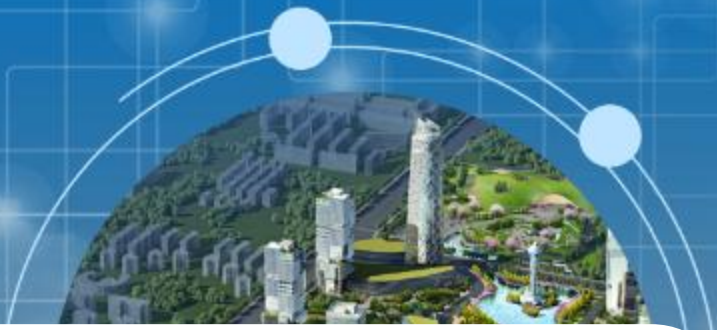


State Space Model

--- Model Analysis and Applications

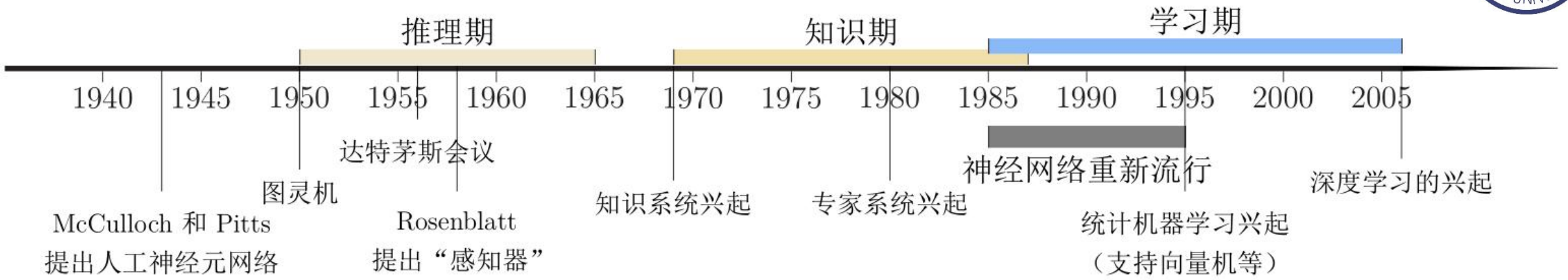
Xiao Wang (王逍)
xiaowang@ahu.edu.cn

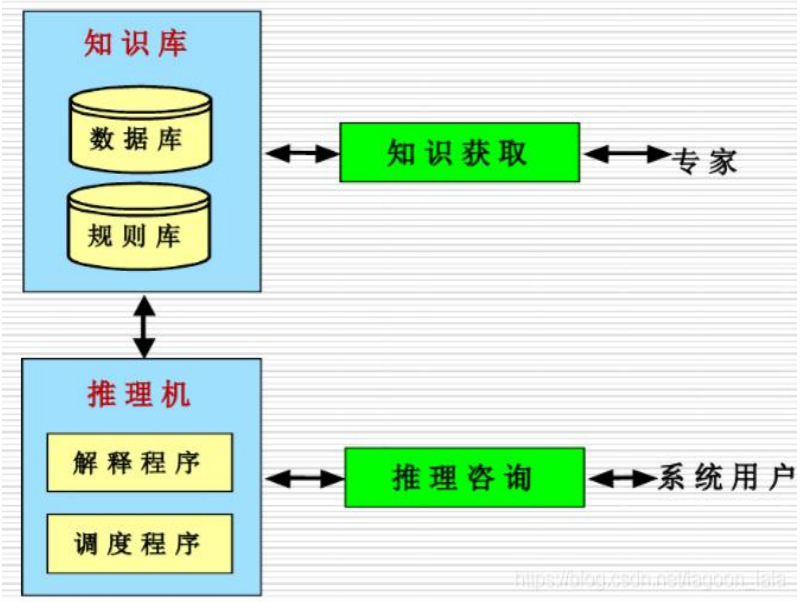
Anhui University School of Computer Science and Technology

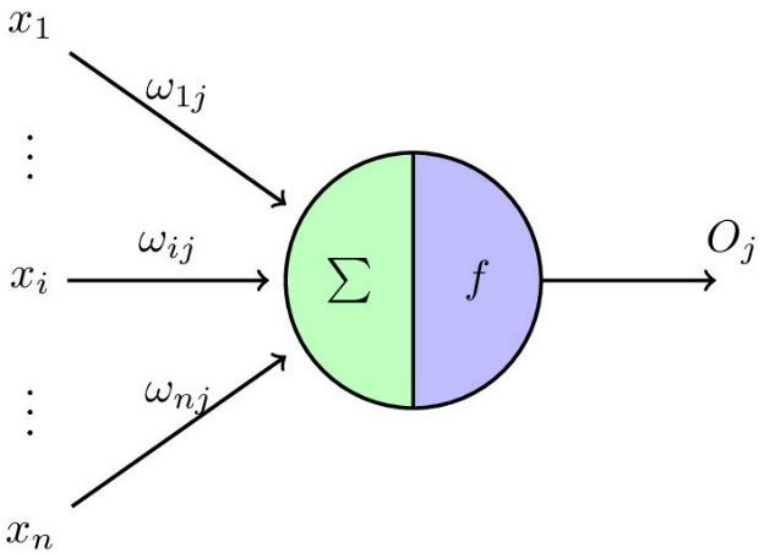
Overview

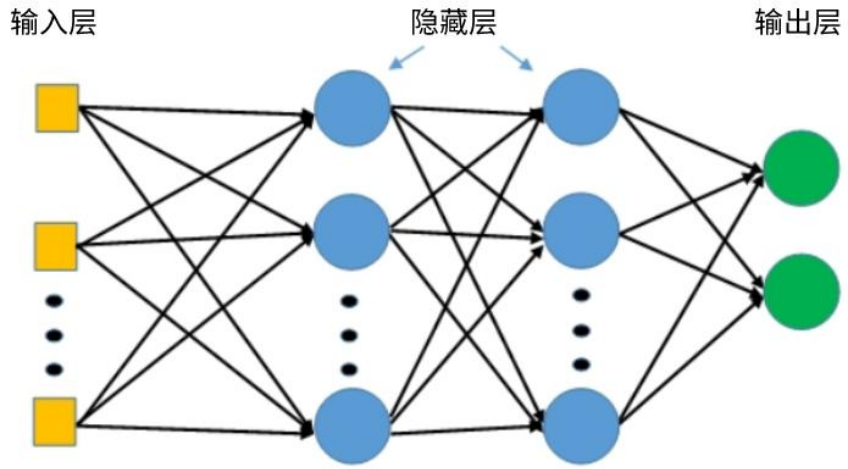
- **Review: CNN, RNN, Transformer**
- **State Space Model**
  - Model Formulation
  - Applications
- **Future Works**
- **Discussion**

