北京工業大學

BEIJINGUNIVERSITYOFTECHNOLOGY

Mamba: Linear-Time Sequence Modeling with Selective State Spaces

汇报人：闫林枝

2024.05.12

CONTENT

P

ART ONE

背景

01

Transformer & RNN

Transformer

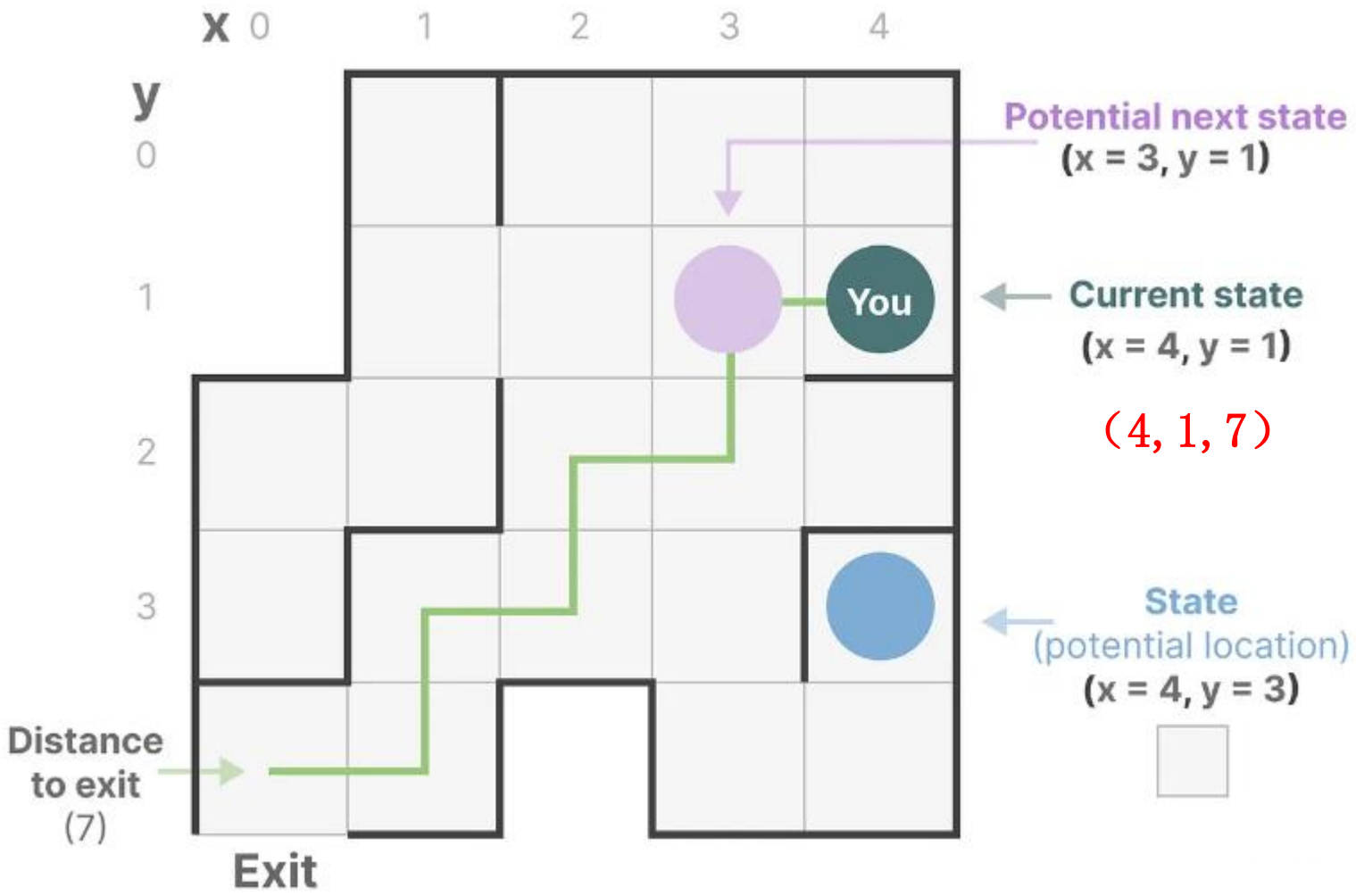
- 优点：可以看到之前所有的序列，而且在训练过程中可以并行化计算；
- 缺点：在推理过程中，每生成一个新的token，都需要重新为整个序列计算一个新的attention map，导致推理性能很慢。

