Mamba: Linear-Time Sequence Modeling with Selective State Spaces

Deep Dive into SSM

허세민

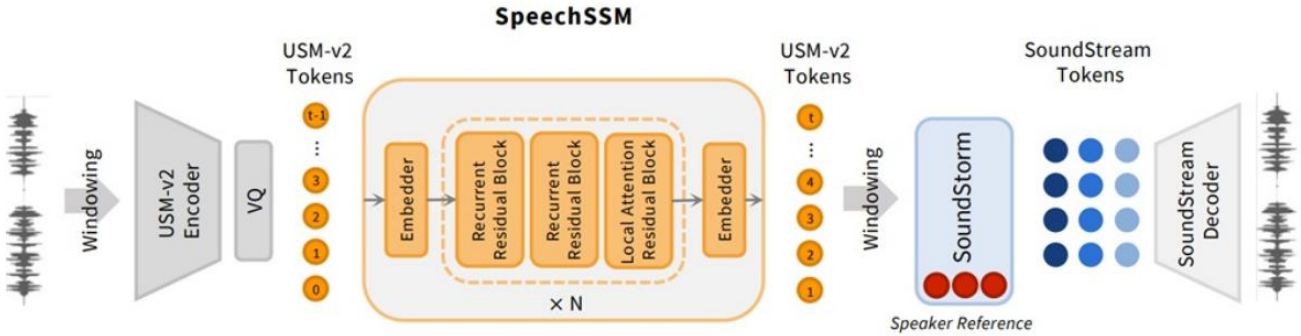
Recently...

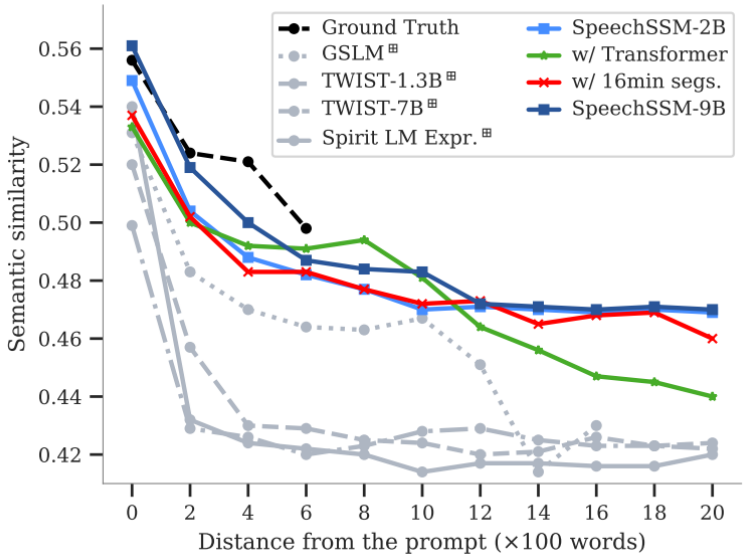
25.07.03 09:24 | 최종 업데이트 25.07.03 09:47

KAIST, 24시간 음성 생성하는 인공지능 비서 실현할 '스피치SSM' 개발

기존 음성 언어 모델 한계 극복, 시간 제약 없는 고품질 음성 생성 가능... ICML 2025서 구두 논문 발표

유창재 (karma50)



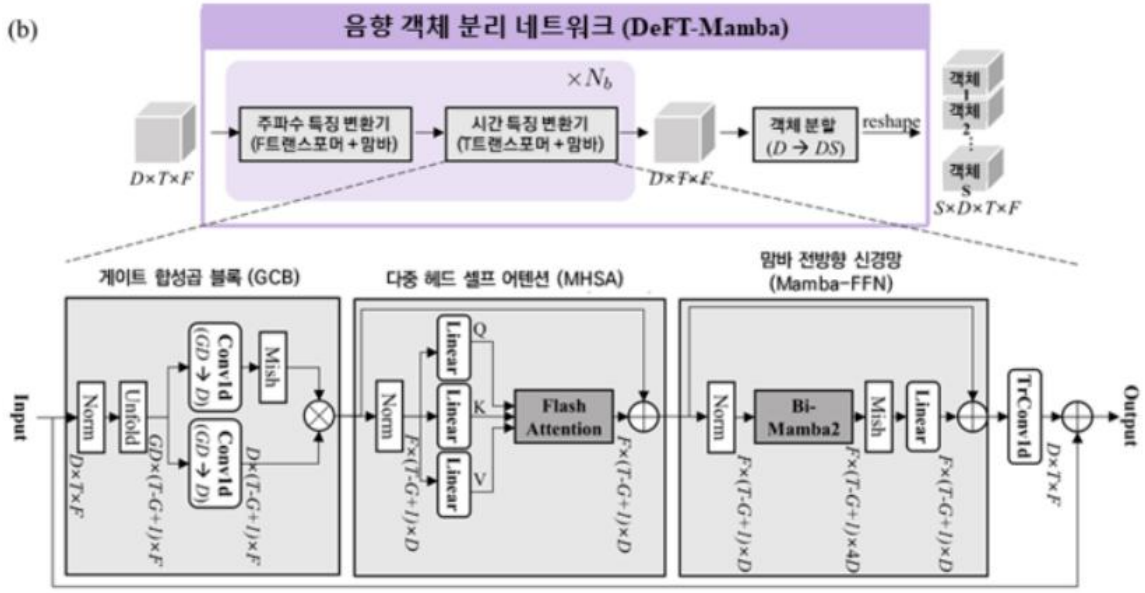


Recently...

HOME > 산업 > AI·자동화·자율생산 > 기업 · 피플

KAIST 최정우 교수팀, 세계 최고 음향 AI 챌린지 세계 1위 쾌거

박현우 기자 | 승인 2025.07.11 09:23 | 댓글 0



	Submission Information		Evaluation Set	
	Submission Code	Technical Report	Official Team Rank	CA-SDRi (eval)
1	Choi_KAIST_task4_3		1	11.00
2	Morocutti_CPJKU_task4_2		2	9.77
3	Wu_SUSTech_task4_submission_2		3	9.73
4	Qian_SJTU_task4_1		4	7.84
5	Bando_AIST_task4_2		5	7.55
6	Park_GIST-HanwhaVision_task4_4		6	6.69
7	Stergioulis_UTh_task4_submission_1		7	6.60
8	Baseline_task4_ResUNetK		8	6.60
9	Zhang_BUPT_task4_1		9	3.84

Author

Albert Gu

팔로우

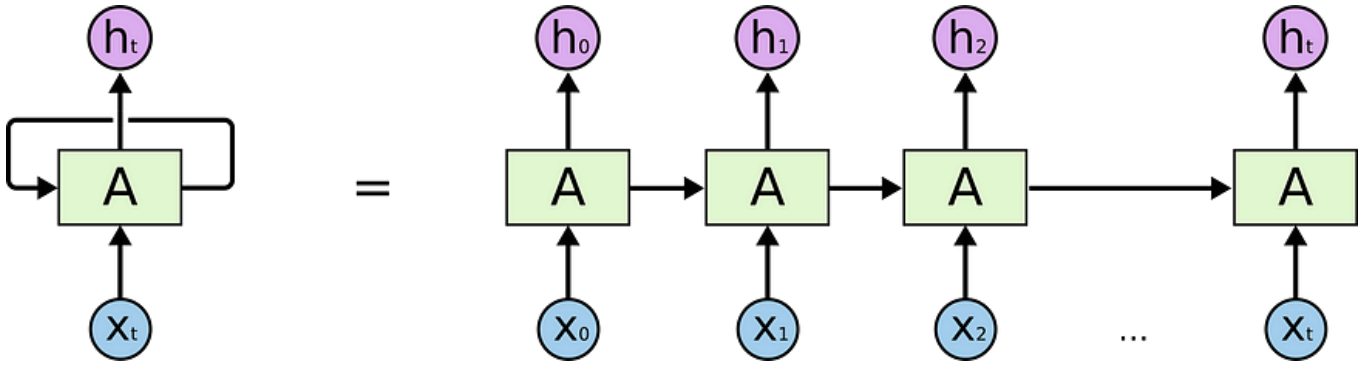
Carnegie Mellon University
andrew.cmu.edu의 이메일 확인됨
Machine Learning

제목	인용	연도
Mamba: Linear-time sequence modeling with selective state spaces A Gu, T Dao arXiv preprint arXiv:2312.00752	4319	2023
Efficiently Modeling Long Sequences with Structured State Spaces A Gu, K Goel, C Ré International Conference on Learning Representations (ICLR) 10	2640	2022
Combining recurrent, convolutional, and continuous-time models with linear state space layers A Gu, I Johnson, K Goel, K Saab, T Dao, A Rudra, C Ré Advances in neural information processing systems 34, 572-585	810	2021
Hippo: Recurrent memory with optimal polynomial projections A Gu, T Dao, S Ermon, A Rudra, C Ré Advances in neural information processing systems 33, 1474-1487	730	2020
Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality T Dao, A Gu arXiv preprint arXiv:2405.21060	692	2024
Representation tradeoffs for hyperbolic embeddings F Sala, C De Sa, A Gu, C Ré International conference on machine learning, 4460-4469	552	2018
On the parameterization and initialization of diagonal state space models A Gu, K Goel, A Gupta, C Ré Advances in Neural Information Processing Systems 35, 35971-35983	464	2022
Diagonal state spaces are as effective as structured state spaces A Gupta, A Gu, J Berant Advances in neural information processing systems 35, 22982-22994	410	2022

	Mamba	Mamba-2
Submit (arXiv)	December 20, 2023	May 31, 2024
1st Submit (Conference)	ICML 2024	ICML 2024
Final Acceptance	COLM 2024	ICML 2024

Background

RNN(Recurrent Neural Network)



