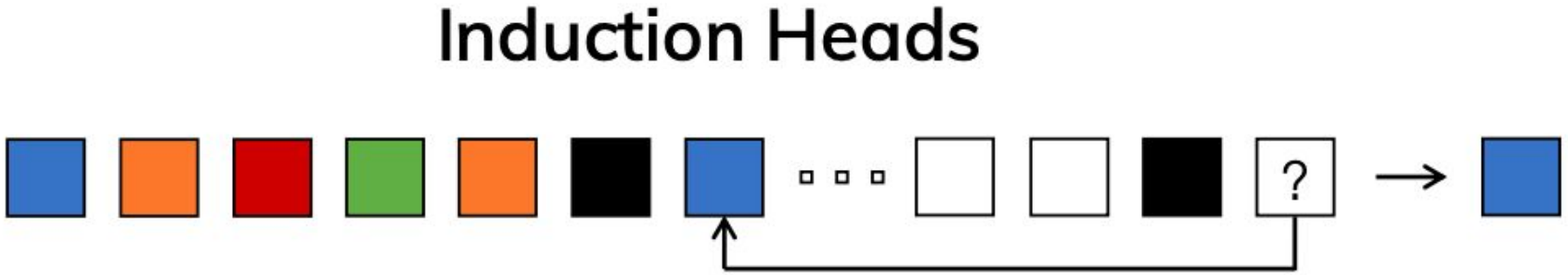
Experiments: Synthetic Tasks



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.

Experiments: Synthetic Tasks



- Context-based retrieval and copy
- Also requires data-dependent dynamics

Experiments: Synthetic Tasks

Perfect performance!

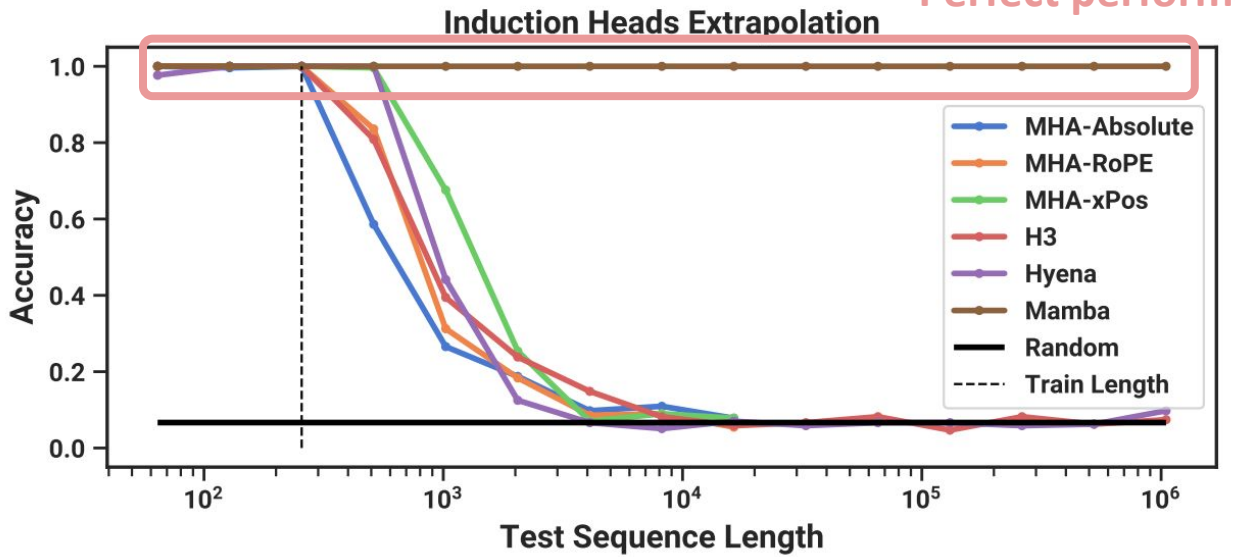


Table 2: (**Induction Heads**.) Models are trained on sequence length $2^8 = 256$, and tested on increasing sequence lengths of $2^6 = 64$ up to $2^{20} = 1048576$. Full numbers in Table 11.