Review: CNN, RNN, Transformer



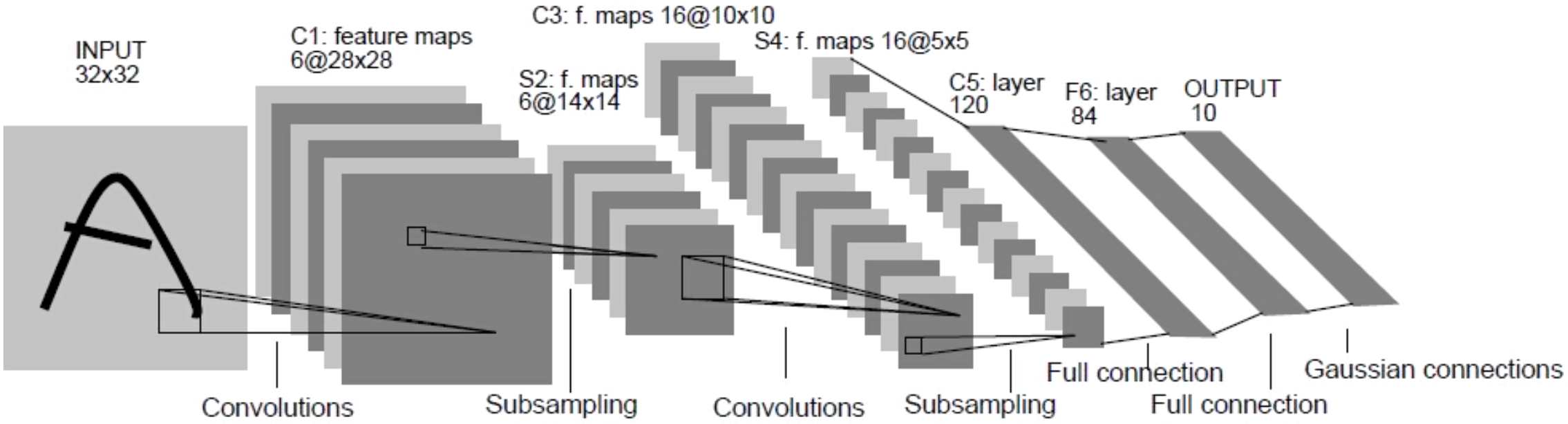


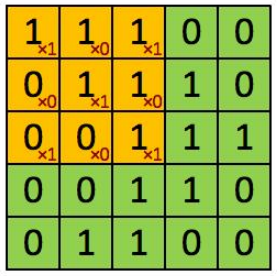


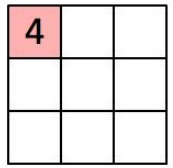


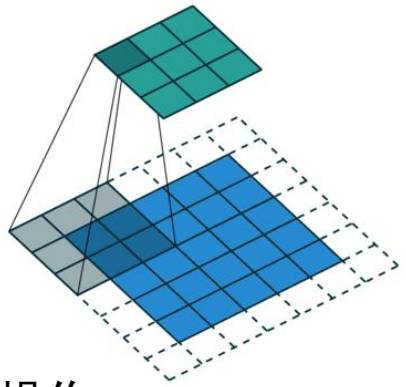
Review: CNN, RNN, Transformer

- Convolutional Neural Networks

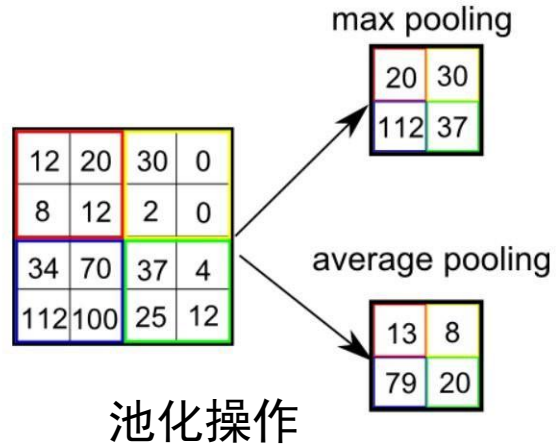








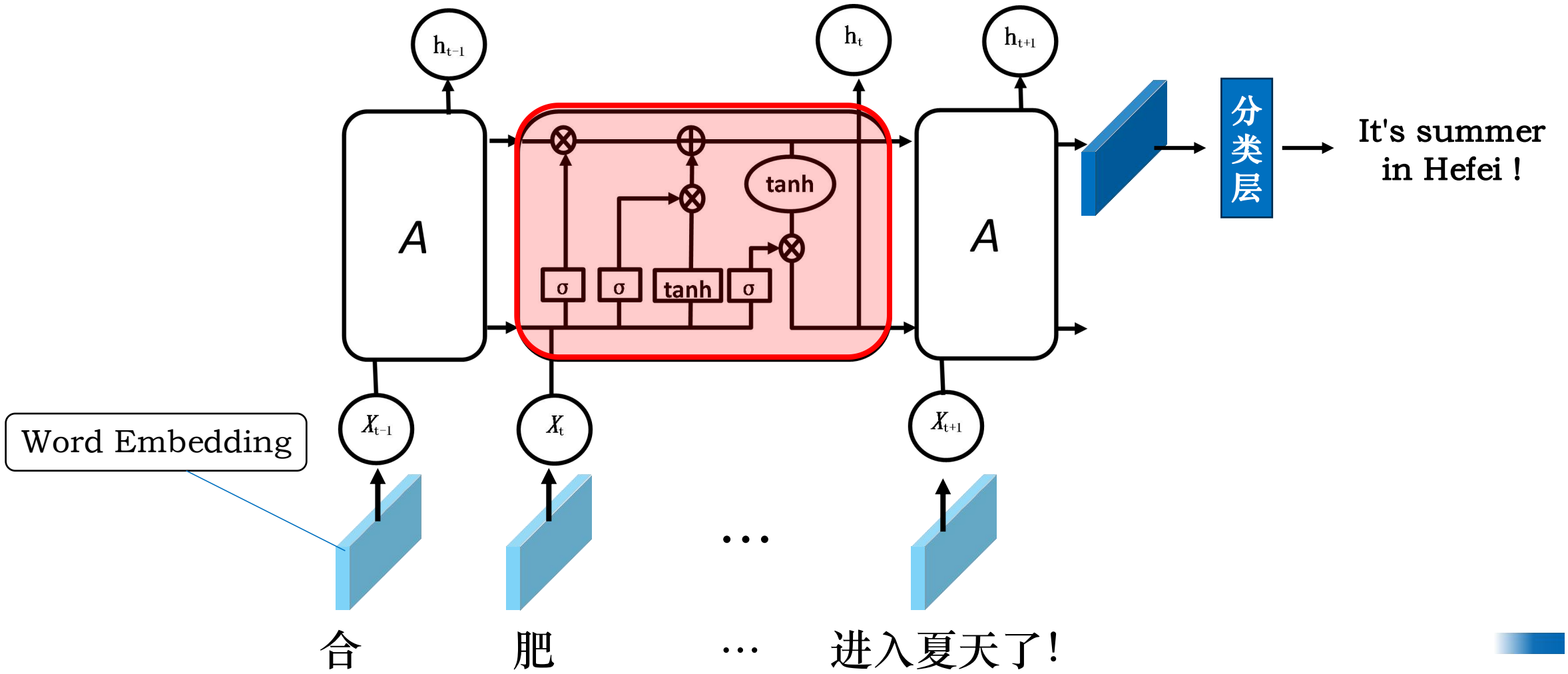
卷积操作



池化操作

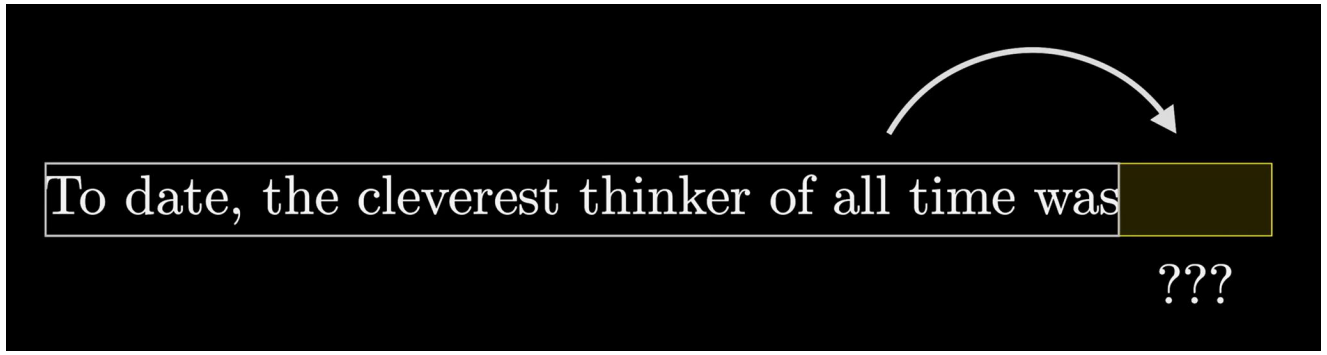
Review: CNN, RNN, Transformer

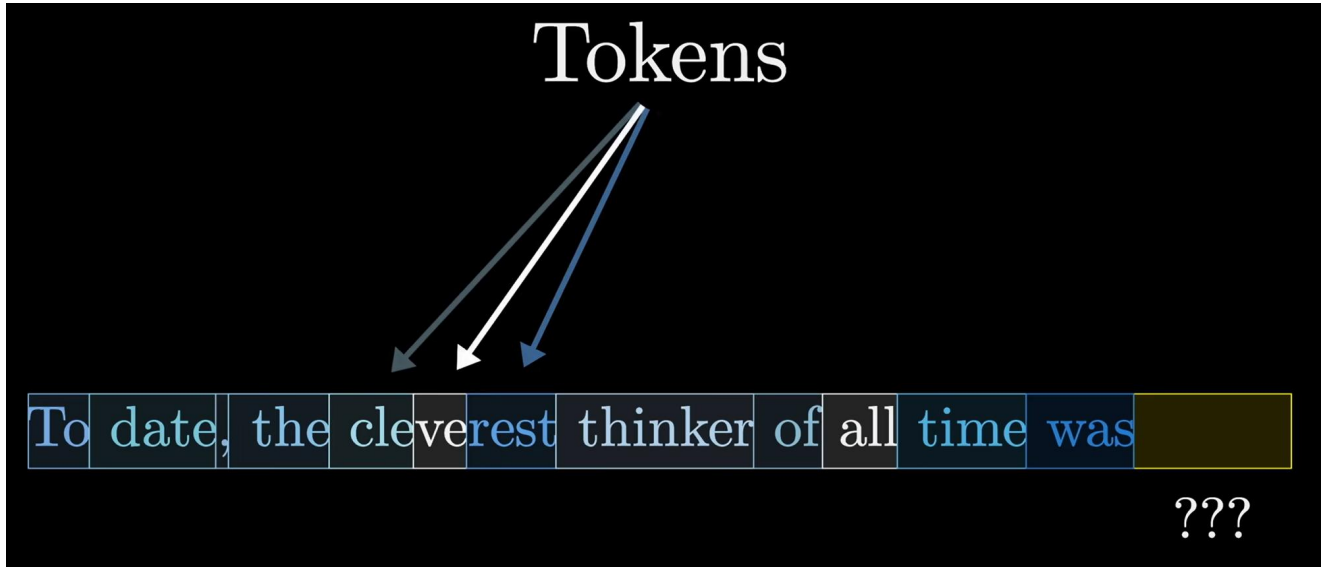
- LSTM (Long Short-term Memory) / GRU