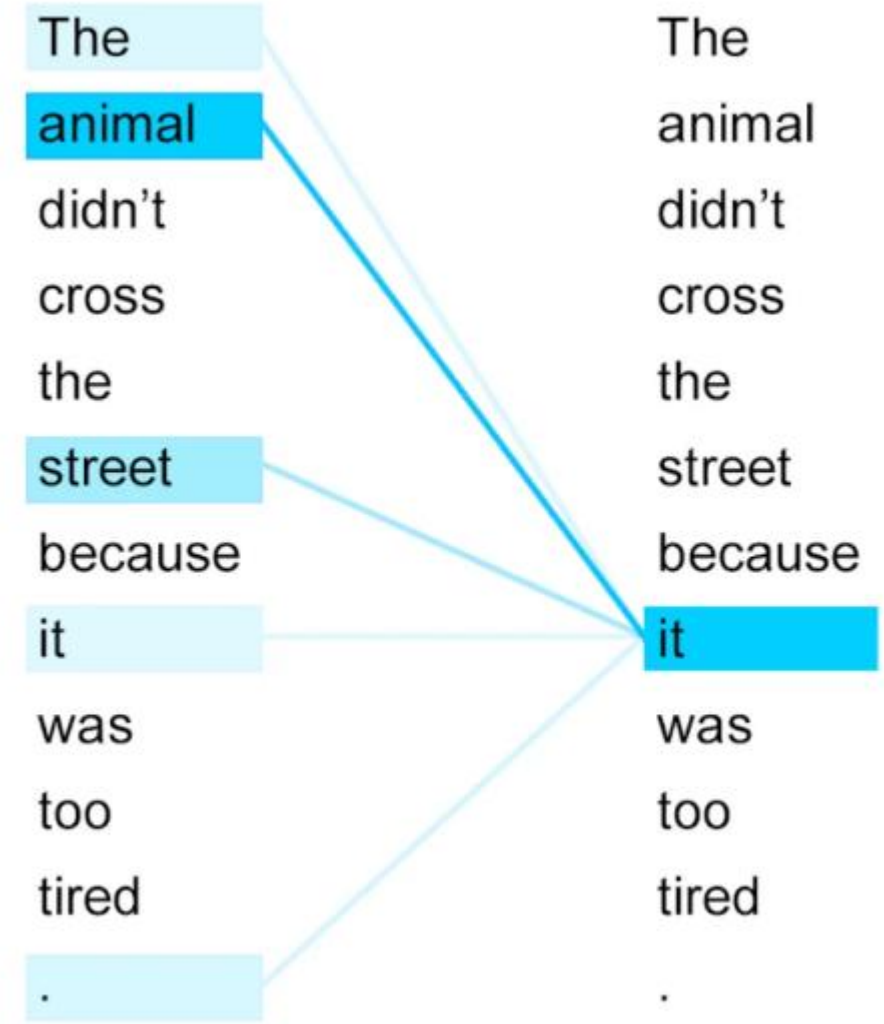
$$h_t = f(W_X X_t + W_h h_{t-1} + b_h)$$
$$y_t = f(W_y h_t + b_y)$$

Advantages
- Naturally models temporal sequences
- Causal and memory-efficient

Limitations
- Hard to learn long-term dependencies
- Slow training due to sequential computation

Alternatives
- LSTM (Long Short-Term Memory)
- GRU (Gated Recurrent Unit)

Attention



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Advantages

- Captures long-range dependencies effectively
- Allows full parallelization during training

Limitations

- Quadratic time & memory complexity
- Struggles with very long sequences

Alternatives

- FlashAttention
- Performer

Overview

Overview

We propose a new class of **selective state space models** that improves on prior work on several axes to achieve the **modeling power of Transformers while scaling linearly** in sequence length

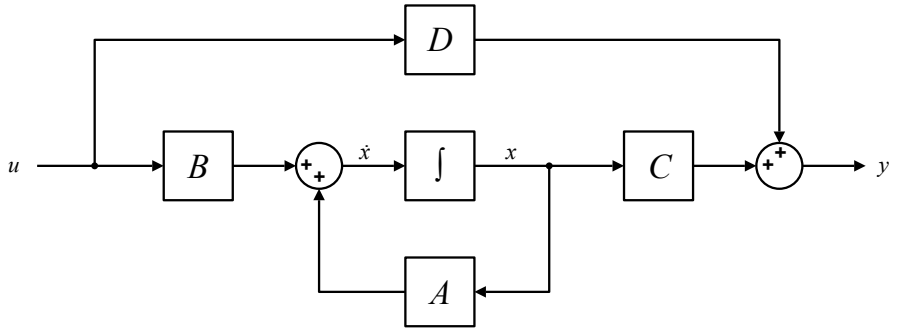
Space-State Model

What is a State Space Model?

시스템의 상태(state)와 입력(input)을 바탕으로
시간에 따른 동작을 수학적으로 모델링하는 방식

✓ A와 B는 Memory update에 관여
- A는 현재 Memory 상태를 transform, B는 현재 input을 transform
- 두 벡터의 합 → Memory 변화율

✓ C와 D는 Output projection에 관여
- C는 현재 Memory로부터, D는 현재 input으로부터 값을 받아 옴
- 두 스칼라의 합 → output
- D는 생략가능



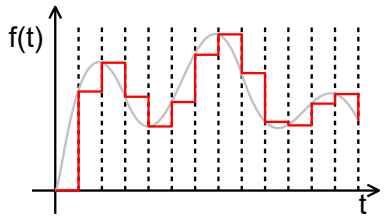
$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t)$$

Discretization

$$h'(t) = Ah(t) + Bx(t)$$
$$y(t) = Ch(t)$$

ZOH (zero-order hold)



$$h_t = \overline{A}h_{t-1} + \overline{B}x_t$$
$$y_t = Ch_t$$

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

$$\frac{d}{dt}h(t) = Ah(t) + Bx(t),$$
$$y(t) = Ch(t)$$

$$h(t+\Delta) = e^{A\Delta}h(t) + \int_0^{\Delta} e^{A(\Delta-\tau)}Bx(t+\tau)d\tau$$
$$x(t+\tau) = x_k$$

$$h(t+\Delta) = e^{A\Delta}h(t) + \left(\int_0^{\Delta} e^{A(\Delta-\tau)}d\tau\right)Bx_k$$

$$s = \Delta - \tau$$
$$\int_0^{\Delta} e^{A(\Delta-\tau)}d\tau = \int_0^{\Delta} e^{As}ds$$
$$\int_0^{\Delta} e^{As}ds = A^{-1}(e^{A\Delta} - I)$$

$$h_{k+1} = e^{A\Delta}h_k + A^{-1}(e^{A\Delta} - I)Bx_k$$

$$\bar{A} = e^{A\Delta}$$
$$\bar{B} = A^{-1}(e^{A\Delta} - I)B$$
$$\bar{B} = (\Delta A)^{-1}(e^{\Delta A} - I)\Delta B$$

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

RNN vs SSM

RNN

$$h_t = f(W_X X_t + W_h h_{t-1} + b_h)$$
$$y_t = f(W_y h_t + b_y)$$

✓ 비선형성

✓ 학습 기반

Discretized SSM

$$h_t = \overline{A} h_{t-1} + \overline{B} x_t$$
$$y_t = C h_t$$

✓ 선형성

✓ 수학적 기반

LTI System

$$
\begin{aligned}
h_0 &= \bar{B} \cdot x_0 \\
y_0 &= C \cdot h_0
\end{aligned}
$$

$$
\begin{aligned}
h_1 &= \bar{A} \cdot h_0 + \bar{B} \cdot x_1 \\
&= \bar{A} \cdot \bar{B} \cdot x_0 + \bar{B} \cdot x_1 \\
y_1 &= C \cdot h_1 \\
&= C \cdot \bar{A} \cdot \bar{B} \cdot x_0 + C \cdot \bar{B} \cdot x_1
\end{aligned}
$$

$$
\begin{aligned}
h_2 &= \bar{A} \cdot h_1 + \bar{B} \cdot x_2 \\
&= \bar{A} \cdot \left(\bar{A} \cdot \bar{B} \cdot x_0 + \bar{B} \cdot x_1 \right) + \bar{B} \cdot x_2 \\
&= \bar{A}^2 \cdot \bar{B} \cdot x_0 + \bar{A} \cdot \bar{B} \cdot x_1 + \bar{B} \cdot x_2 \\
y_2 &= C \cdot h_2 \\
&= C \cdot \bar{A}^2 \cdot \bar{B} \cdot x_0 + C \cdot \bar{A} \cdot \bar{B} \cdot x_1 + C \cdot \bar{B} \cdot x_2 \\
&\vdots
\end{aligned}
$$

$$
\overline{K} = \left(C\overline{B}, C\overline{AB}, \ldots, C\overline{A}^{k}\overline{B}, \ldots \right)
$$

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

Train mode

→global convolution

→parallel learning

Inference mode

→recurrent form

→Linear-time processing

HiPPO (high-order polynomial projection operater)

- ✓ RNN의 한계:
  - 긴 시퀀스(long sequences)를 효과적으로 처리하지 못함
  - 과거 정보를 잊어버림 (gradient vanishing)

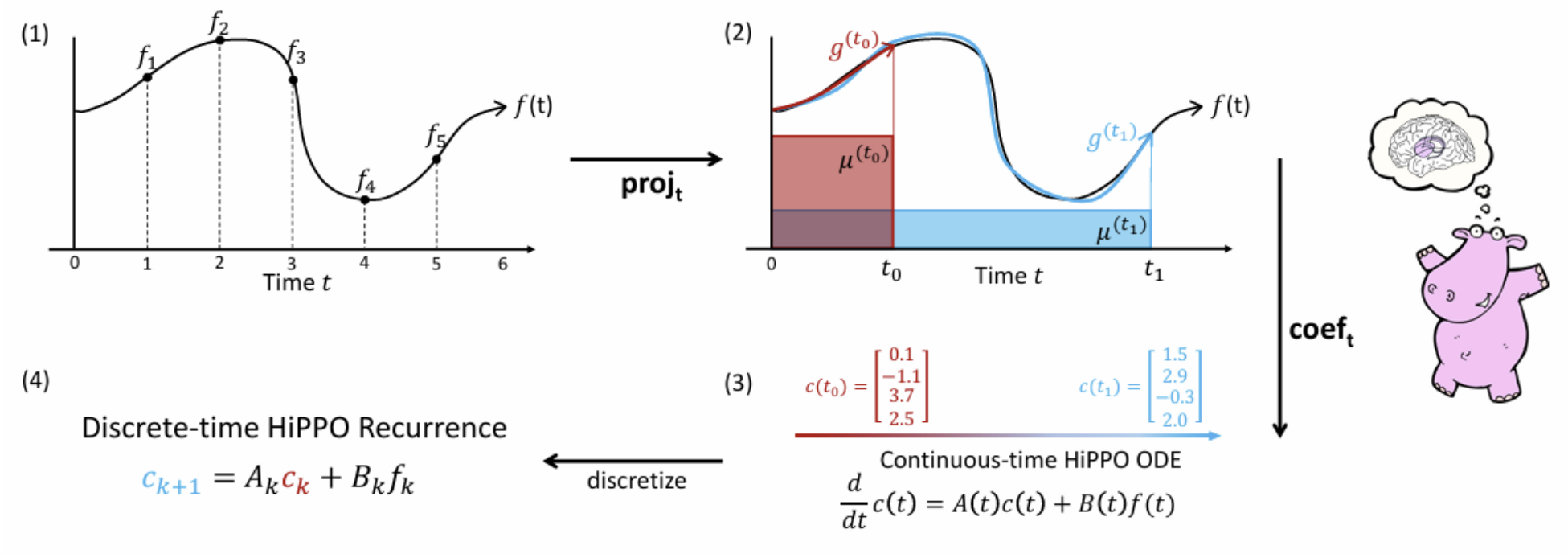
- ✓ Problem:
  - 누적된 과거의 모든 정보를 유한한 상태로 요약할 수 있어야 한다.