Language Modeling: Scaling Law

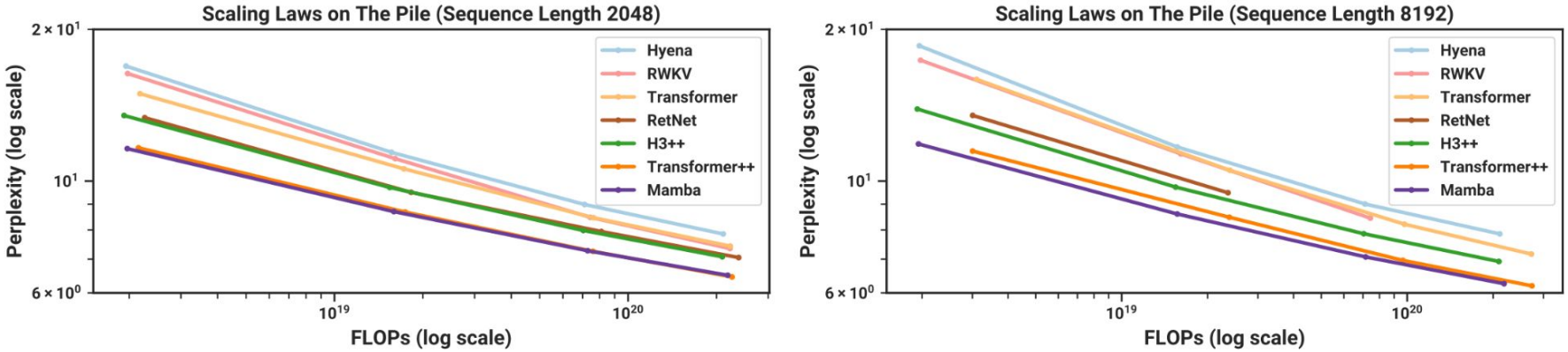


Figure 4: (**Scaling Laws.**) Models of size $\approx 125M$ to $\approx 1.3B$ parameters, trained on the Pile. Mamba scales better than all other attention-free models and is the first to match the performance of a very strong "Transformer++" recipe that has now become standard, particularly as the sequence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

- Mamba scales **better than other attention-free** models
- Mamba is the **first attention-free** approach to match a very strong Transformer

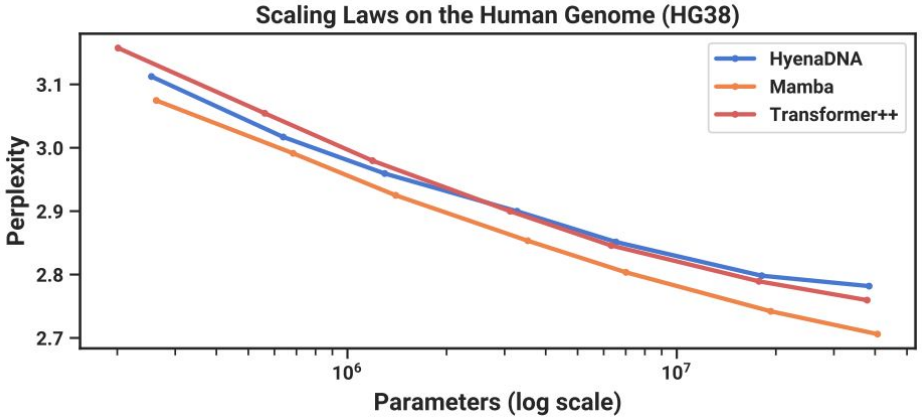
Language Modeling: Downstream Tasks

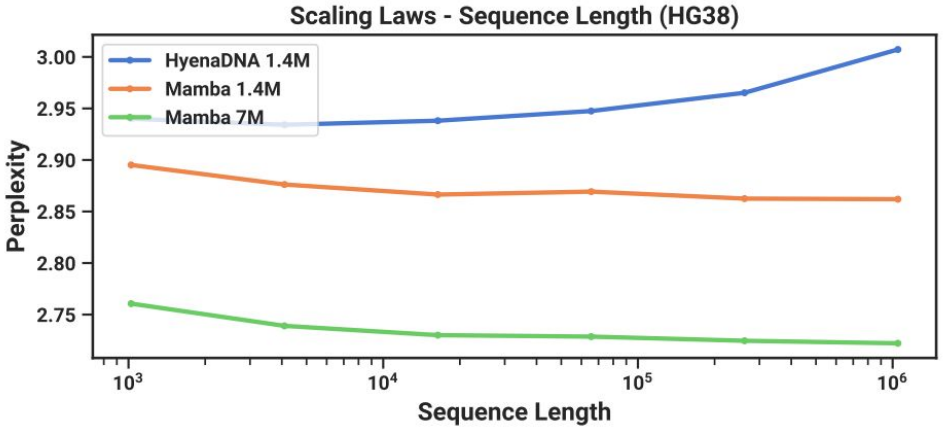
Table 3: (**Zero-shot Evaluations**.) Best results for each size in bold. We compare against open source LMs with various tokenizers, trained for up to 300B tokens. Pile refers to the validation split, comparing only against models trained on the same dataset and tokenizer (GPT-NeoX-20B). For each model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