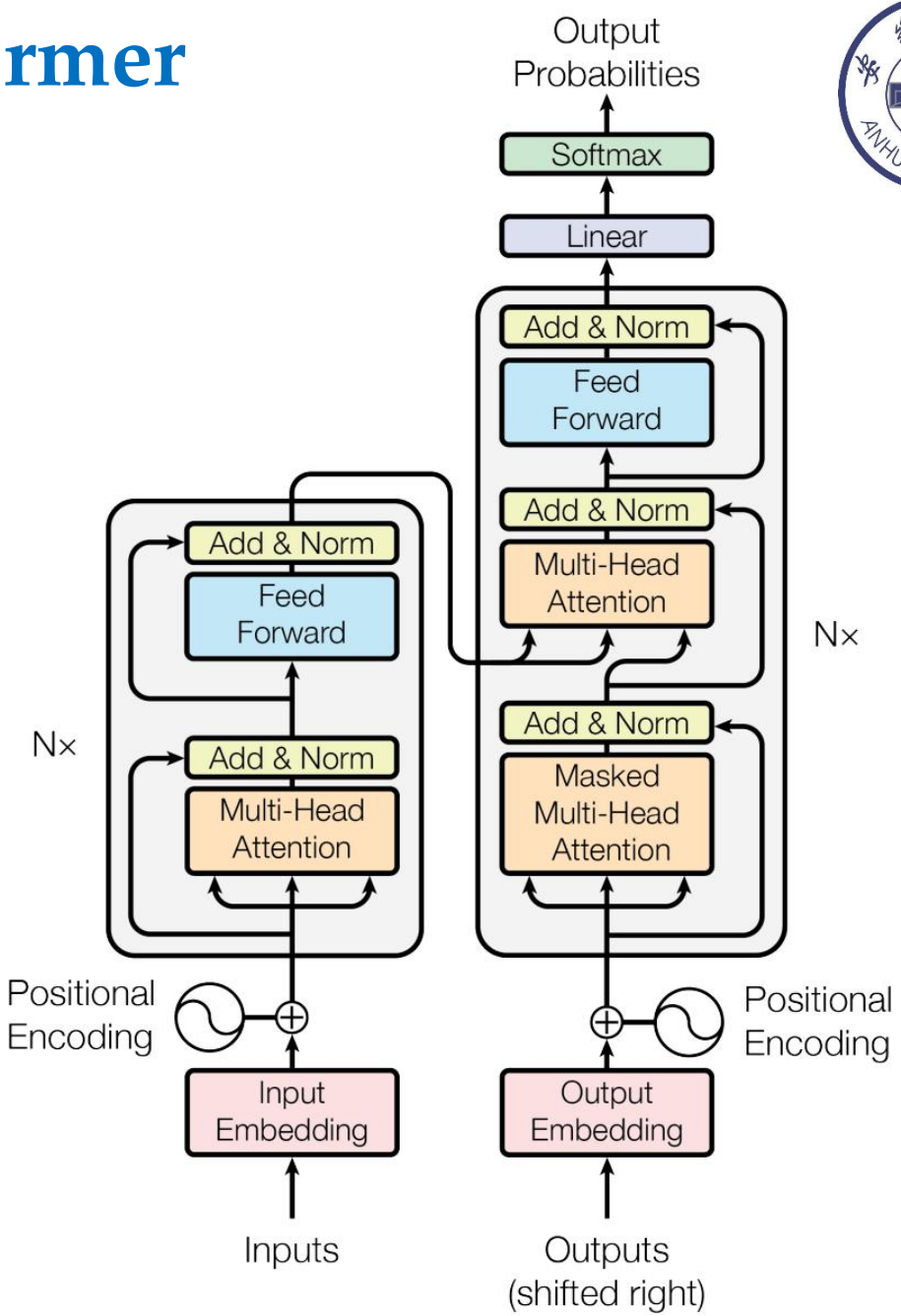


Review: CNN, RNN, Transformer

- Transformer

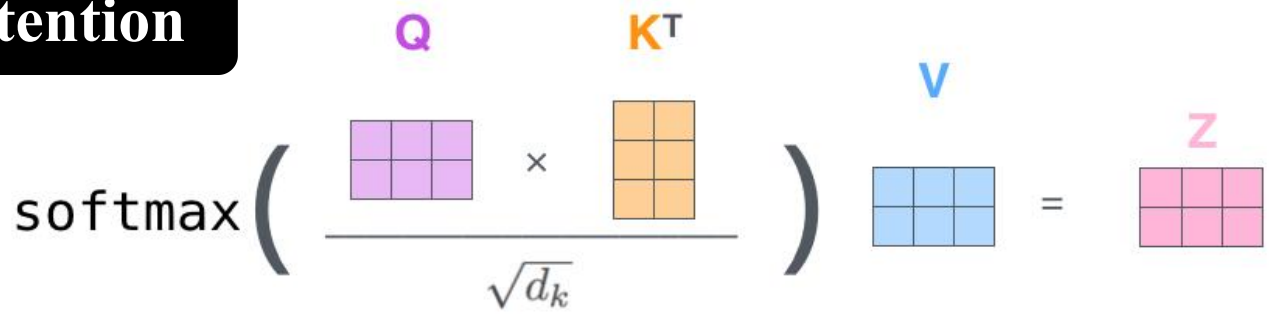




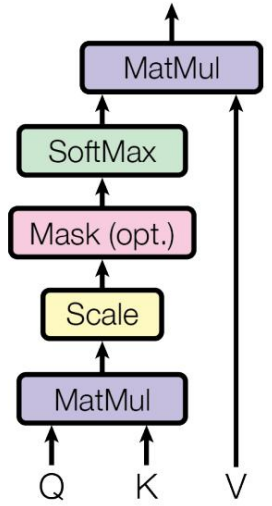


Review: CNN, RNN, Transformer

Self-Attention



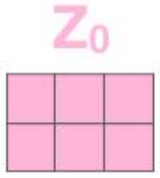
Scaled Dot-Product Attention



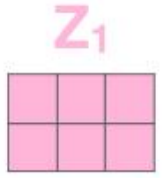
Multi-Head Self-Attention

Calculating attention separately in eight different attention heads

ATTENTION HEAD #0



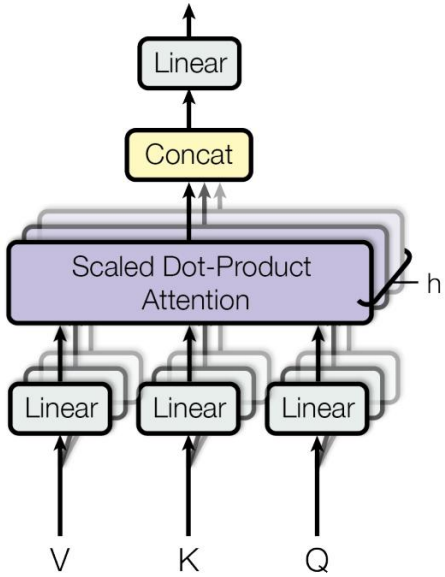
ATTENTION HEAD #1