RNN

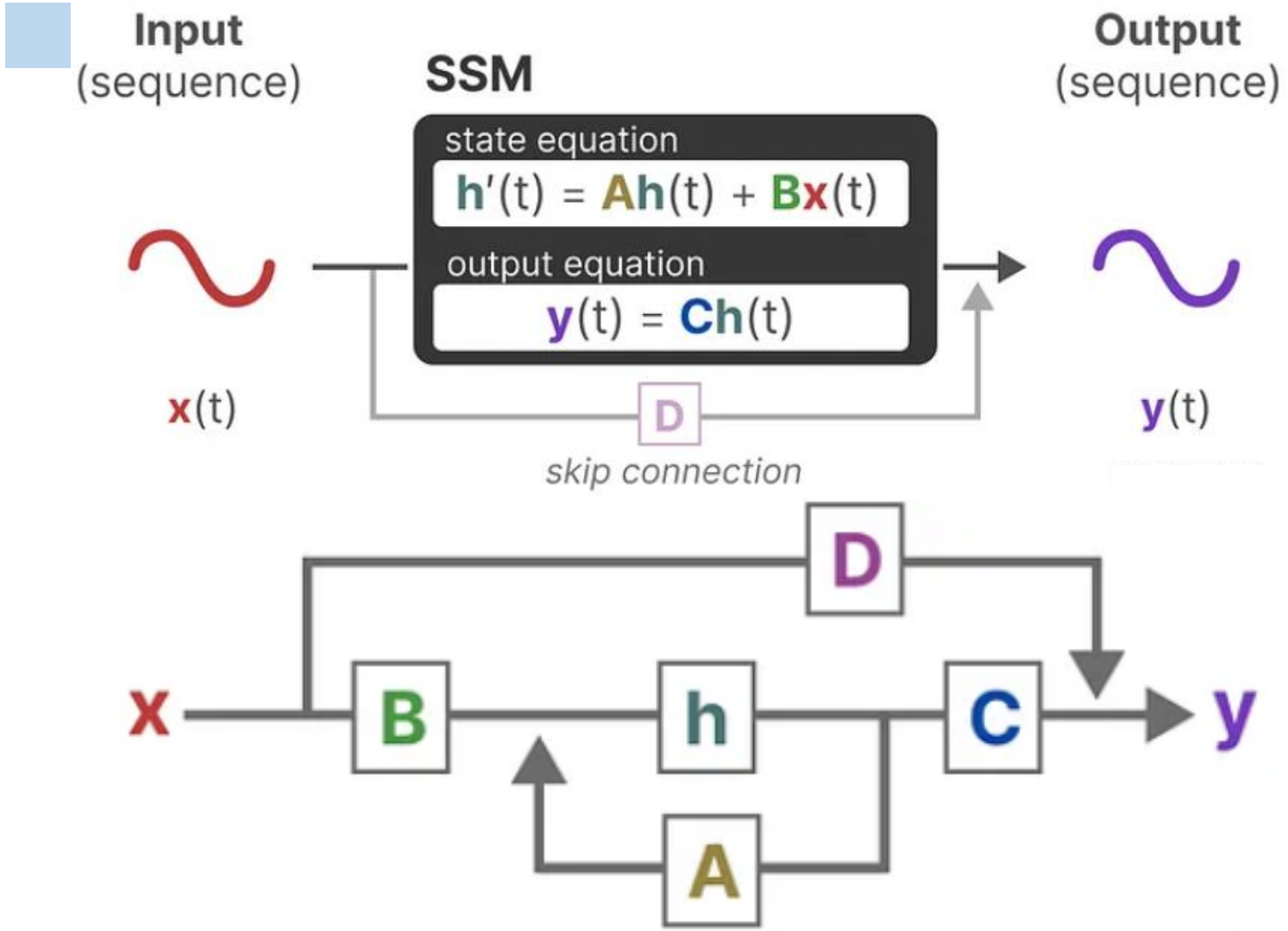
- 优点：在生成当前的输出时，RNN只需要考虑当前的输入和上一时刻的隐藏状态，和Transformer相比，它不需要重新计算先前所有的隐藏状态，可以快速进行推理；
- 缺点：训练不能并行计算，因为它需要按照时间顺序地完成每个步骤。