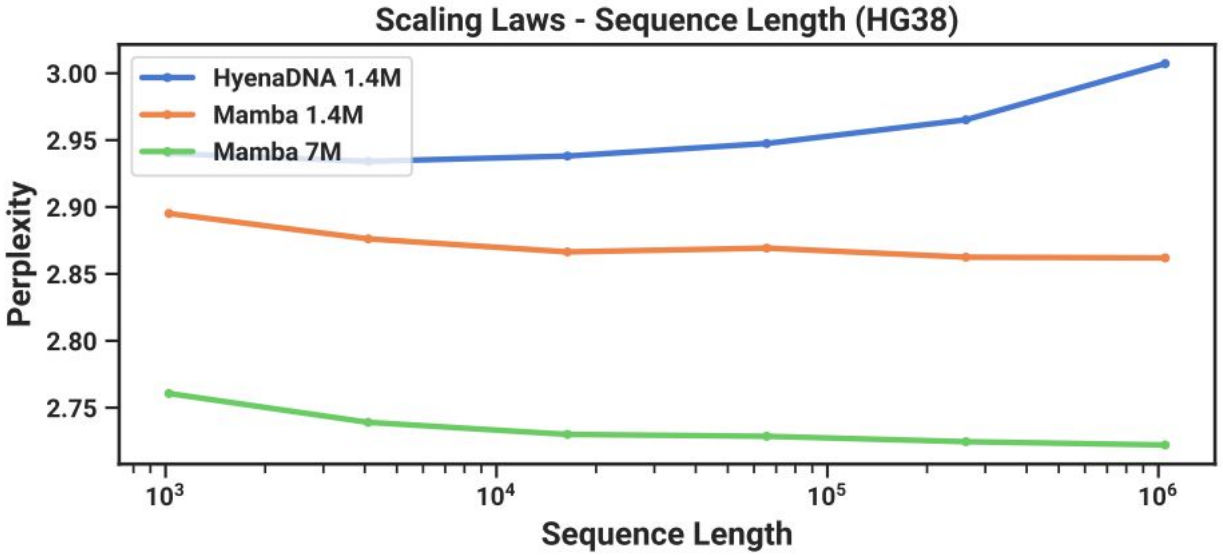
DNA Modeling

- Pretrain on HG38 human genome tokens (next-token prediction)
- **Size scaling:** Mamba improves smoothly and scales better than HyenaDNA / Transformer++
- **Length scaling:** Mamba keeps improving up to ~1M tokens; HyenaDNA degrades with length





DNA Modeling



Intuition on context length scaling: Without the selection mechanism, the long input sequence may be very **noisy**!

DNA Modeling: Synthetic Species Classification

- Classify between the five great apes species (human, chimpanzee, gorilla, orangutan, bonobo).
- Shares 99% of their DNA!

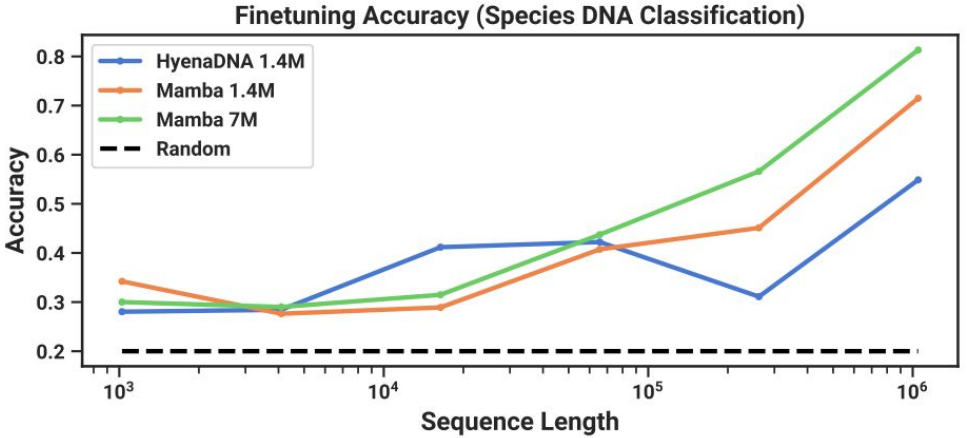


Figure 6: (**Great Apes DNA Classification.**) Accuracy after fine-tuning on sequences of length $2^{10} = 1024$ up to $2^{20} = 1048576$ using pretrained models of the same context length. Numerical results in Table 13.