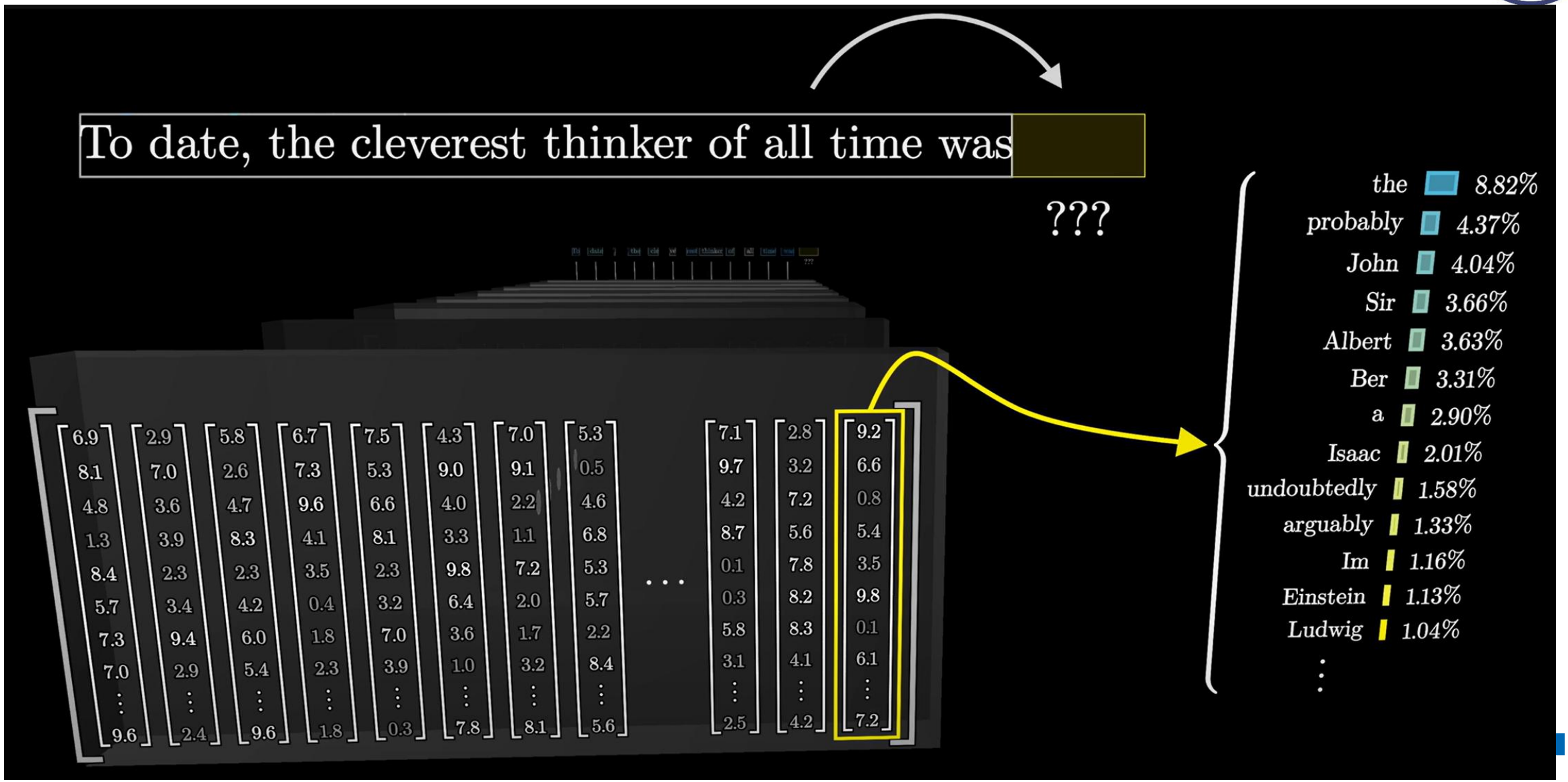
...

ATTENTION HEAD #7

Multi-Head Attention

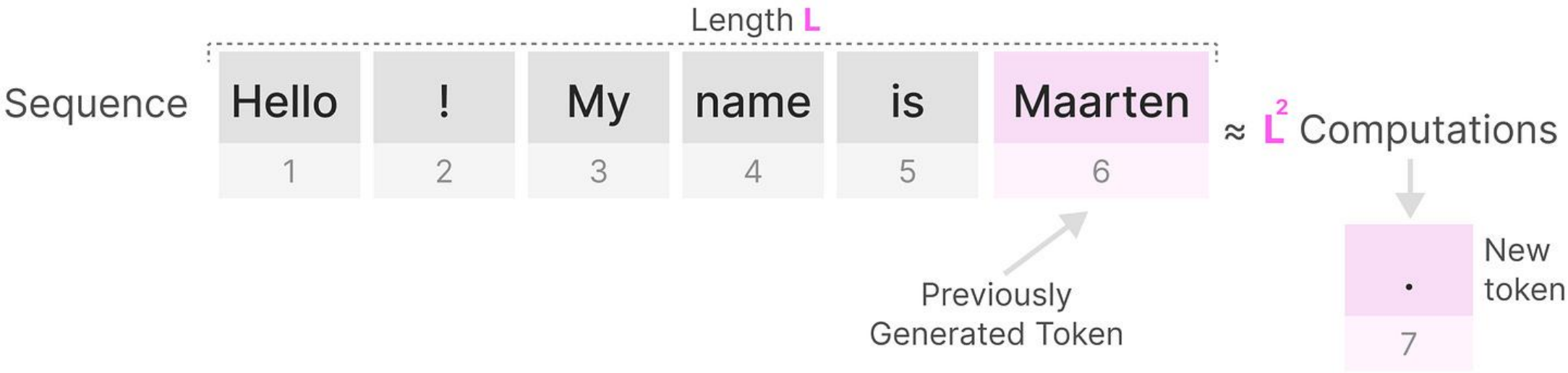


Review: CNN, RNN, Transformer



Review: CNN, RNN, Transformer

When generating the next token, we need to **re-calculate the attention for the entire sequence**, even if we already generated some tokens.



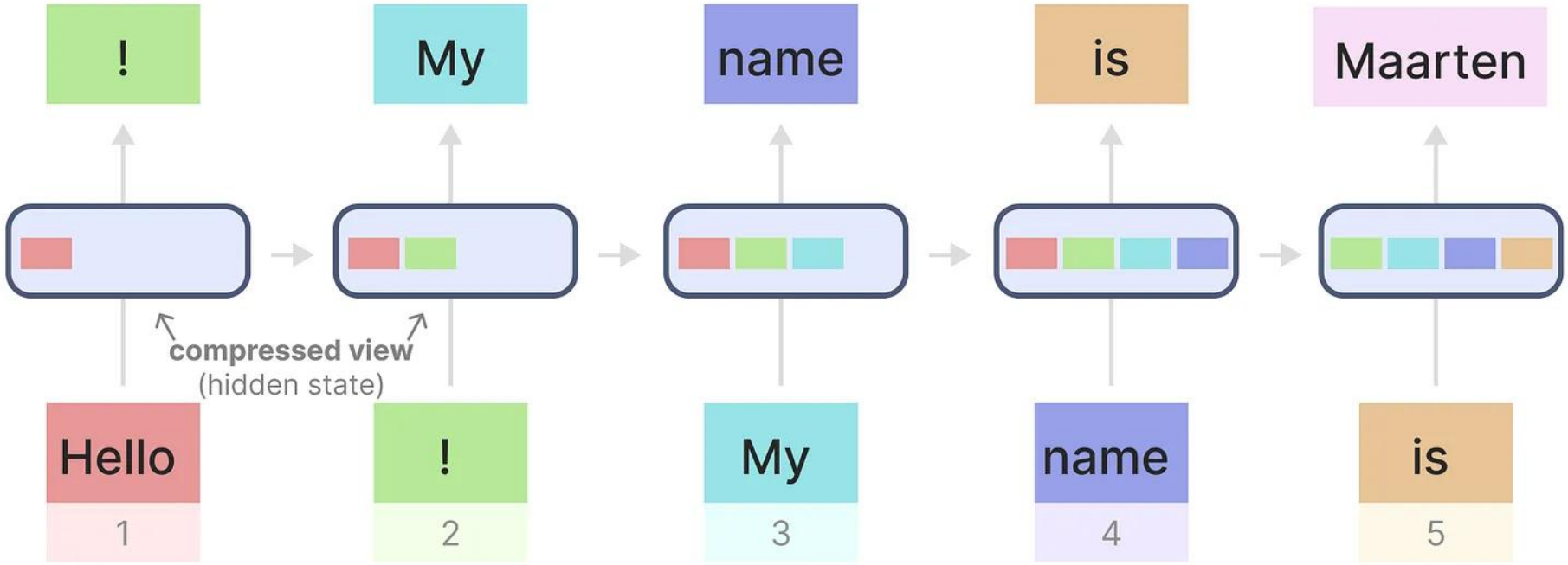
Generating tokens for a sequence of length L needs roughly L^2 computations which can be costly if the sequence length increases.