- ✓ Objective
  - 임의의 함수 f를 메모리 상태 $c(t)$에 저장하는 구조를 설계한다.

HiPPO (high-order polynomial projection operater)

✓ Solution:
- n차 이하의 유한한 다항함수들로 구성된 함수 g로 근사한다.

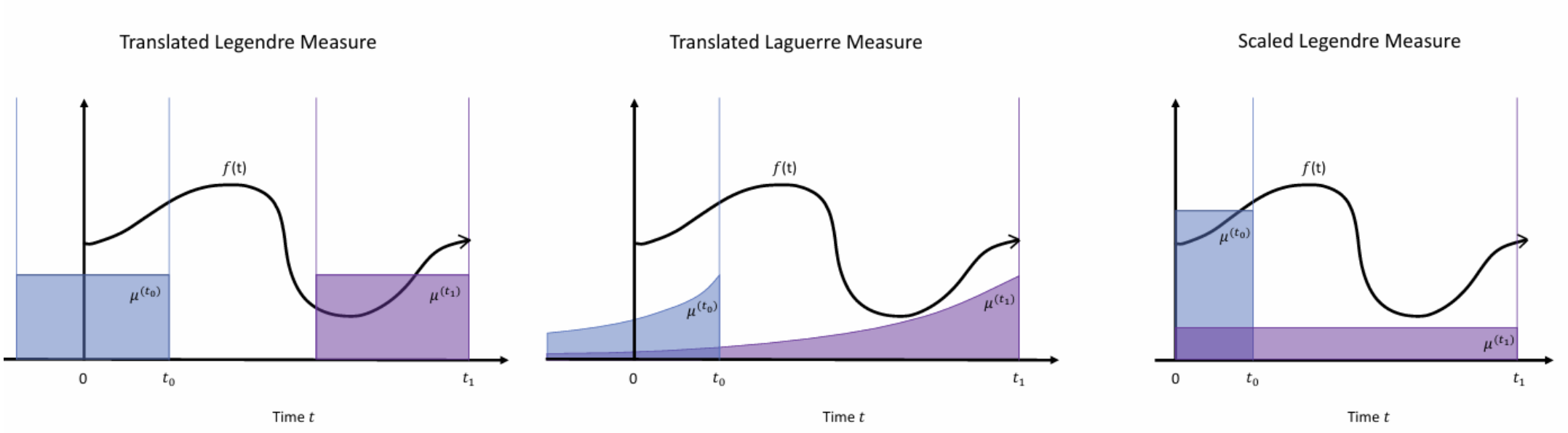
✓ Detail:
- 가중함수 $w(t)$와 다항식 기반 함수 g가 주어졌을 때, 함수 f를 g에 투영하는 최적계수 $c(t)$를 도출
- 이때 $c_k(t)$는 k번째 다항식 g_k의 계수로서, f를 g로 근사하는 데 최적인 값



HiPPO (high-order polynomial projection operater)

✓ Solution:
- n차 이하의 유한한 다항함수들로 구성된 함수 g로 근사한다.

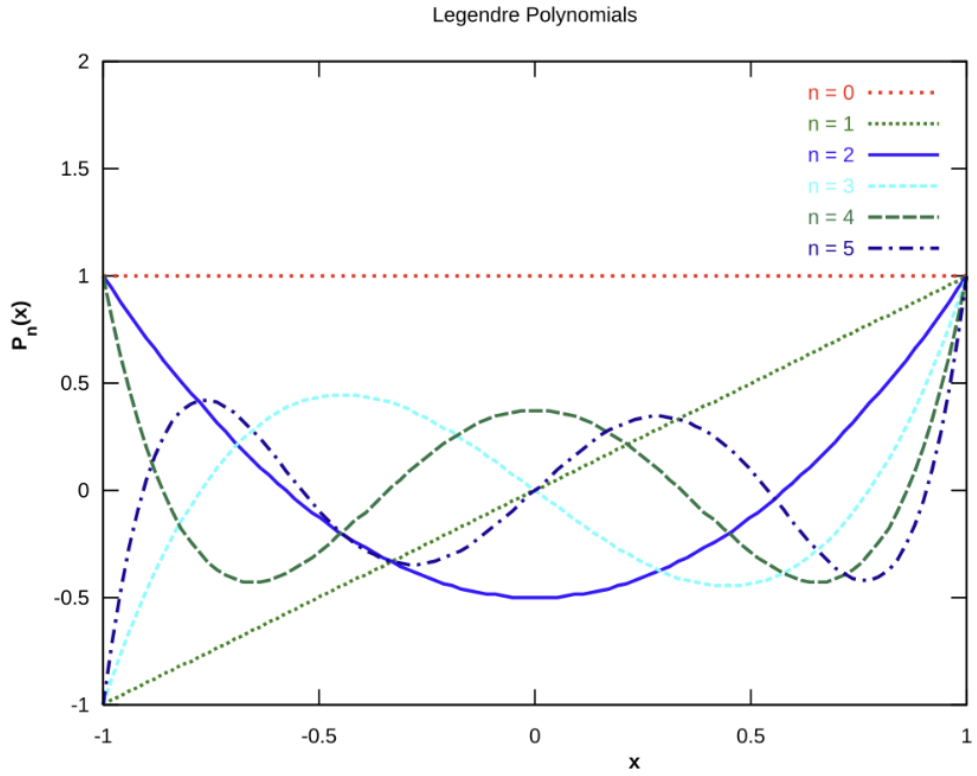
✓ Detail:
- 가중함수 $w(t)$와 다항식 기반 함수 g가 주어졌을 때, 함수 f를 g에 투영하는 최적계수 $c(t)$를 도출
- 이때 $c_k(t)$는 k번째 다항식 g_k의 계수로서, f를 g로 근사하는 데 최적인 값



HiPPO (high-order polynomial projection operater)

✓ Solution:
- n차 이하의 유한한 다항함수들로 구성된 함수 g로 근사한다.

✓ Detail:
- 가중함수 $w(t)$와 다항식 기반 함수 g가 주어졌을 때, 함수 f를 g에 투영하는 최적계수 $c(t)$를 도출
- 이때 $c_k(t)$는 k번째 다항식 g_k의 계수로서, f를 g로 근사하는 데 최적인 값



$$(P_n, P_m) = \int_{-1}^{1} P_m(x) P_n(x)\, dx = \frac{2}{2n+1}\delta_{mn}$$

HiPPO (high-order polynomial projection operater)

✓ Solution:
- n차 이하의 유한한 다항함수들로 구성된 함수 g로 근사한다.

✓ Detail:
- 가중함수 $w(t)$와 다항식 기반 함수 g가 주어졌을 때, 함수 f를 g에 투영하는 최적계수 $c(t)$를 도출
- 이때 $c_k(t)$는 k번째 다항식 g_k의 계수로서, f를 g로 근사하는 데 최적인 값

$$\| f_{x \leq t} - g^{(t)} \|_{L_2(\mu^{(t)})}$$

$$c_n(t) = \langle f_{\leq t}, g_n^{(t)} \rangle_{\nu^{(t)}}$$

→

$$\frac{d}{dt} c(t) = -A(t)c(t) + B(t)f(t).$$

HiPPO (high-order polynomial projection operater)

✓ Solution:
- n차 이하의 유한한 다항함수들로 구성된 함수 g로 근사한다.

✓ Detail:
- 가중함수 $w(t)$와 다항식 기반 함수 g가 주어졌을 때, 함수 f를 g에 투영하는 최적계수 $c(t)$를 도출
- 이때 $c_k(t)$는 k번째 다항식 g_k의 계수로서, f를 g로 근사하는 데 최적인 값

$$\frac{d}{dt}c(t) = -\frac{1}{t}Ac(t) + \frac{1}{t}Bf(t)$$

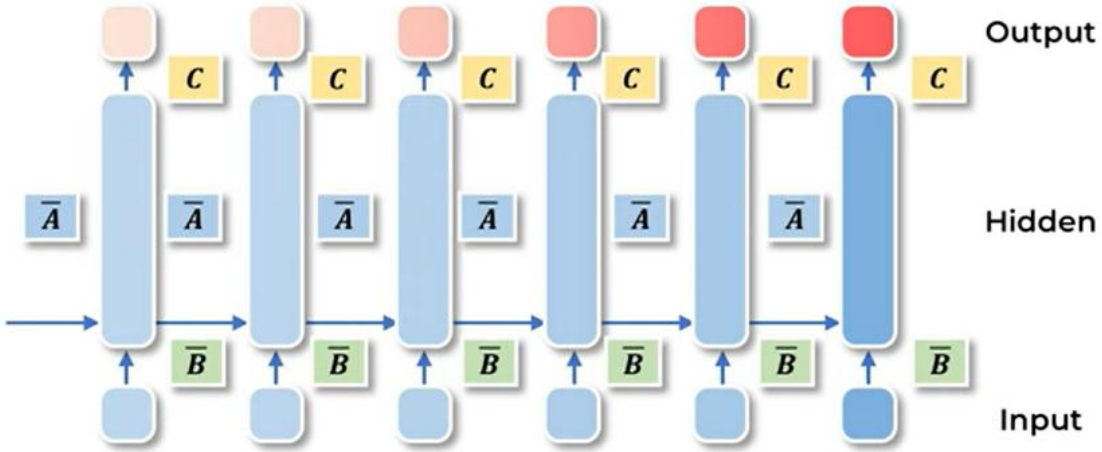
$$A_{nk} = \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k \\ n+1 & \text{if } n = k \\ 0 & \text{if } n < k \end{cases},$$

$$B_n = (2n+1)^{\frac{1}{2}}$$

Now, Train SSM with HiPPO matrix!