Mamba：是否能找到一种架构，既能像Transformer一样并行训练，又能执行与序列长度线性扩展的推理？

状态空间（State Space）



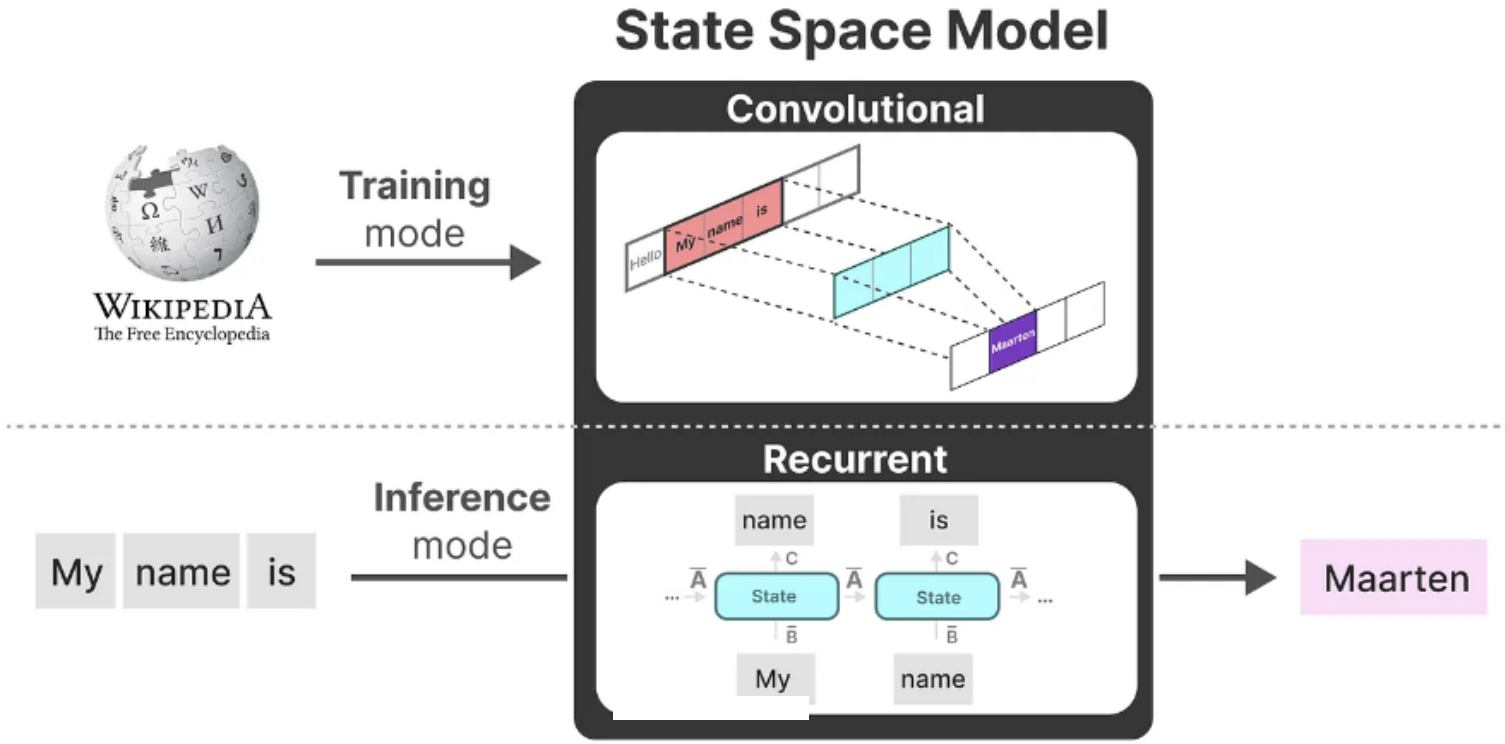
状态空间模型（SSM）



- Step1：输入信号x（t）与矩阵B相乘；
- Step2：矩阵A与当前状态相乘；
- Step3：矩阵C与新的状态相乘；

线性状态空间层（LSSL）

- **核心思想**：把连续时间的SSM离散化
- **方法**：ZOH（零阶保持器） $\overline{A} = \exp(\Delta A)$ $\overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B$
- **重要特性**：时间不变性（LTI）



状态空间模型-S4

通过隐含的潜在状态映射一个一维函数或序列x（t）∈R→y（t）∈R。

A、B：状态转移矩阵

下一个状态　当前状态　输入序列

状态更新阶段　$h'(t) = Ah(t) + Bx(t)$　(1a)

输出阶段　$y(t) = Ch(t)$　(1b)