Audio Modeling

- **Baseline:** SaShiMi architecture (U-Net with alternating S4 and MLP blocks)
- Replace S4+MLP blocks with Mamba blocks
- **Dataset:** YouTubeMix (4 hours of solo piano music)
- Uses complex number parameterization

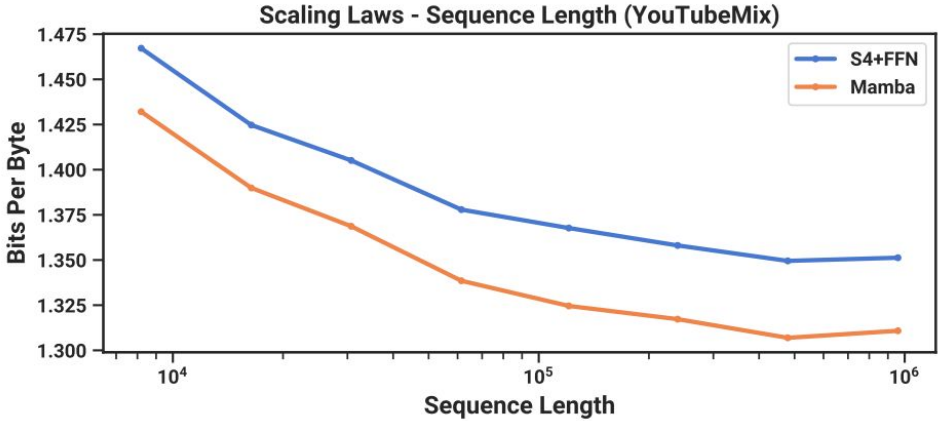


Figure 7: (**Audio Pretraining**.) Mamba improves performance over prior state-of-the-art (Sashimi) in autoregressive audio modeling, while improving up to minute-long context or million-length sequences (controlling for computation).

Audio Generation

- **Benchmark:** SC09, 1-second clips of digits “zero” through “nine”
- Small Mamba model outperforms SOTA (and much larger) GAN or diffusion models

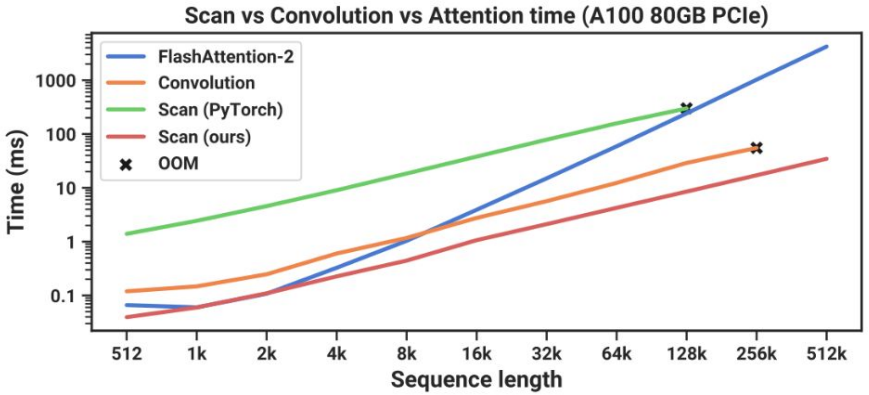
Table 4: (**SC09**) Automated metrics for unconditional generation on a challenging dataset of fixed-length speech clips. (*Top to Bottom*) Autoregressive baselines, non-autoregressive baselines, Mamba, and dataset metrics.