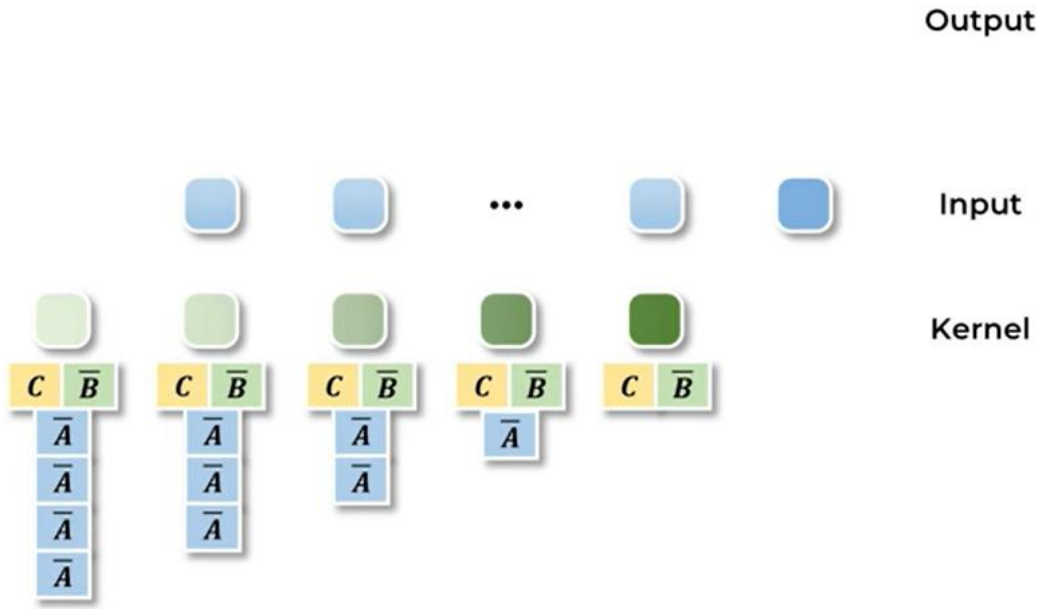
S4 (Structured State Space for Sequence Modeling)

[Recurrent]



$$\mathbf{A} \in \mathbb{R}^{n \times n}$$
$$\mathbf{B} \in \mathbb{R}^{n \times 1}$$
$$\mathbf{C} \in \mathbb{R}^{1 \times n}$$

[Convolution]

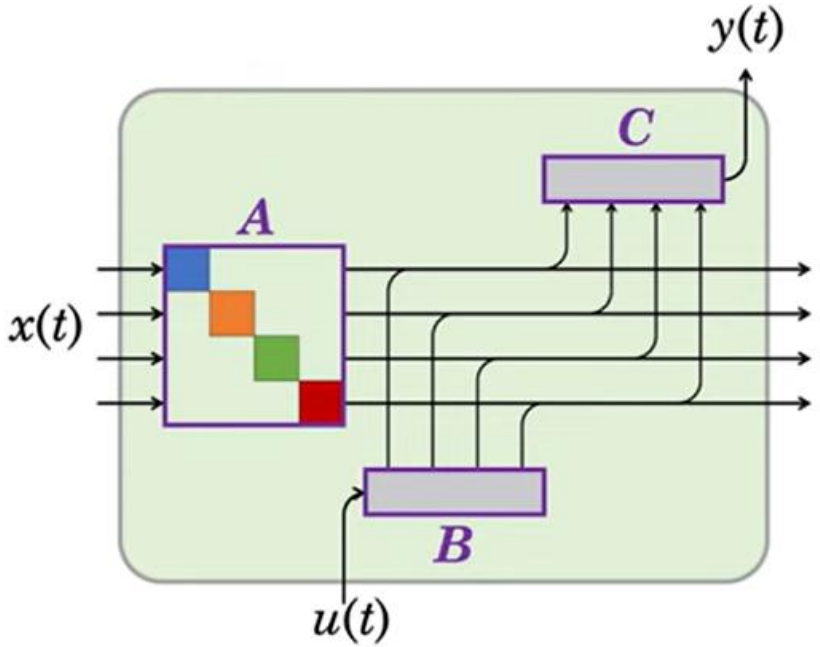


S4 (Structured State Space for Sequence Modeling)

- ✓ Problem:
  - Training 단계에서 convolution 연산이 실질적으로 어려움

- ✓ Solution: Structured State Space Sequence Model
  - HiPPO 행렬 A에 대해 완전한 Diagonalization을 만들어 냄

A를 안정적으로 대각화하여 연산량을 대폭 낮춤
Deep SSM setting에서도 실질적으로 훈련이 feasible & stable해짐

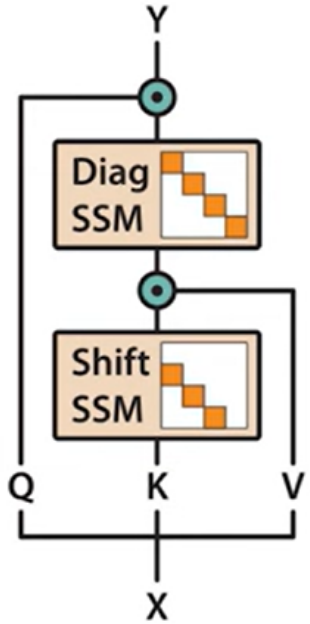


S4D Recurrent View

H3 (Hungry Hungry HiPPO)

- ✓ S4 이후, SSM을 더 고도화 하고자 한 논문
  - 특히 Language Modeling에서의 Structured SSM 성능을 올리고자 함
- ✓ 이를 위해, Language Modeling을 잘 하는 Transformer에 비해, Structured SSM이 무엇을 못하는지 연구
  - 이후 Mamba도 이와 비슷한 approach를 취함
- ✓ 결과적으로, 단순히 SSM을 쌓는 것이 아니라, 하나의 block 단위에 좀 더 Complex한 연산이 들어감

Former SSMs



H3 Layer

H3 (Hungry Hungry HiPPO)

Task	Input	Output
Induction Head	$a\ b\ c\ d\ e \vdash f\ g\ h\ i \ldots x\ y\ z \vdash$	f
Associative Recall	$a\ 2\ c\ 4\ b\ 3\ d\ 1\ a$	2

정해진 Special token 뒤에
어떤 글자가 왔었는지
기억하고 생성해내는 Task

여러 Key-Value 쌍을 기억해서
주어진 Key에 대한 Value를
생성해내는 Task

Task	Random	S4D	Gated State Spaces	H3	Attention
Induction Head	5.0	35.6	6.8	100.0	100.0
Associative Recall	25.0	86.0	78.0	99.8	100.0

어떤 Capability가 없어서 못 하는 걸까?

→ 어떤 Capability를 심어줘야 할까?

H3 (Hungry Hungry HiPPO)

Task	Input	Output
Induction Head	$a\ b\ c\ d\ e \vdash f\ g\ h\ i \ldots x\ y\ z \vdash$	f
Associative Recall	$a\ 2\ c\ 4\ b\ 3\ d\ 1\ a$	2

정해진 Special token 뒤에
어떤 글자가 왔었는지
기억하고 생성해내는 Task

여러 Key-Value 쌍을 기억해서
주어진 Key에 대한 Value를
생성해내는 Task

Task	Random	S4D	Gated State Spaces	H3	Attention
Induction Head	5.0	35.6	6.8	100.0	100.0
Associative Recall	25.0	86.0	78.0	99.8	100.0

(1)

특정 Input 뒤에 등장한 Input을
기억하는 능력

(2)

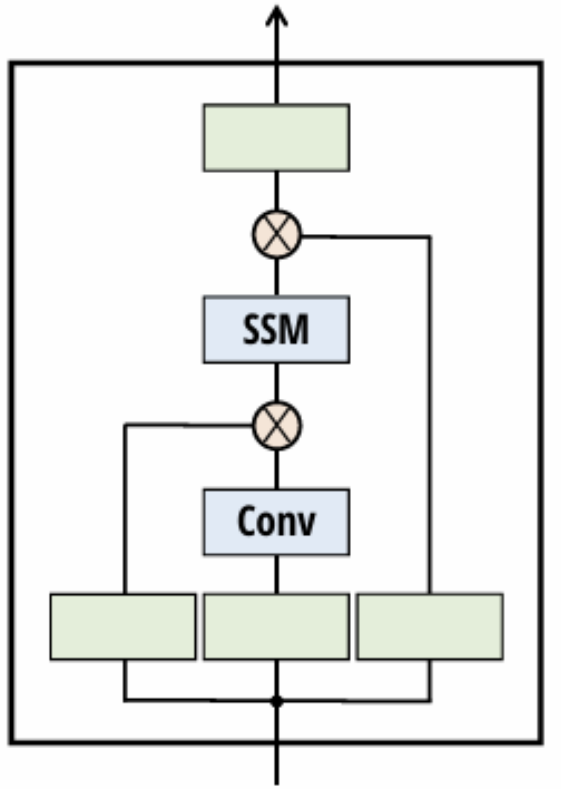
지금 시점에 Input된 정보를
지금 시점의 Memory와 비교하는 능력

H3 (Hungry Hungry HiPPO)

- ✓ 기존의 SSM도 Global memory가 있기 때문에, 그런 능력이 있음
  - 실험적으로 잘 못하는 것을 확인
  - 그런 능력을 추가하여 Transformer와의 간극을 줄임

- ✓ 특히, (1)을 input에 대해 Sequence 방향으로의 Conv1D Layer를 추가하는 것으로 쉽게 해결

$$\mathbf{Q} \odot \mathrm{SSM}_{\mathrm{diag}}\left(\mathrm{SSM}_{\mathrm{shift}}(\mathbf{K}) \odot \mathbf{V}\right)$$



Mamba

Mamba

✓ Transformer의 High Performance와 Domain-agnostic한 Robustness는 유지하면서 Sub-quadratic한 architecture에 대한 연구가 지속되어왔지만, 만족할만한 성능은 내지 못했음

SSSSSS...