输出序列　输出矩阵

状态更新

$$h_t = \overline{A}h_{t-1} + \overline{B}x_t \quad (2a)$$

$$y_t = Ch_t \quad (2b)$$

线性递归（循环模式）：进行有效的自回归推理（一次看到一个时间步）

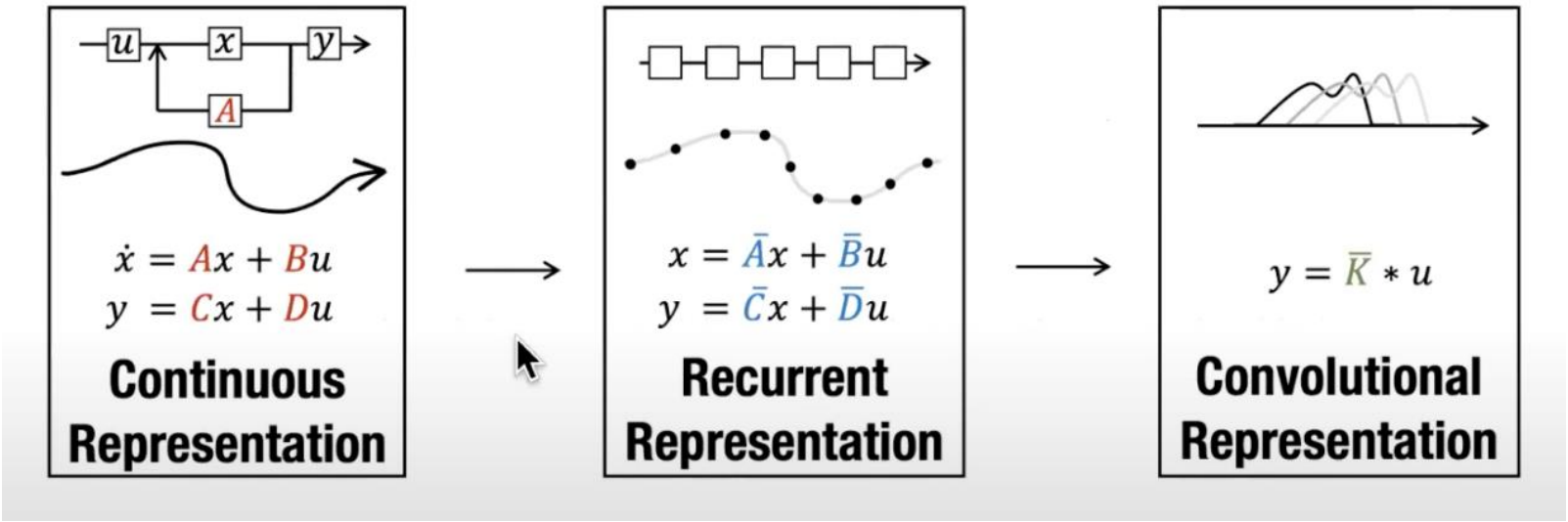
输出更新

所有可能的组合矩阵

$$\overline{K} = (C\overline{B}, C\overline{AB}, \dots, C\overline{A}^k\overline{B}, \dots) \quad (3a)$$

$$y = x * \overline{K} \quad (3b)$$

全局卷积：进行有效地并行训练（提前看到整个输入序列）



状态空间模型-S4

- **离散化**：通过固定公式$\bar{A} = f_A\ (\Delta,\ A)$ 和$\bar{B} = f_B\ (\Delta,\ A,\ B)$ 将“连续参数”(Δ, A, B)转换为“离散参数”$(\bar{A},\bar{B})$。离散化规则：零阶保持器（ZOH）；
- **作用**：离散化与连续时间系统有深刻的联系，还可以赋予它们额外的属性；
- **计算**：线性递归（2）　全局卷积（3）；
- **过程**：该模型使用卷积模式进行有效的并行训练（提前看到整个输入序列），并切换到循环模式进行有效的自回归推理（一次看到一个时间步）；
- LTI（线性时不变）：$(\Delta,\ A,\ B,\ C)$以及$(\bar{A},\bar{B})$对于所有时间步长都是固定的；
- ✓ Mamba认为LTI在对某些类型的数据进行建模时具有根本性的限制，mamba消除LTI限制，同时克服效率瓶颈。

P

ART THREE

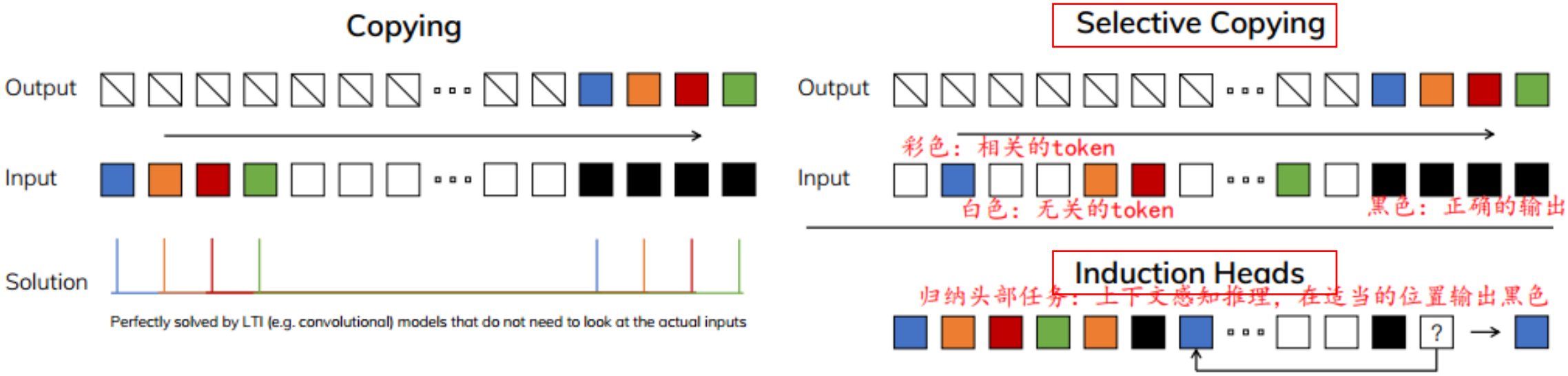
方法

02

Mamba（S6 / selective SSM）

- **选择性扫描算法（selective scan algorithm）**：允许模型过滤相关或者不相关的信息；
- **硬件感知的算法（hardware-aware algorithm）**：允许通过并行扫描、核融合和重计算有效地存储中间结果。

Mamba要解决的问题



- **Selection Copying**：选择性复制任务的输入间距随机，需要时变模型能够根据输入内容选择性地记住或忽略输入；
- **Induction Heads**：用于解释大型语言模型（LLMs）的上下文学习能力，要求模型在适当的上下文中产生正确的输出。