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

Review: CNN, RNN, Transformer

When generating the output, the RNN only needs to consider the **previous hidden state** and **current input**. It prevents recalculating all previous hidden states which is what a Transformer would do.

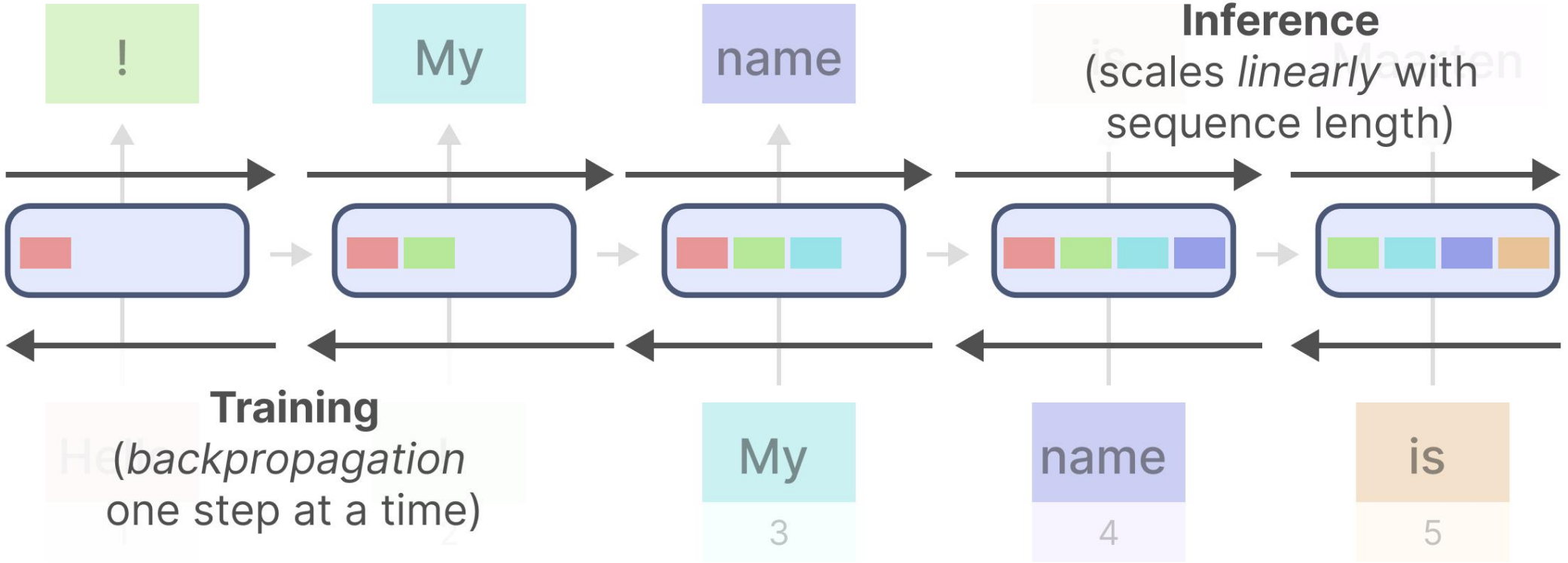
RNNs can do inference fast as it scales linearly with the sequence length! In theory, it can even have an *infinite context length*.



Review: CNN, RNN, Transformer

- **Issues of RNN:**

  Training cannot be done in parallel since it needs to go through each step at a time sequentially.

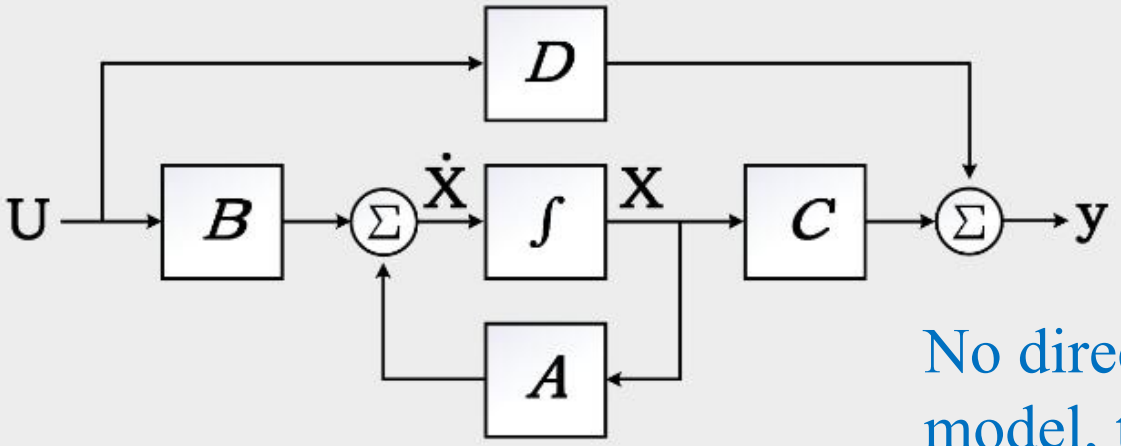


Review: CNN, RNN, Transformer

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

Can we somehow find an architecture that does **parallelize training like Transformers** whilst still performing **inference that scales linearly with sequence length**?

State Space Model : Continuous SSM System



No direct feedthrough in the system model, the D(t) is a zero matrix.

$$\dot{\mathbf{X}}(t) = \mathbf{A}(t)\mathbf{X}(t) + \mathbf{B}(t)\mathbf{U}(t)$$
$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{X}(t) + \mathbf{D}(t)\mathbf{U}(t)$$

$$\dot{\mathbf{X}}(t) = \mathbf{A}(t)\mathbf{X}(t) + \mathbf{B}(t)\mathbf{U}(t)$$
$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{X}(t).$$

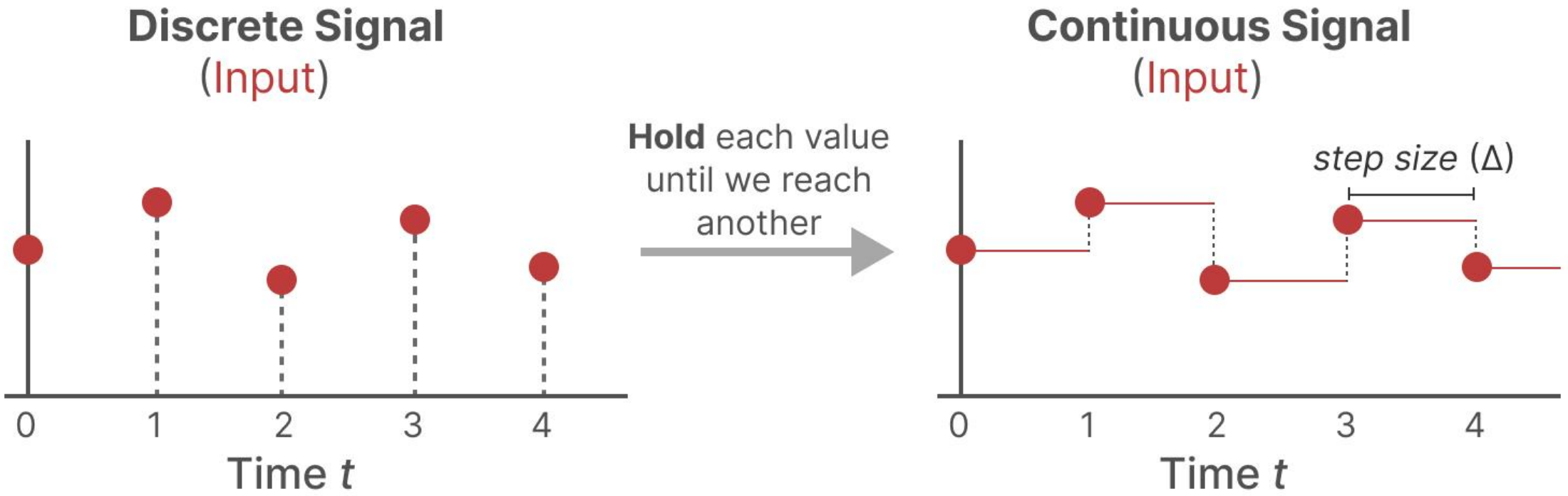
A(t): state matrix
B(t): input matrix
C(t): output matrix
D(t): feed-forward matrix