✓ Mamba는 그 갈래 중 하나인 deep structured SSM의 철학을 이어받는 연구

- 이쪽 계열의 기존 장점인 Fast Inference는 물론,
- High-performance를 달성
- 특히, Language Modelling에도 정말 Transformer와 비견될 정도의 성능을 보여 줌

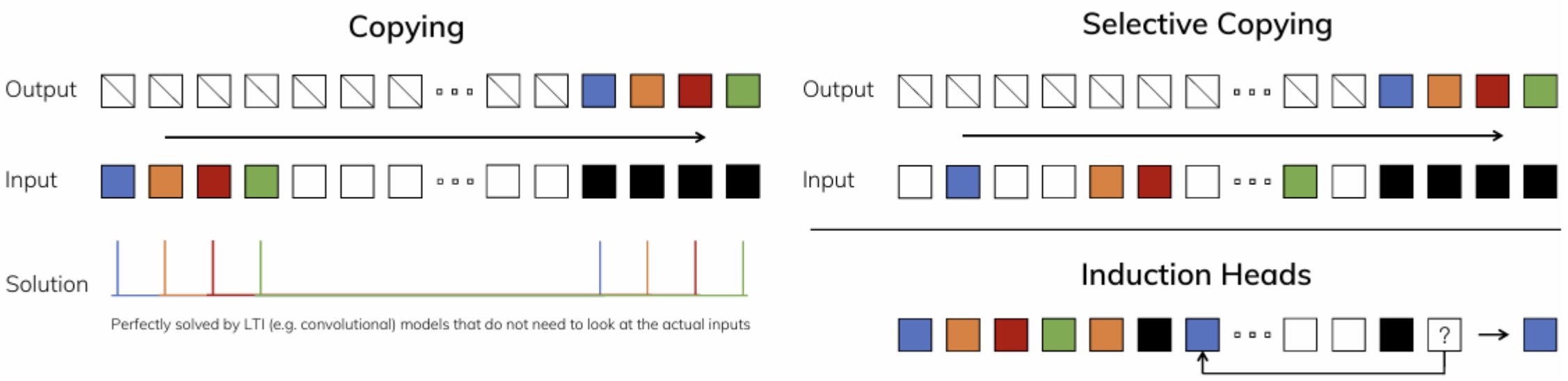
✓ 이를 위해,

- SSM의 성능 문제는 Selectivity를 도입하여 해결
- 이로 인한 연산 속도의 문제는 Hardware-aware Parallel Scan으로 해결

What's matter with SSMS

- ✓ SSMs는 RNN과 CNN의 Interchangeable한 Combination
  - 그로서 LRA, Audio, Vision등, 특히 Continuous한 Domain에서는 Dominant한 성능을 보여줌
  - 하지만, Text와 같은 info-dense하고 discrete한 분야에서는 약함

- ✓ 이와 관련해서, H3와 비슷한 Approach를 취함
  - Selective Copying



특정 입력들만 Hidden Memory에 싣는 것이 가능해야 함

Selection

Algorithm 1 SSM (S4)

Input: $x : (\mathsf{B}, \mathsf{L}, \mathsf{D})$
Output: $y : (\mathsf{B}, \mathsf{L}, \mathsf{D})$

$\boldsymbol{A} : (\mathsf{D}, \mathsf{N}) \leftarrow$ Parameter
▷ Represents structured $N \times N$ matrix
$\boldsymbol{B} : (\mathsf{D}, \mathsf{N}) \leftarrow$ Parameter
$\boldsymbol{C} : (\mathsf{D}, \mathsf{N}) \leftarrow$ Parameter
$\Delta : (\mathsf{D}) \leftarrow \tau_\Delta(\text{Parameter})$
$\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathsf{D}, \mathsf{N}) \leftarrow \text{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$
$y \leftarrow \text{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$
▷ Time-invariant: recurrence or convolution
return y

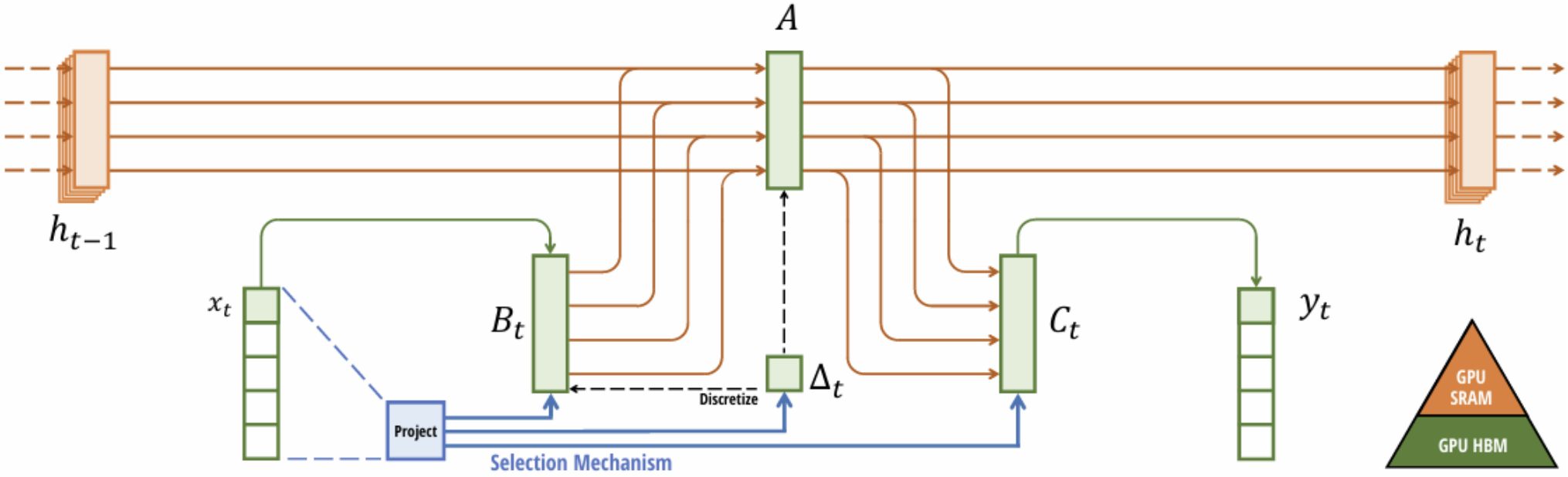
Algorithm 2 SSM + Selection (S6)

Input: $x : (\mathsf{B}, \mathsf{L}, \mathsf{D})$
Output: $y : (\mathsf{B}, \mathsf{L}, \mathsf{D})$

$\boldsymbol{A} : (\mathsf{D}, \mathsf{N}) \leftarrow$ Parameter
▷ Represents structured $N \times N$ matrix
$\boldsymbol{B} : (\mathsf{B}, \mathsf{L}, \mathsf{N}) \leftarrow s_B(x)$
$\boldsymbol{C} : (\mathsf{B}, \mathsf{L}, \mathsf{N}) \leftarrow s_C(x)$
$\Delta : (\mathsf{B}, \mathsf{L}, \mathsf{D}) \leftarrow \tau_\Delta(\text{Parameter} + s_\Delta(x))$
$\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathsf{B}, \mathsf{L}, \mathsf{D}, \mathsf{N}) \leftarrow \text{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$
$y \leftarrow \text{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$
▷ Time-varying: recurrence (*scan*) only
return y

$$
\begin{aligned}
s_B(x) &= \text{Linear}_N(x) \\
s_C(x) &= \text{Linear}_N(x) \\
s_\Delta(x) &= \text{Broadcast}_D(\text{Linear}_1(x))
\end{aligned}
$$

Selection



Algorithm 2 SSM + Selection (S6)

Input: $x : (\mathsf{B}, \mathsf{L}, \mathsf{D})$
Output: $y : (\mathsf{B}, \mathsf{L}, \mathsf{D})$

$\boldsymbol{A} : (\mathsf{D}, \mathsf{N}) \leftarrow$ Parameter
▷ Represents structured $N \times N$ matrix
$\boldsymbol{B} : (\mathsf{B}, \mathsf{L}, \mathsf{N}) \leftarrow s_B(x)$
$\boldsymbol{C} : (\mathsf{B}, \mathsf{L}, \mathsf{N}) \leftarrow s_C(x)$
$\Delta : (\mathsf{B}, \mathsf{L}, \mathsf{D}) \leftarrow \tau_\Delta(\text{Parameter} + s_\Delta(x))$
$\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathsf{B}, \mathsf{L}, \mathsf{D}, \mathsf{N}) \leftarrow \text{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$
$y \leftarrow \text{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$
▷ Time-varying: recurrence (*scan*) only
return y