Mamba：选择性的保留信息

- 有选择地将数据压缩到状态中来保留一个小且有用的状态信息；
- 需要参数依赖于输入。

Algorithm 1 SSM (S4)

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

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

Algorithm

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

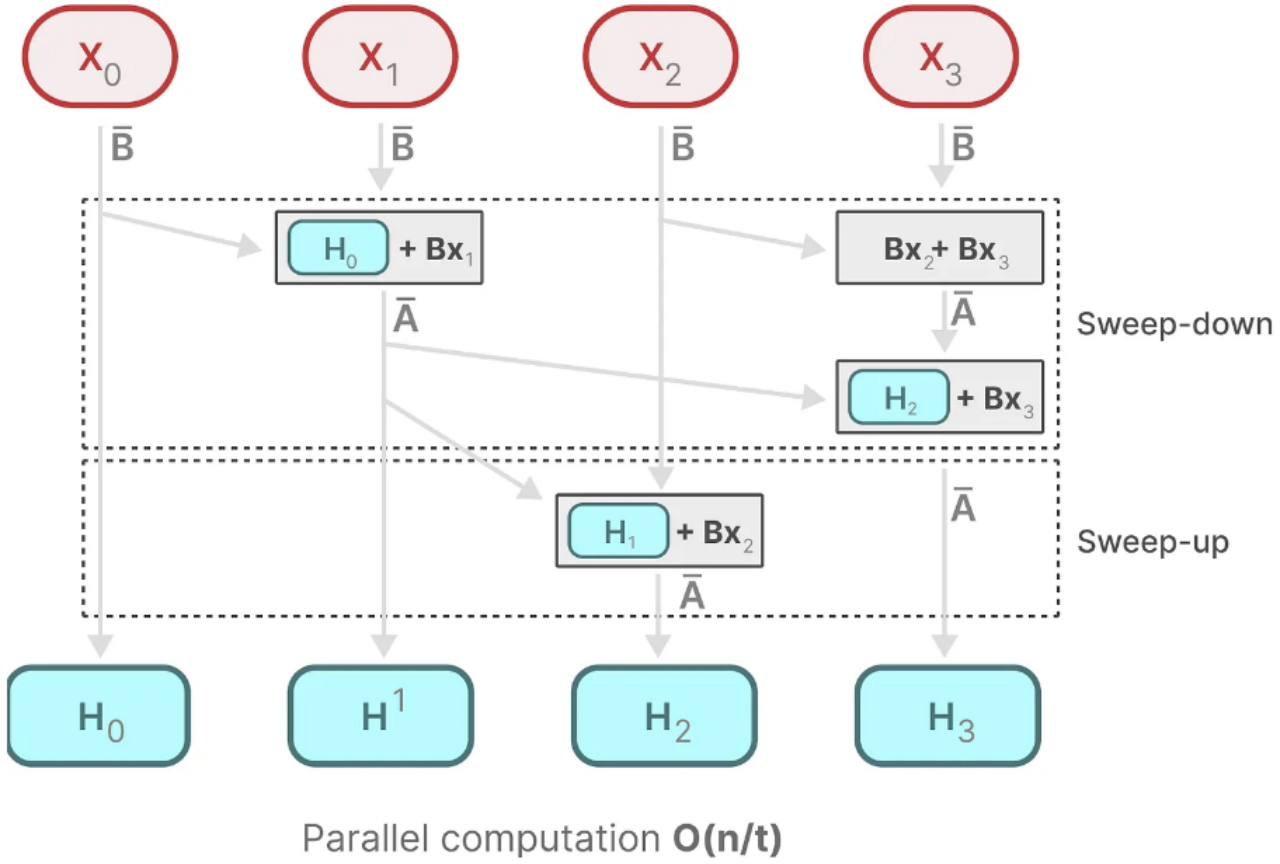
Algorithm 2 SSM + Selection (S6)

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

- 对于每个输入标记，有不同的B和C矩阵，这解决了内容感知的问题；
- 矩阵A保持不变，状态本身保持静态，但是它被影响的方式（通过B和C）是动态的；
- 对于△，当步长较小时，模型更倾向于忽略特定的单词，而更多的依赖前一个上下文，当步长较大时，模型更多地关注当前输入单词而不是上下文。

Mamba：并行扫描算法

- **原因：**选择性保留信息带来了问题，矩阵（B，C，△）是动态的，所以不能使用卷积表示进行计算，因为它假设一个固定的核。只能使用递归表示，就失去了卷积提供的并行化。