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**

Speed and Throughput



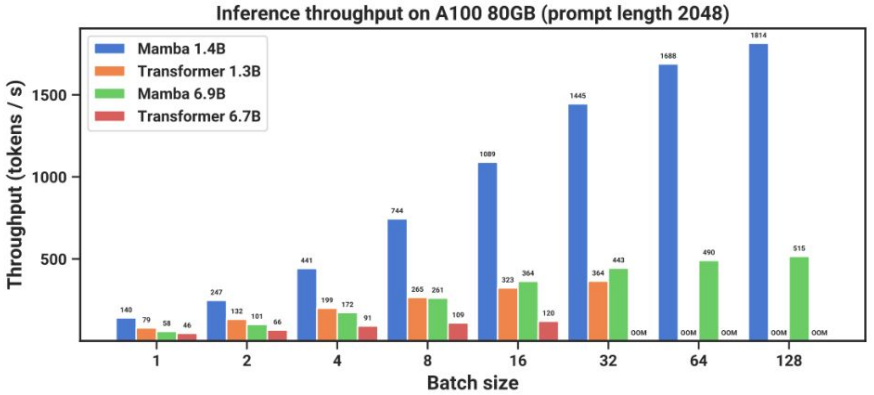


Figure 8: (**Efficiency Benchmarks.**) (*Left*) Training: our efficient scan is 40× faster than a standard implementation. (*Right*) Inference: as a recurrent model, Mamba can achieve 5× higher throughput than Transformers.

Ablation

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**

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

Summary

- State space models (SSMs) become **competitive with Transformers** once you add
  - ***Selectivity*** (input-dependent dynamics): choose what to keep/forget based on content
  - Implement the computation in a **hardware-aware way**.

- **Possible future directions:**
  - Have a better theoretical algorithm for the recurrence that does not rely on specific hardware accelerations
  - **Mamba 2:** More theory; connects SSMs and attentions through a special family of matrices
  - **Mamba 3:** A more expressive discretization rule; complex-valued state space update; more efficient decoding

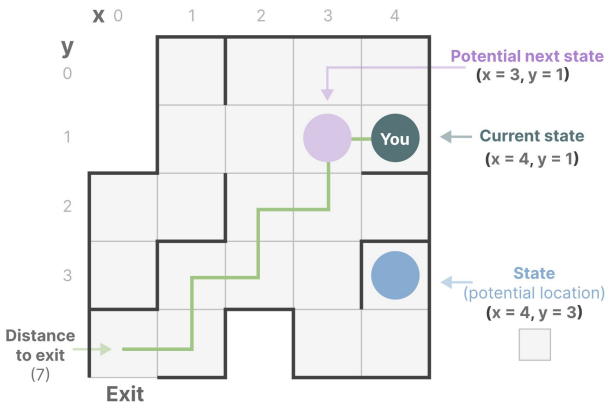
END

BACKUP

Space State

- A State Space contains the minimum number of variables that fully describe a system

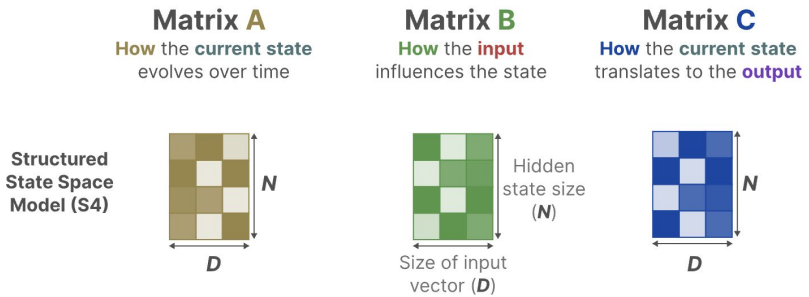
- It is a way to mathematically represent a problem by defining a system's possible states



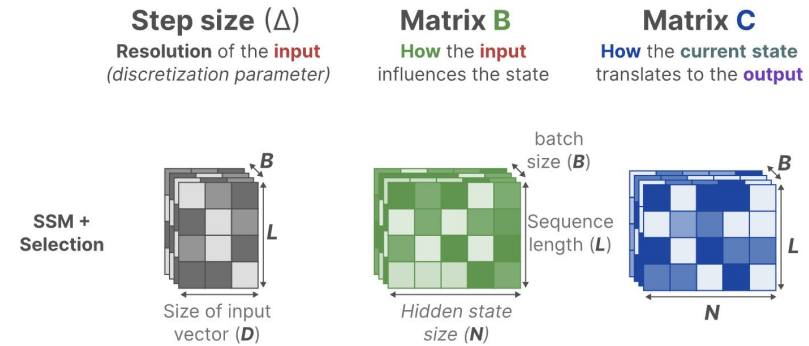
[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/

Selective Scan Algorithm

In a Structured State Space Model (S4), the matrices A, B, and C are independent of the input since their dimensions **N** and **D** are static and do not change.



Instead, Mamba makes matrices B and C, and even the *step size* Δ, dependent on the input by incorporating the sequence length and batch size of the input:



[*] Maarten Grootendorst. A Visual Guide to Mamba and State Space Models.https://www.maartengrootendorst.com/blog/mamba/