Selection

ZOH Discretization

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

$\Delta \to 0$:
현재 Hidden State 유지
현재 Input 무시

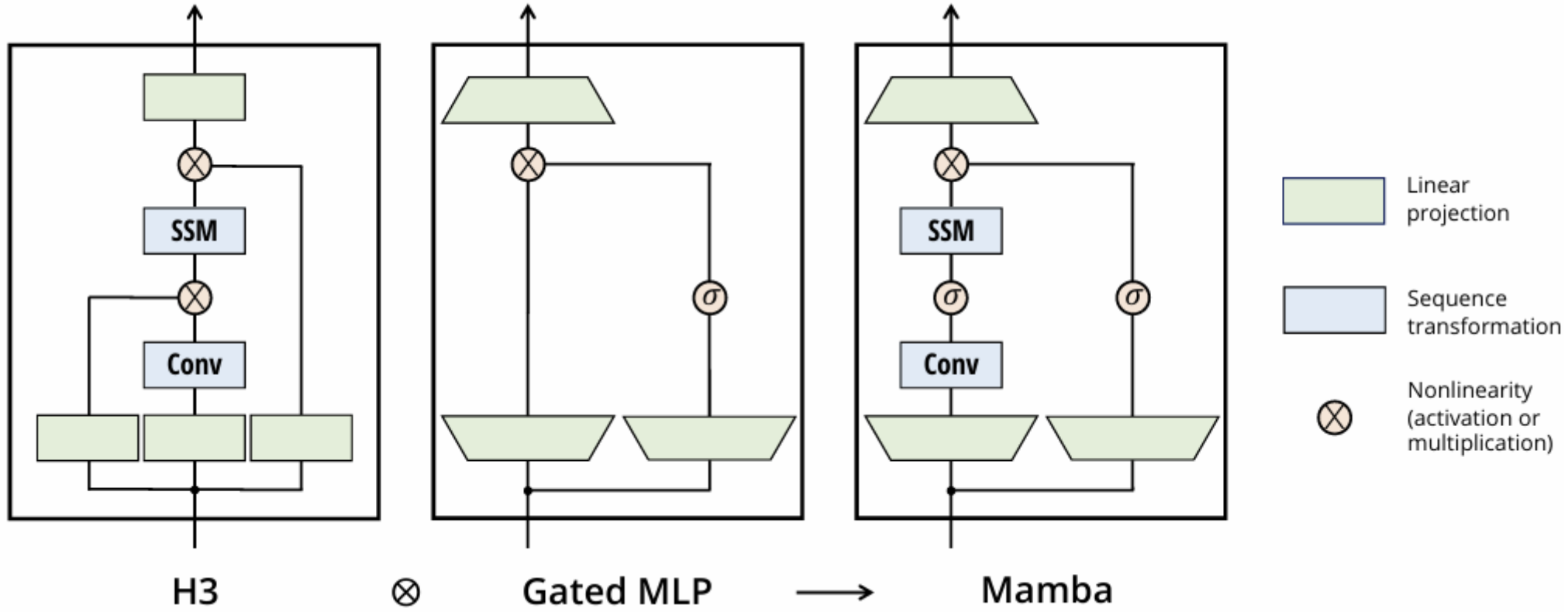
$\Delta \to \infty$:
현재 Input에 대한
영향 증가

Algorithm 2 SSM + Selection (S6)

Input: $x : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
Output: $y : (\mathtt{B}, \mathtt{L}, \mathtt{D})$

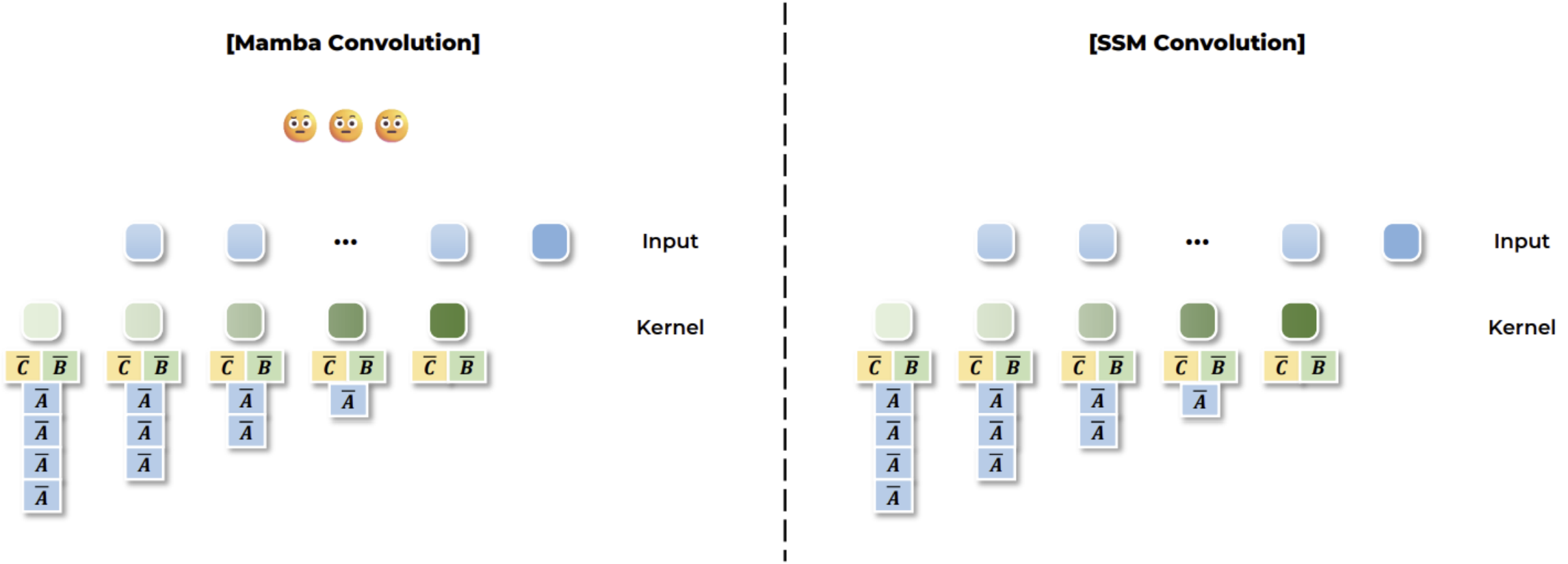
$\boldsymbol{A} : (\mathtt{D}, \mathtt{N}) \leftarrow$ Parameter
▷ Represents structured $N \times N$ matrix
$\boldsymbol{B} : (\mathtt{B}, \mathtt{L}, \mathtt{N}) \leftarrow s_B(x)$
$\boldsymbol{C} : (\mathtt{B}, \mathtt{L}, \mathtt{N}) \leftarrow s_C(x)$
$\Delta : (\mathtt{B}, \mathtt{L}, \mathtt{D}) \leftarrow \tau_\Delta(\text{Parameter} + s_\Delta(x))$ $\tau_\Delta = \text{softplus}$
$\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathtt{B}, \mathtt{L}, \mathtt{D}, \mathtt{N}) \leftarrow \text{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$
$y \leftarrow \text{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$
▷ Time-varying: recurrence (*scan*) only
return y

Mamba


SSM
Conv
SSM
Conv
H3
Gated MLP
Mamba
Linear projection
Sequence transformation
Nonlinearity (activation or multiplication)

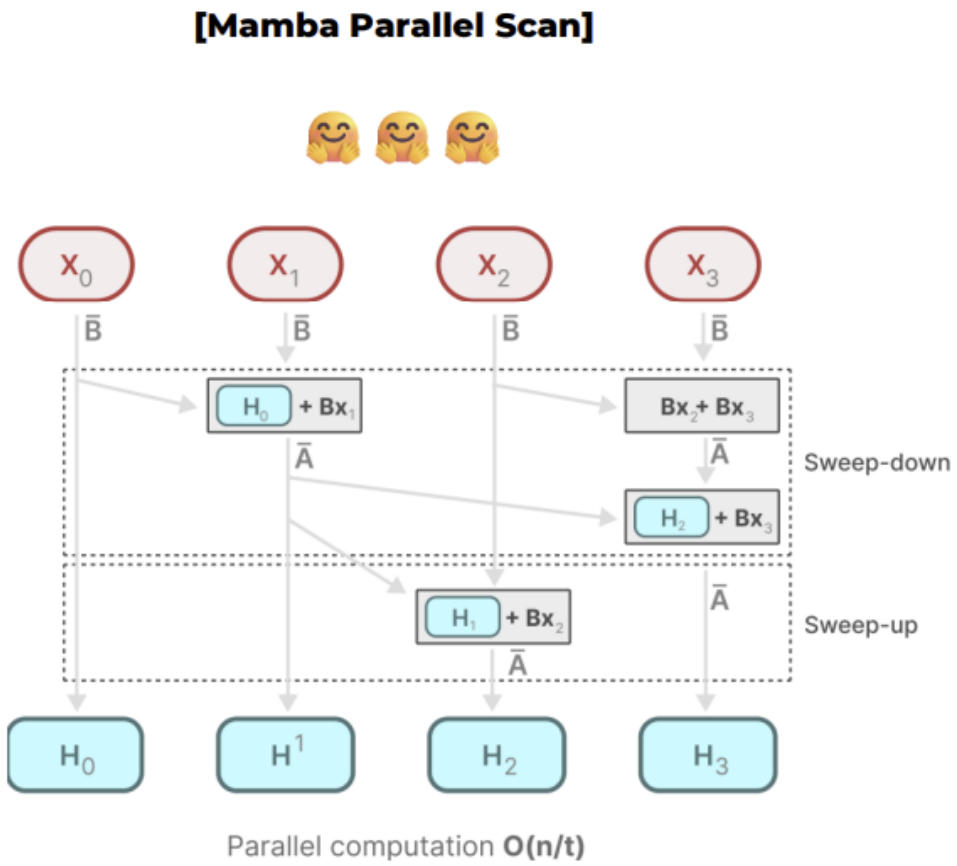
Training

✓ Convolution 연산 불가능
- Train Parallelism? -> Parallel scan!



Training

- ✓ Convolution 연산 불가능
  - Train Parallelism? -> Parallel associative scan!



Scan

- ✓ Convolution 연산 불가능
  - Train Parallelism? -> Parallel scan!