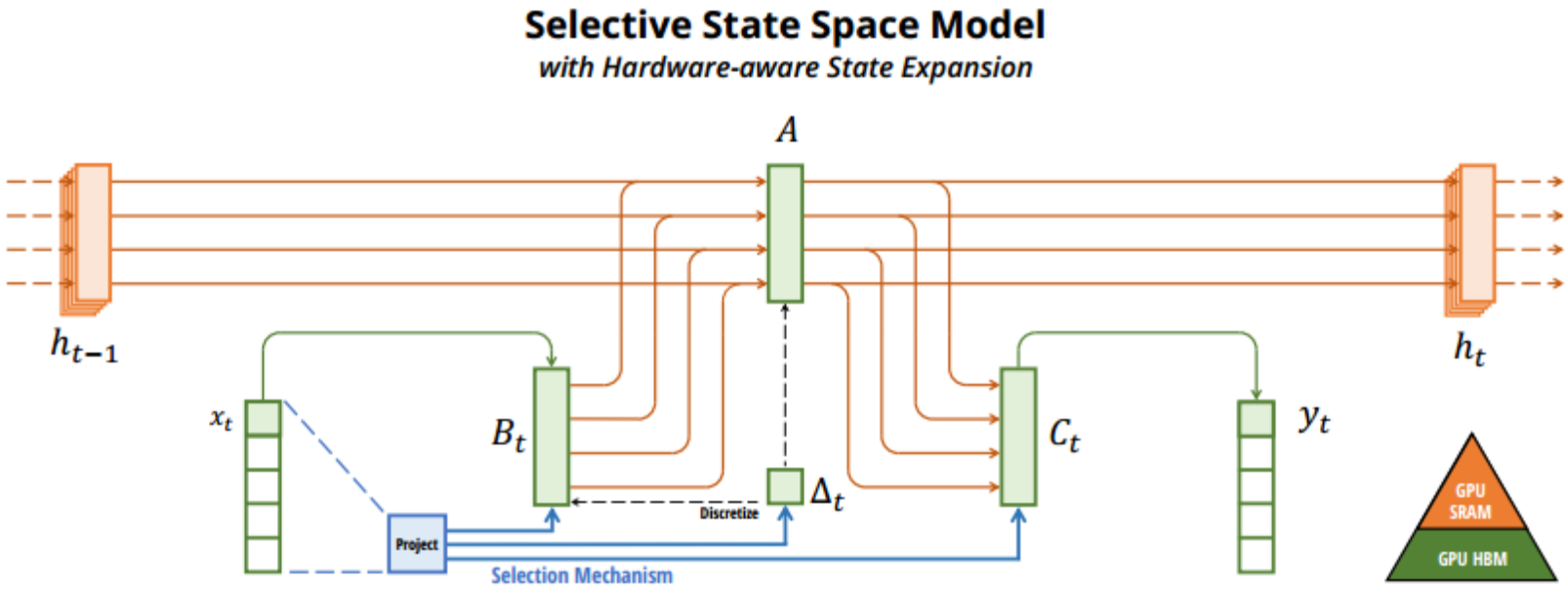


- 通过并行扫描算法使这成为可能；
- **优点**：允许模型在每个步骤中选择性地集成新信息，同时考虑之前的状态，从而在整个序列上有效地传播信息。

Mamba：硬件感知算法

- **GPU：**在小型但高效的SRAM和大型但略低效的DRAM之间的传输速度有限。频繁地在SRAM和DRAM之间的复制信息成为瓶颈。
- **核融合**：通过核融合来限制从DRAM到SRAM的次数，核融合允许模型防止写入中间结果，并持续执行计算，直到完成。
- **重计算**：中间状态不保存，但对于反向传递计算梯度是必要的。相反，作者在反向传递期间重新计算这些中间状态。虽然这看起来效率不高，但与从相对较慢的DRAM读取所有中间状态相比，它的开销要小得多。