U(t): input vector
X(t): state vector
y(t): output vector

State Space Model : Continuous SSM System

- **From a Continuous to a Discrete Signal** *ZOH (Zero-order hold)*

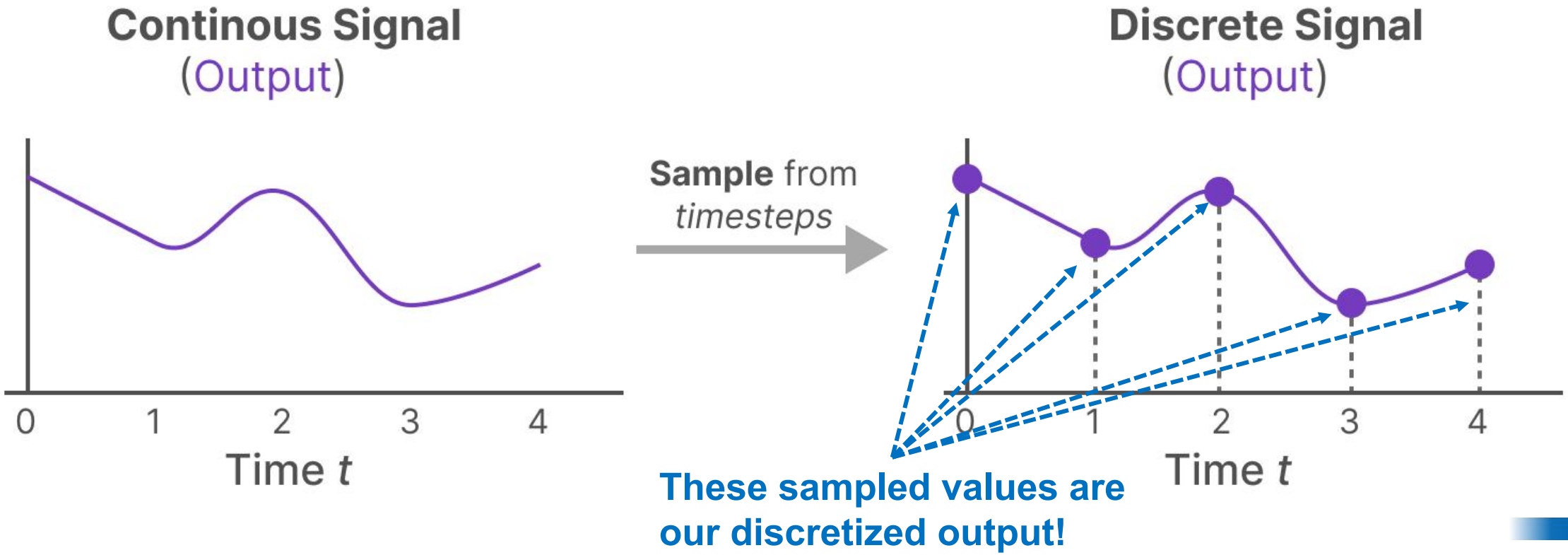
Every time we receive a discrete signal, we hold its value until we receive a new discrete signal. This process creates a continuous signal the SSM can use:



State Space Model : Continuous SSM System

- **From a Continuous to a Discrete Signal** *ZOH (Zero-order hold)*

Now that we have a continuous signal for our input, we can generate a continuous output and only sample the values according to the time steps of the input.



State Space Model : Continuous SSM System

Input
(Sequence)

Continuous SSM

$$\dot{\mathbf{X}}(t) = \mathbf{A}(t)\mathbf{X}(t) + \mathbf{B}(t)\mathbf{U}(t)$$
$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{X}(t).$$

Output
(Sequence)

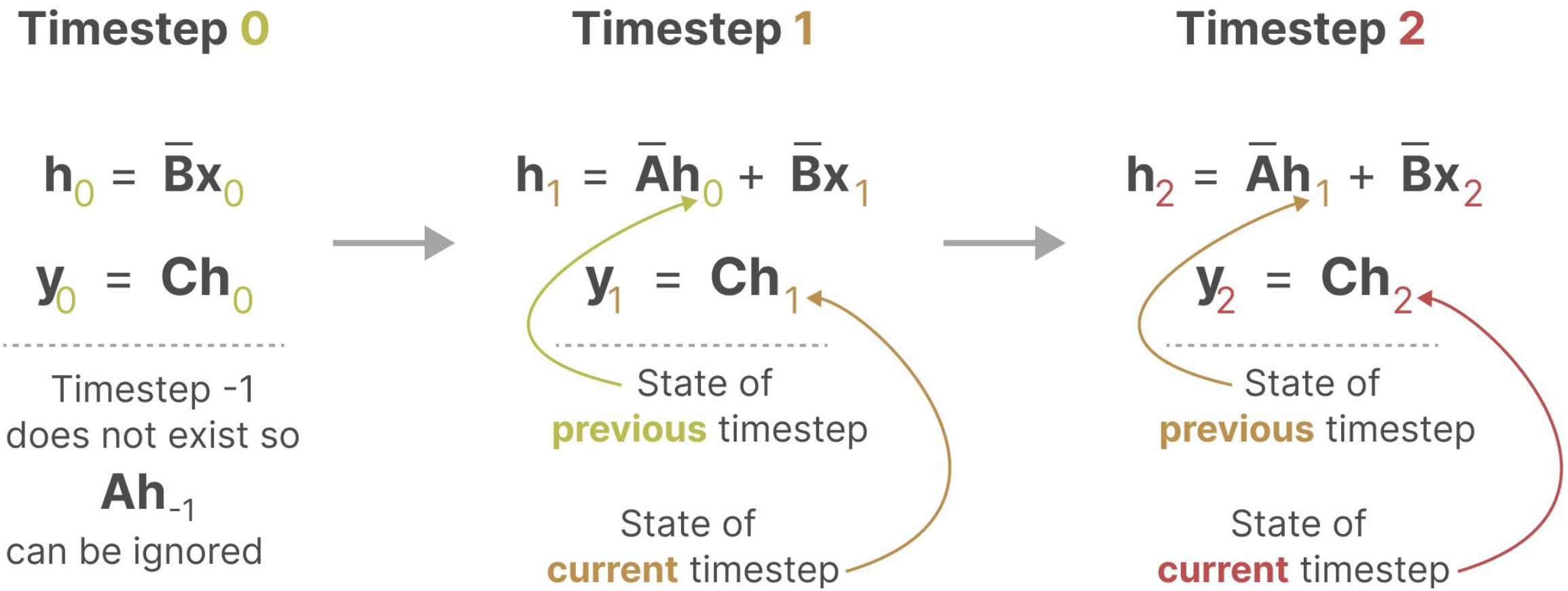
Discrete SSM

$$\mathbf{X}_t = \overline{\mathbf{A}}\mathbf{X}_{t-1} + \overline{\mathbf{B}}\mathbf{U}_t$$
$$\mathbf{y}_t = \mathbf{C}\mathbf{X}_t$$

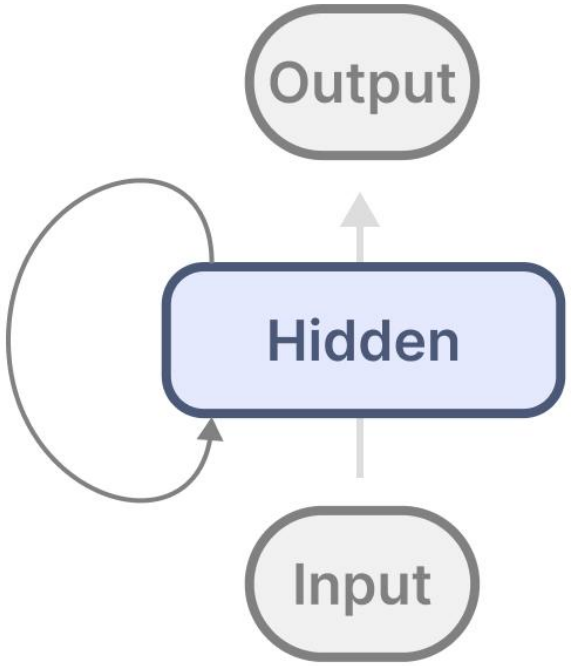
$$\overline{\mathbf{A}} = exp(\Delta\mathbf{A}), \quad \overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B},$$

Δ denotes the step size.