Looks Like $O(n)$...
Cannot Compute H_t Before H_{t-1}

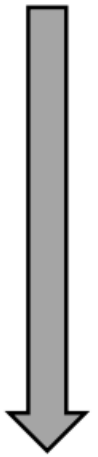
$$
\begin{aligned}
\mathbf{H}_0 &= \overline{\mathbf{B}}_0\mathbf{x}_0 \\
\mathbf{H}_1 &= \overline{\mathbf{A}}_1(\overline{\mathbf{B}}_0\mathbf{x}_0) + \overline{\mathbf{B}}_1\mathbf{x}_1 \\
&= \overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{B}}_1\mathbf{x}_1 \\
\mathbf{H}_2 &= \overline{\mathbf{A}}_2(\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{B}}_1\mathbf{x}_1) + \overline{\mathbf{B}}_2\mathbf{x}_2 \\
&= \overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{B}}_2\mathbf{x}_2 \\
\mathbf{H}_3 &= \overline{\mathbf{A}}_3(\overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{B}}_2\mathbf{x}_2) + \overline{\mathbf{B}}_3\mathbf{x}_3 \\
&= \overline{\mathbf{A}}_3\overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_3\overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{A}}_3\overline{\mathbf{B}}_2\mathbf{x}_2 + \overline{\mathbf{B}}_3\mathbf{x}_3
\end{aligned}
$$

Scan

- ✓ Convolution 연산 불가능
  - Train Parallelism? -> Parallel scan!

$$
\begin{aligned}
\mathbf{H}_0 &= \overline{\mathbf{B}}_0\mathbf{x}_0 \\
\mathbf{H}_1 &= \overline{\mathbf{A}}_1(\overline{\mathbf{B}}_0\mathbf{x}_0) + \overline{\mathbf{B}}_1\mathbf{x}_1 \\
&= \overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{B}}_1\mathbf{x}_1 \\
\mathbf{H}_2 &= \overline{\mathbf{A}}_2(\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{B}}_1\mathbf{x}_1) + \overline{\mathbf{B}}_2\mathbf{x}_2 \\
&= \overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{B}}_2\mathbf{x}_2 \\
\mathbf{H}_3 &= \overline{\mathbf{A}}_3(\overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{B}}_2\mathbf{x}_2) + \overline{\mathbf{B}}_3\mathbf{x}_3 \\
&= \overline{\mathbf{A}}_3\overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_3\overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{A}}_3\overline{\mathbf{B}}_2\mathbf{x}_2 + \overline{\mathbf{B}}_3\mathbf{x}_3
\end{aligned}
$$

Compute Matrices Multiplication with Multiple Threads



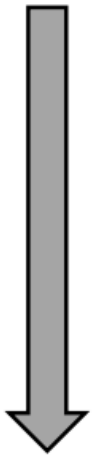
Complexity Drops to $O(n/T)$

Scan

- ✓ Convolution 연산 불가능
  - Train Parallelism? -> Parallel scan!

$$
\begin{aligned}
\mathbf{H}_0 &= \overline{\mathbf{B}}_0\mathbf{x}_0 \\
\mathbf{H}_1 &= \overline{\mathbf{A}}_1(\overline{\mathbf{B}}_0\mathbf{x}_0) + \overline{\mathbf{B}}_1\mathbf{x}_1 \\
&= \overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{B}}_1\mathbf{x}_1 \\
\mathbf{H}_2 &= \overline{\mathbf{A}}_2(\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{B}}_1\mathbf{x}_1) + \overline{\mathbf{B}}_2\mathbf{x}_2 \\
&= \overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{B}}_2\mathbf{x}_2 \\
\mathbf{H}_3 &= \overline{\mathbf{A}}_3(\overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{B}}_2\mathbf{x}_2) + \overline{\mathbf{B}}_3\mathbf{x}_3 \\
&= \overline{\mathbf{A}}_3\overline{\mathbf{A}}_2\overline{\mathbf{A}}_1\overline{\mathbf{B}}_0\mathbf{x}_0 + \overline{\mathbf{A}}_3\overline{\mathbf{A}}_2\overline{\mathbf{B}}_1\mathbf{x}_1 + \overline{\mathbf{A}}_3\overline{\mathbf{B}}_2\mathbf{x}_2 + \overline{\mathbf{B}}_3\mathbf{x}_3
\end{aligned}
$$

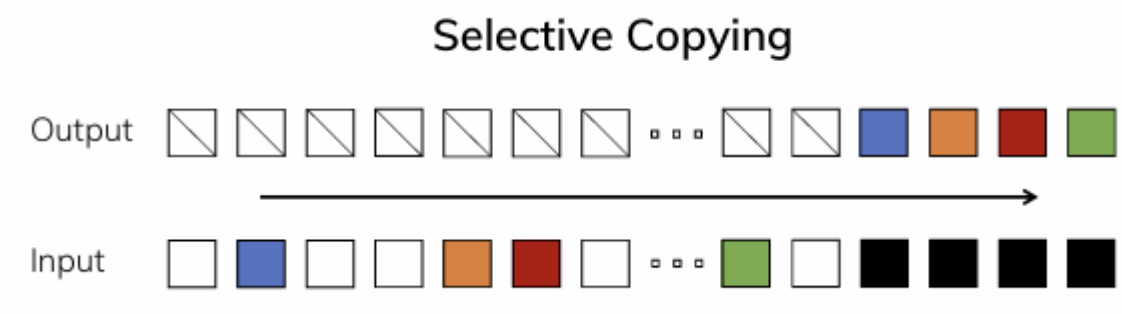
Compute Matrices Multiplication with Multiple Threads



Complexity Drops to $O(n/T)$

Evaluation

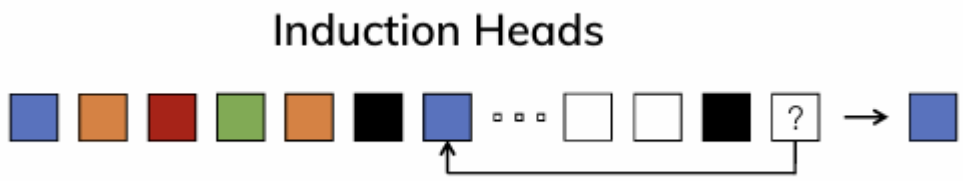
Selective Copying



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.

Induction Heads



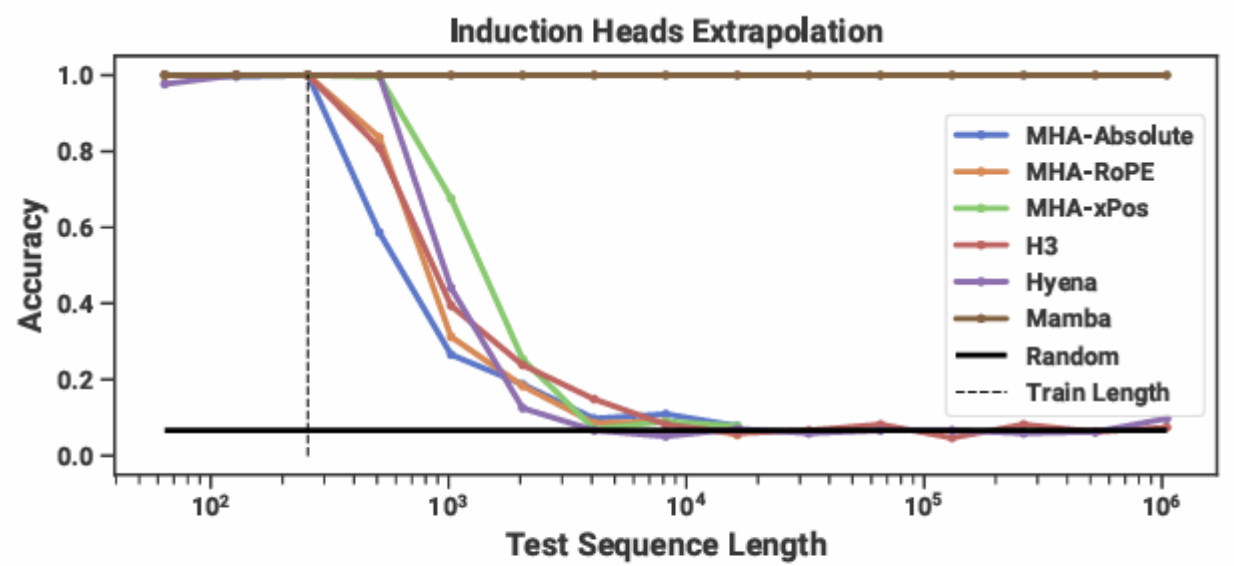


Table 2: (**Induction Heads**.) Models are trained on sequence length $2^8 = 256$, and tested on increasing sequence lengths of $2^6 = 64$ up to $2^{20} = 1048576$. Full numbers in Table 11.

Scaling Law

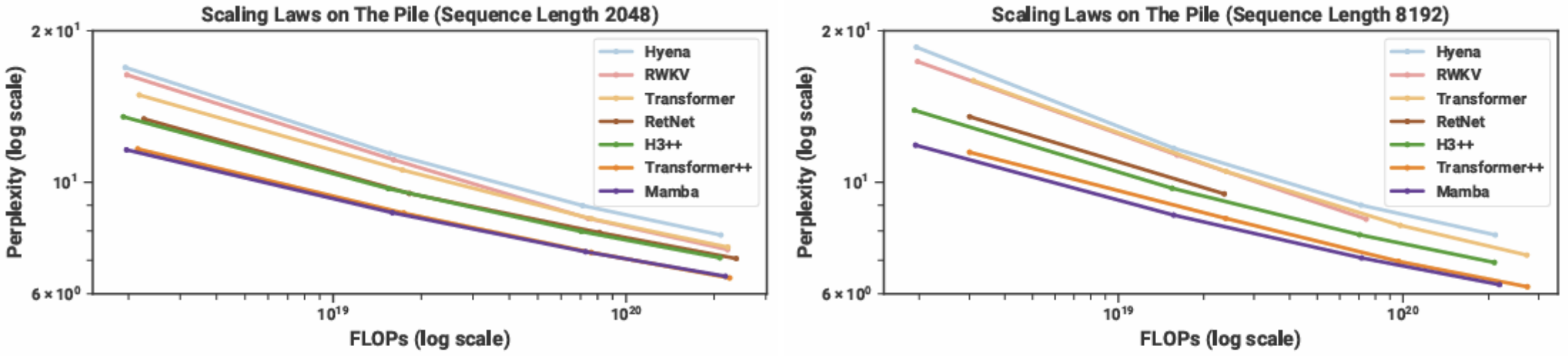


Figure 4: (**Scaling Laws.**) Models of size $\approx 125M$ to $\approx 1.3B$ parameters, trained on the Pile. Mamba scales better than all other attention-free models and is the first to match the performance of a very strong "Transformer++" recipe that has now become standard, particularly as the sequence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