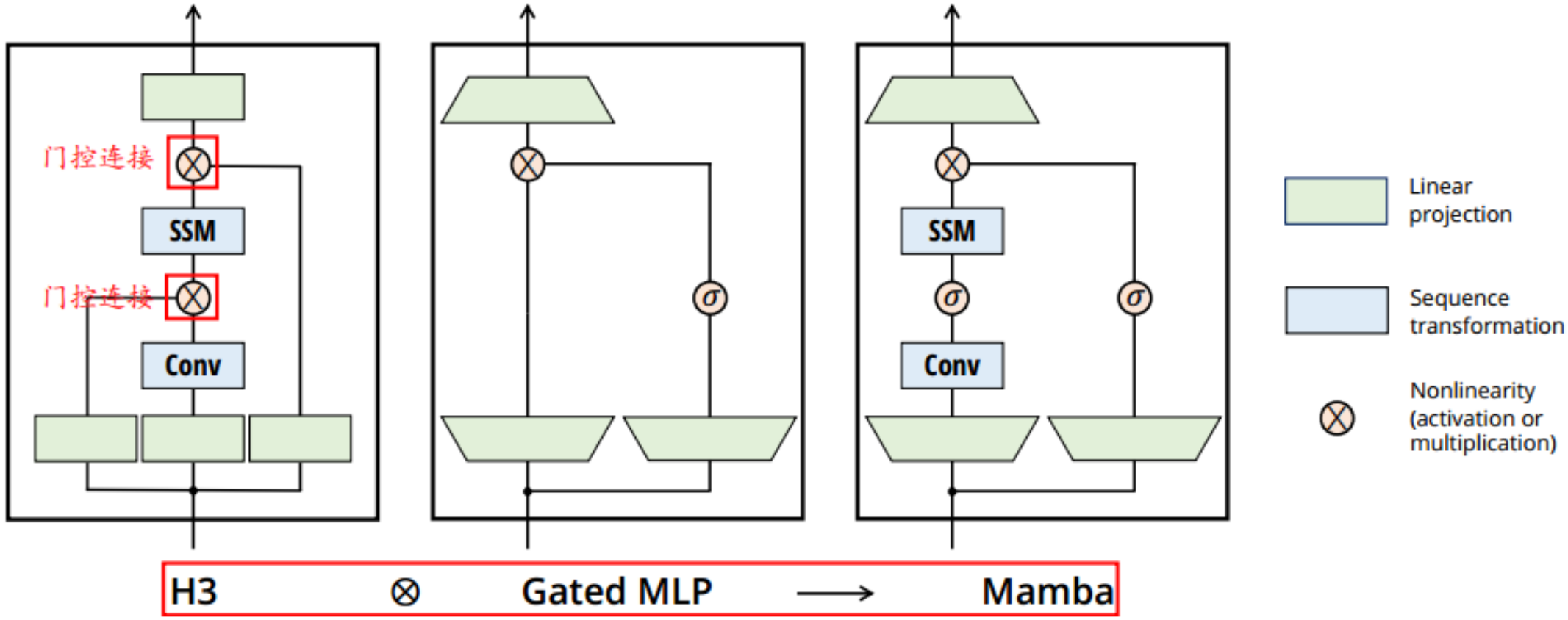
Overview



- 输入X_t通过选择机制映射得到B_t，$\triangle t$，C_t；
- 使用$\triangle t$，用零阶保持技术对A和B_t进行离散化；
- 离散化后的B和输入X_t相乘，离散化后的A和原始状态 h_{t-1} 相乘，将这两项相加得到新的状态h_t；
- 新状态和C_t相乘，得到输出y_t。