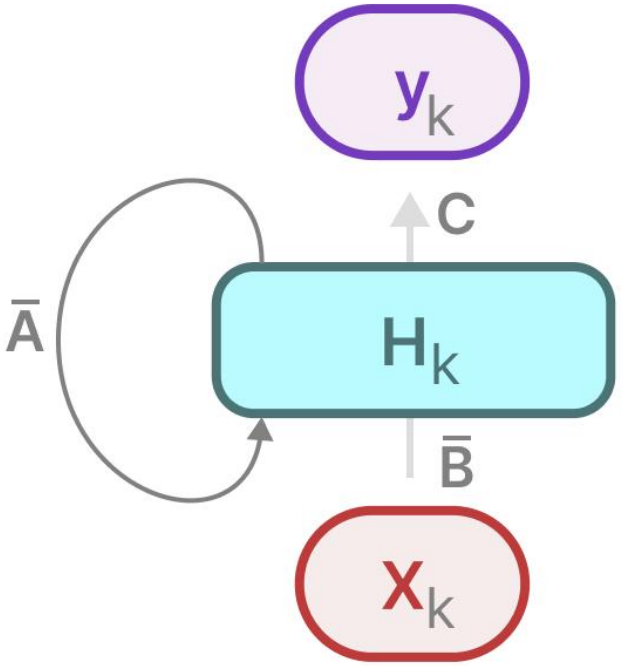
State Space Model : RNN view



State Space Model : RNN view

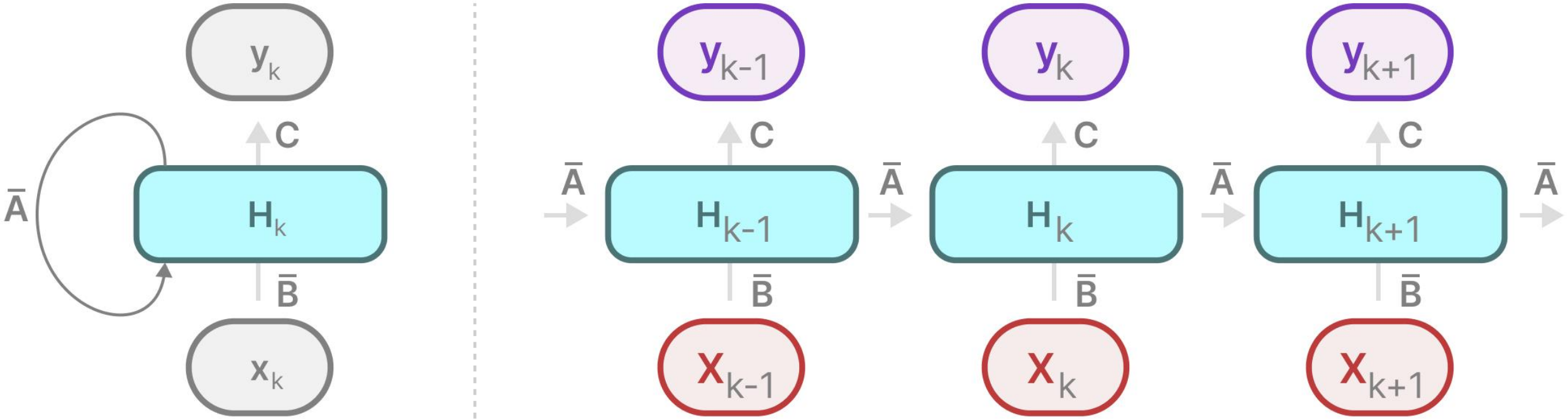


RNN



SSM
(Recurrent)

State Space Model : RNN view



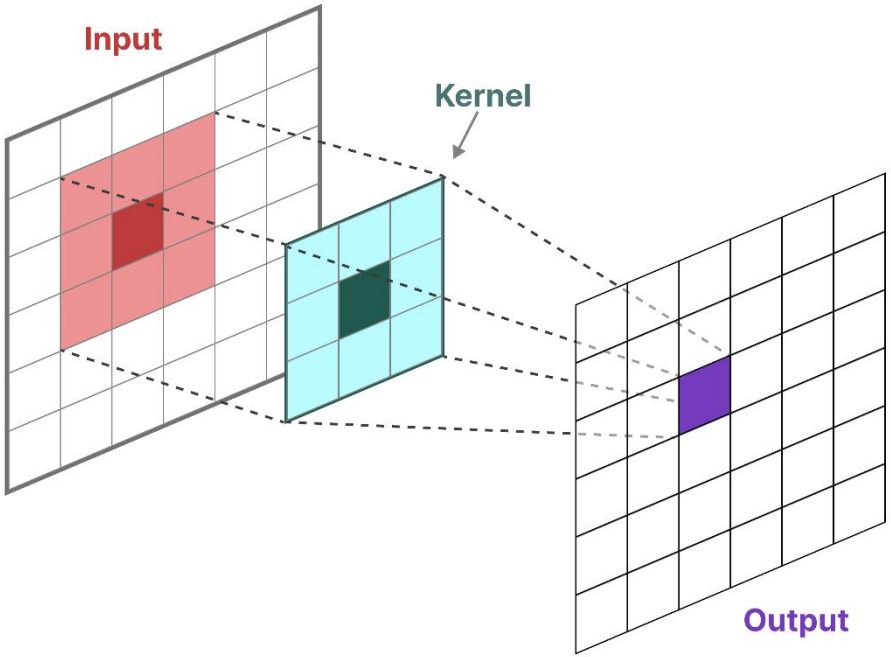
SSM
(Recurrent)

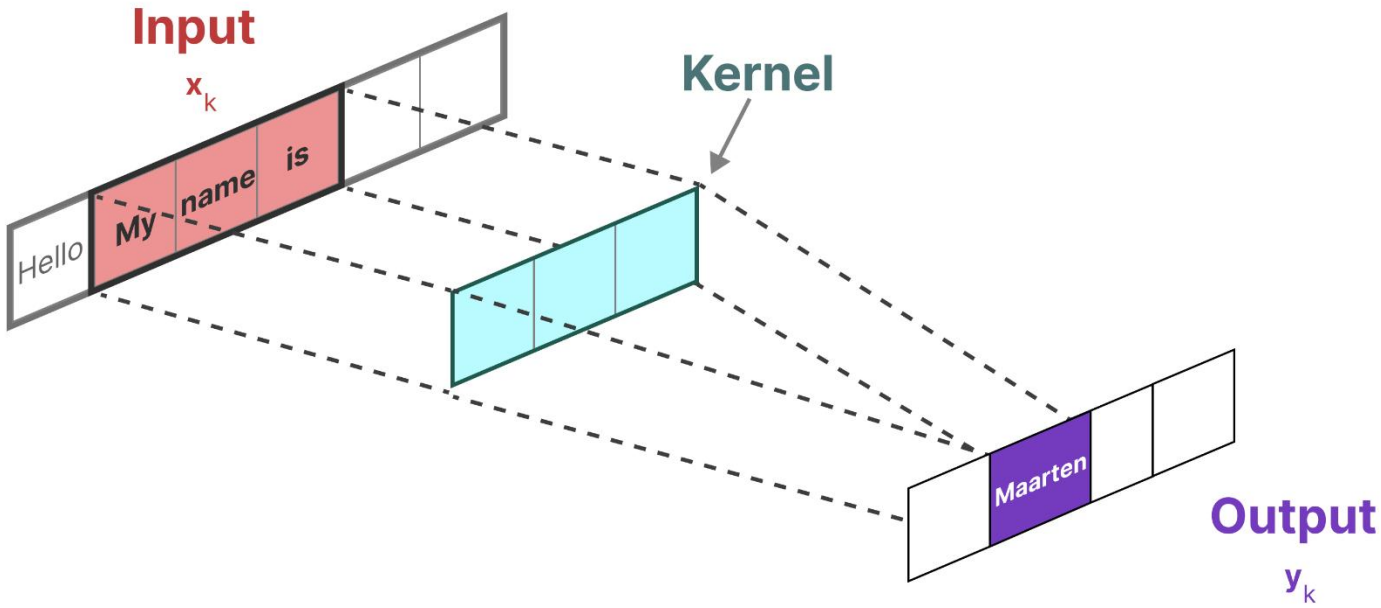
- Fast Inference
- Slow Training

SSM
(Recurrent + Unfolded)

State Space Model : CNN view

- Convolution Operation





State Space Model : CNN view

- **Convolution Operation**

$$\begin{aligned} \mathbf{h}_t &= \bar{\mathbf{A}}\mathbf{h}_{t-1} + \bar{\mathbf{B}}\mathbf{x}_t \\ \mathbf{y}_t &= \mathbf{C}\mathbf{h}_t. \end{aligned}$$