Downstream

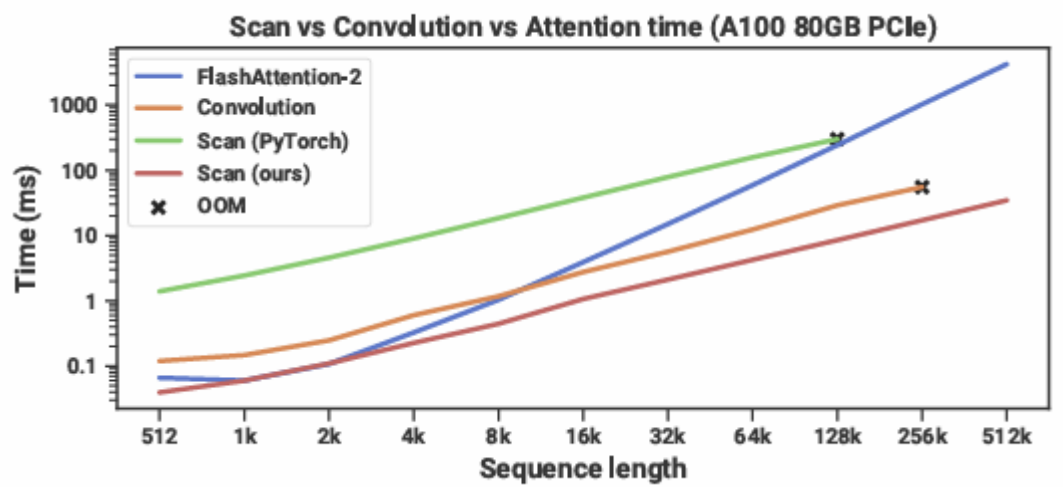
Model	Token.	Pile ppl ↓	LAMBADA ppl ↓	LAMBADA acc ↑	HellaSwag acc ↑	PIQA acc ↑	Arc-E acc ↑	Arc-C acc ↑	WinoGrande acc ↑	Average acc ↑
Hybrid H3-130M	GPT2	—	89.48	25.77	31.7	64.2	44.4	24.2	50.6	40.1
Pythia-160M	NeoX	29.64	38.10	33.0	30.2	61.4	43.2	24.1	**51.9**	40.6
Mamba-130M	NeoX	**10.56**	**16.07**	**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

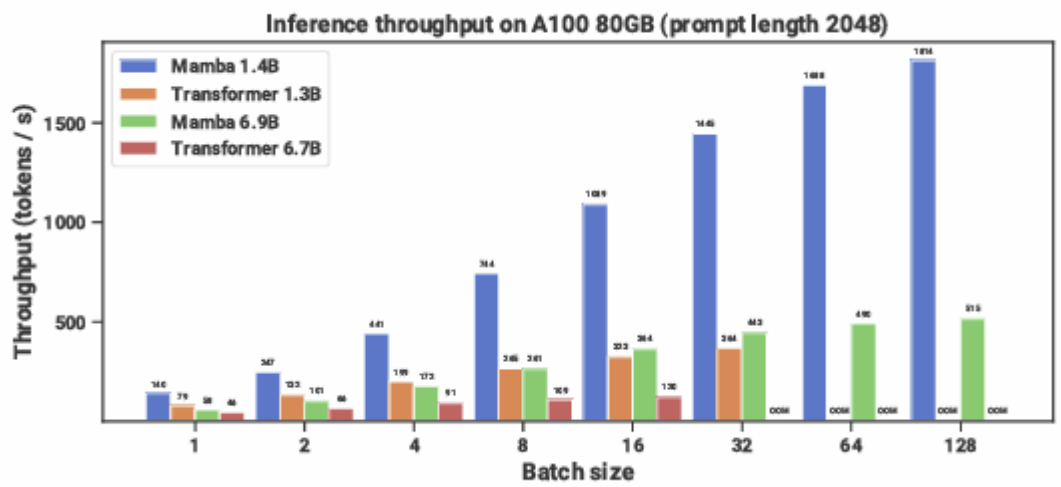
Audio Generation

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.

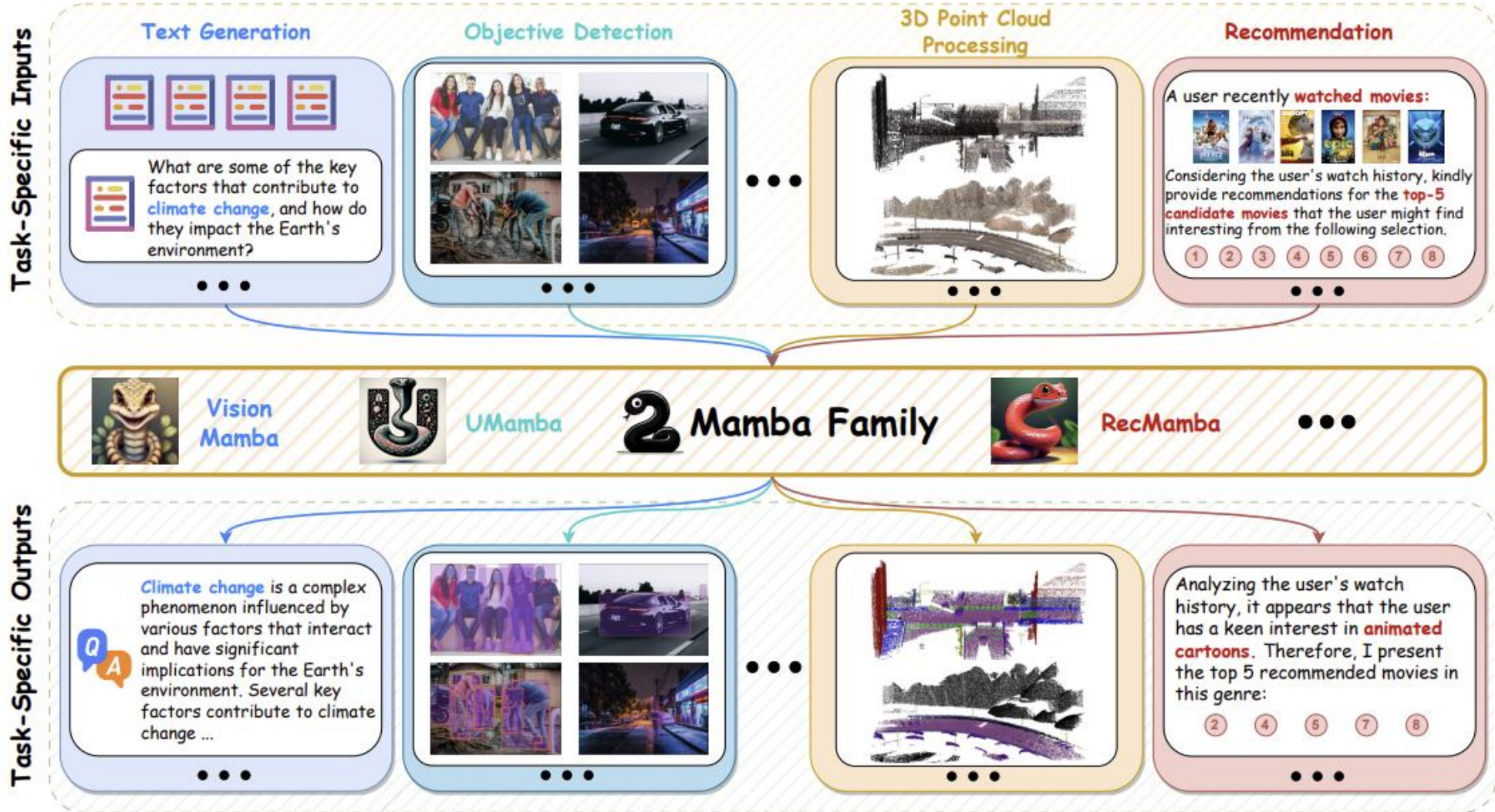
Model	Params	NLL ↓	FID ↓	IS ↑	mIS ↑	AM ↓
SampleRNN	35.0M	2.042	8.96	1.71	3.02	1.76
WaveNet	4.2M	1.925	5.08	2.27	5.80	1.47
SaShiMi	5.8M	1.873	1.99	5.13	42.57	0.74
WaveGAN	19.1M	-	2.03	4.90	36.10	0.80
DiffWave	24.1M	-	1.92	5.26	51.21	0.68
+ SaShiMi	23.0M	-	1.42	5.94	69.17	0.59
Mamba	6.1M	**1.852**	0.94	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

Speed


Scan vs Convolution vs Attention time (A100 80GB PCIe)
FlashAttention-2
Convolution
Scan (PyTorch)
Scan (ours)
OOM
Time (ms)
1000
100
10
1
0.1
512
1k
2k
4k
8k
16k
32k
64k
128k
256k
512k
Sequence length


Inference throughput on A100 80GB (prompt length 2048)
Mamba 1.4B
Transformer 1.3B
Mamba 6.9B
Transformer 6.7B
Throughput (tokens / s)
1500
1000
500
1
2
4
8
16
32
64
128
Batch size

Recent


Task-Specific Inputs
Text Generation
What are some of the key factors that contribute to climate change, and how do they impact the Earth's environment?
Objective Detection
3D Point Cloud Processing
Recommendation
A user recently watched movies:
Considering the user's watch history, kindly provide recommendations for the top-5 candidate movies that the user might find interesting from the following selection.
1 2 3 4 5 6 7 8
Vision Mamba
UMamba
Mamba Family
RecMamba
Task-Specific Outputs
Climate change is a complex phenomenon influenced by various factors that interact and have significant implications for the Earth's environment. Several key factors contribute to climate change ...
Analyzing the user's watch history, it appears that the user has a keen interest in animated cartoons. Therefore, I present the top 5 recommended movies in this genre:
2 4 5 7 8

Conclusion

Conclusion

✓ Mamba는 Transformer보다 낮은, Sub-quadratic한 Time-complexity를 가지는 Sequential Model

- structured SSM과 철학을 공유하며, 더 발전시킨 model

✓ 기존의 structured SSM이 본질적으로 'selectivity'라는 능력을 가질 수 없다는 것을 증명

- 이를 보완해줌으로서 structured SSM의 성능을 높일 수 있음을 주장하고 증명함
- Language에도 적용될 수 있을 정도로 Transformer급의 Performance를 보임

✓ 이 보완점을 위해 추가한 연산에 의해 SSM의 training parallelism의 핵심이었던 convolution 연산이 불가능해짐

- Parallel associative scan을 통해 해결, 결론적으로 삼각형의 세 변 모두를 잡음

감사합니다