A Simplified Selective SSM Architecture

- 将H3和一个多层感知机（MLP）组合成一个组件



P

ART FOUR

实验

03

Selective Copying

- ✓ **结论**：H3和Mamba等门控机制只能部分提高性能，而选择机制很容易解决这个问题，特别是在与这些更强大的架构结合使用时。

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

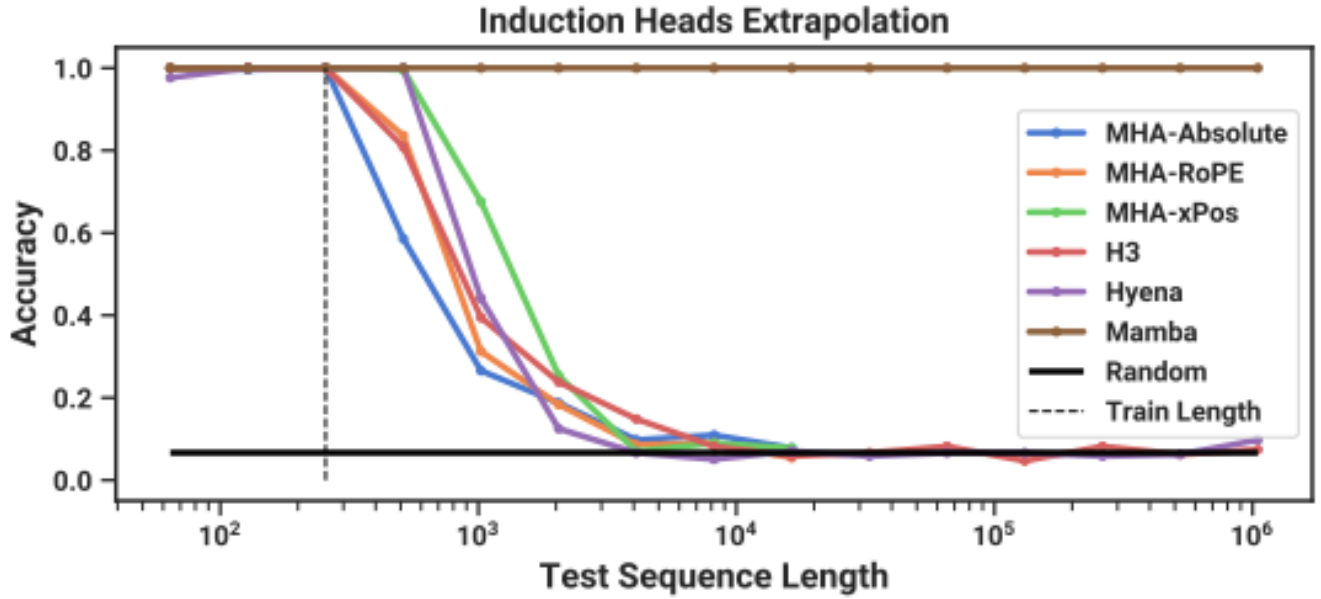


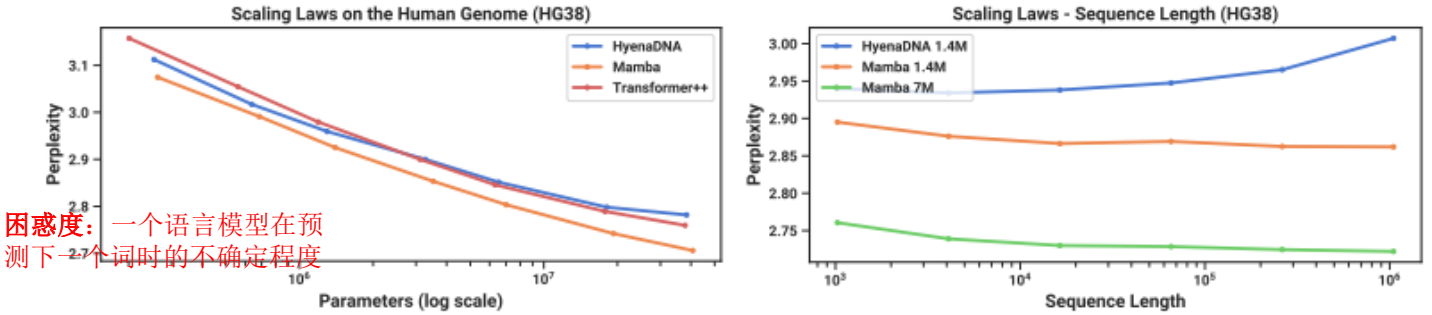
Table 2: (**Induction Heads.**) Models are trained on sequence length $2^8 = 256$, and tested on increasing sequence lengths of $2^6 = 64$ up to $2^{20} = 1048576$. Full numbers in Table 11.

✓ **结论**：Mamba(或者更准确地说，是它的选择性SSM层)能够完美地解决这个任务，因为它能够选择性地记住相关的token，同时忽略中间的所有其他内容。

语言建模&DNA建模&音频建模和生成

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



困惑度：一个语言模型在预测下一个词时的不确定程度

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.

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**

速度和内存基准& 模型削减

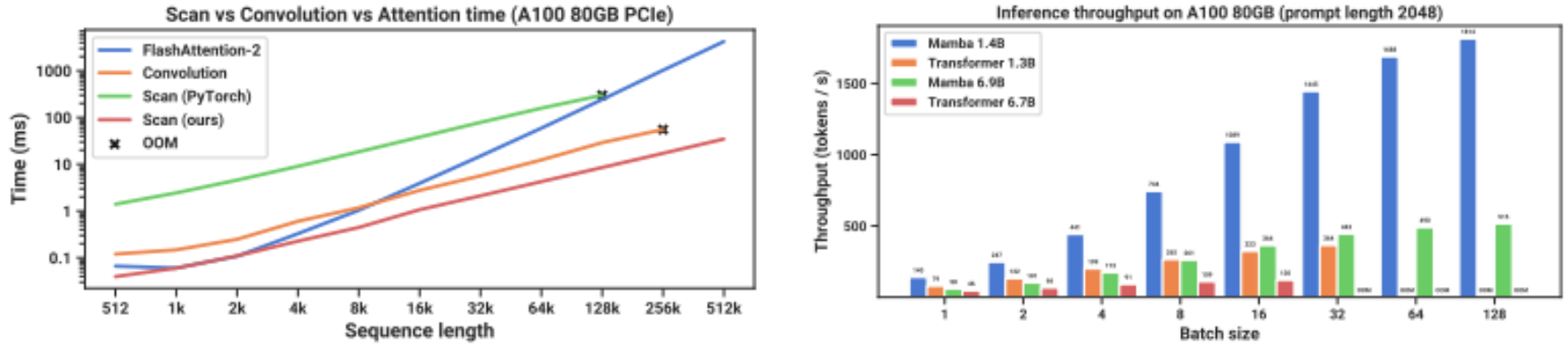


Figure 8: (**Efficiency Benchmarks.**) (*Left*) Training: our efficient scan is 40× faster than a standard implementation. (*Right*) Inference: as a recurrent model, Mamba can achieve 5× higher throughput than Transformers.

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**

速度和内存基准 & 模型削减

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

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 B and C (*Bottom*) Selective B and 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 B and 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

P

ART FIVE

结论

05

结论

- **核心创新**：引入选择性机制，这一机制通过使SSM参数依赖于输入，使得模型可以根据不同的输入动态调整其行为。这类似于RNN中的门控机制，但在SSM的框架下提供了更广泛的应用可能性。通过这种方式，Mamba能够有效地过滤掉无关信息，同时保留和加强与任务相关的信息，提高了对长序列数据的处理能力。
- 为了**优化计算效率**，Mamba采用了硬件感知算法，特别是利用GPU的内存层次结构来提高扫描操作的计算速度和降低内存需求。这种方法结合了RNN的递归计算效率和CNN的并行处理优势，使得Mamba在处理长序列数据时更加高效。

感谢观看！

汇报人：闫林枝

GOODBYE