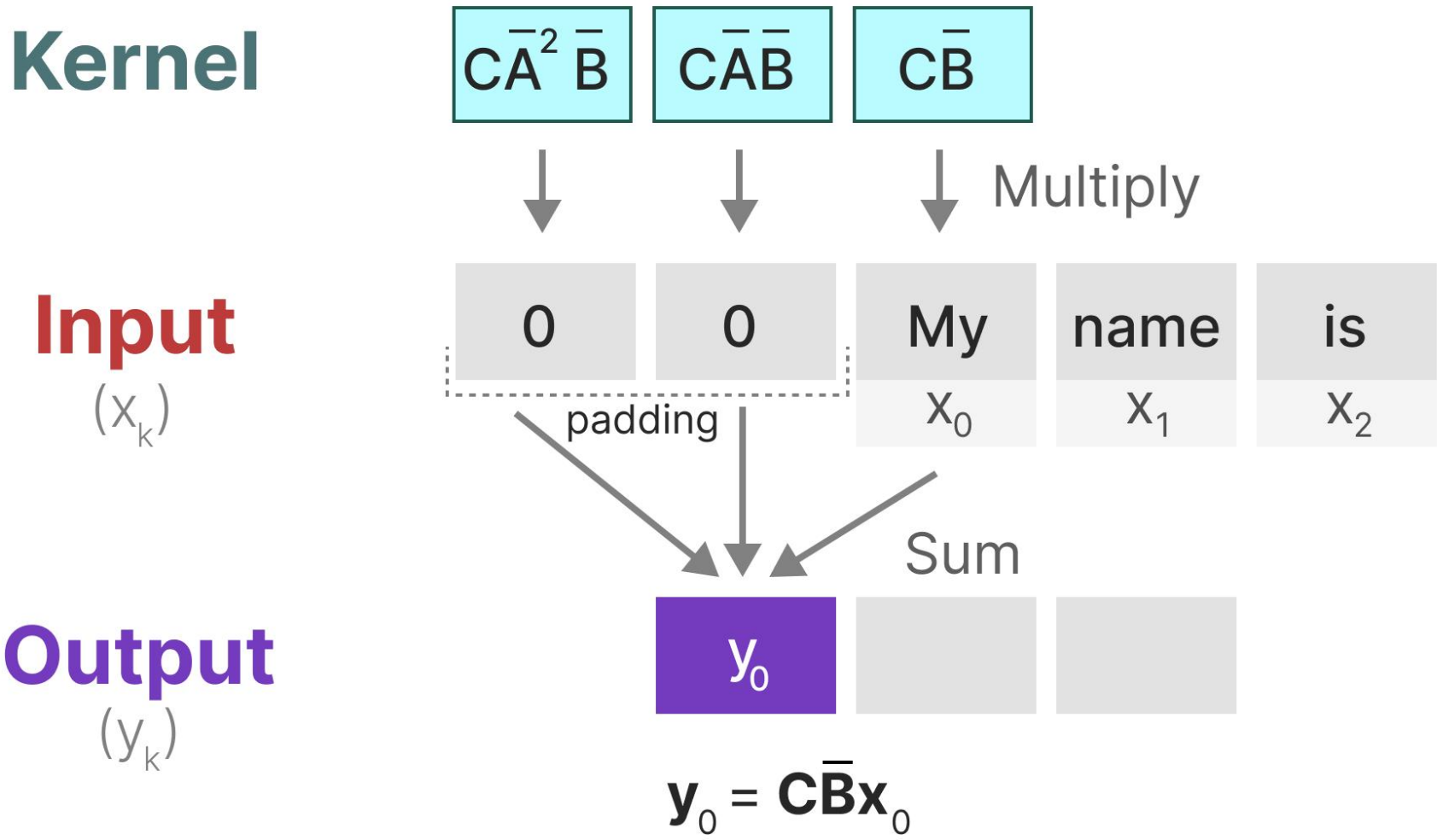
$$\begin{aligned} \mathbf{y}_0 &= \mathbf{C}\bar{\mathbf{A}}^0\bar{\mathbf{B}}\mathbf{x}_0 \\ \mathbf{y}_1 &= \mathbf{C}\bar{\mathbf{A}}^1\bar{\mathbf{B}}\mathbf{x}_0 + \mathbf{C}\bar{\mathbf{A}}^0\bar{\mathbf{B}}\mathbf{x}_1 \\ \mathbf{y}_2 &= \mathbf{C}\bar{\mathbf{A}}^2\bar{\mathbf{B}}\mathbf{x}_0 + \mathbf{C}\bar{\mathbf{A}}^1\bar{\mathbf{B}}\mathbf{x}_1 + \mathbf{C}\bar{\mathbf{A}}^0\bar{\mathbf{B}}\mathbf{x}_2 \end{aligned}$$

The multipliers of the **last item** and **the second to last item** are always: $\mathbf{C}\bar{\mathbf{A}}^0\bar{\mathbf{B}}$ $\mathbf{C}\bar{\mathbf{A}}^1\bar{\mathbf{B}}$

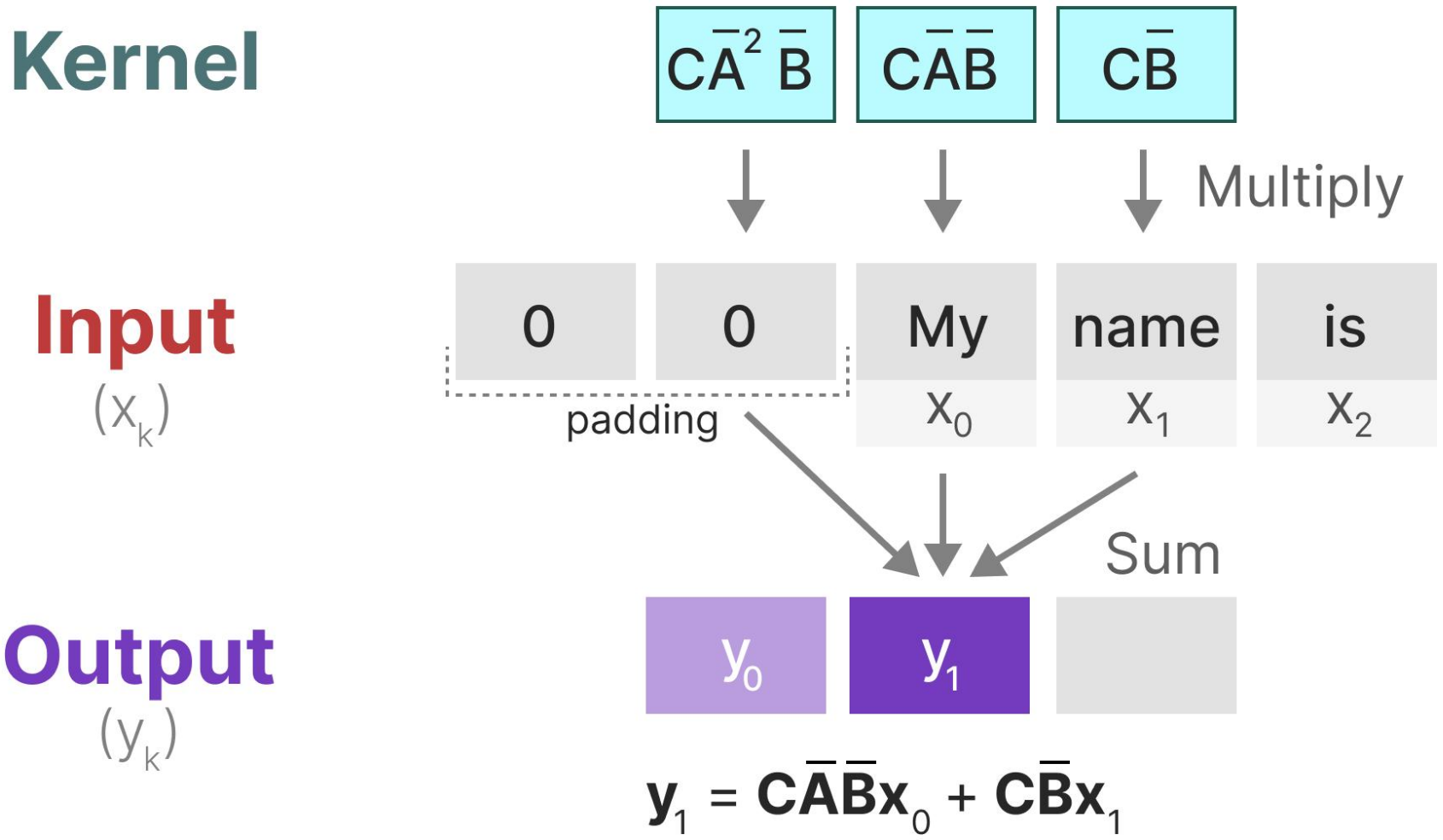
Thus, we can treat them as the ***convolutional filters*** (or ***kernels***):

$$\begin{aligned} \bar{\mathbf{K}} &= (\mathbf{C}\bar{\mathbf{B}}, \mathbf{C}\overline{\mathbf{AB}}, \ldots, \mathbf{C}\bar{\mathbf{A}}^k\bar{\mathbf{B}}, \ldots) \\ \mathbf{y} &= \mathbf{x} * \bar{\mathbf{K}} \end{aligned}$$

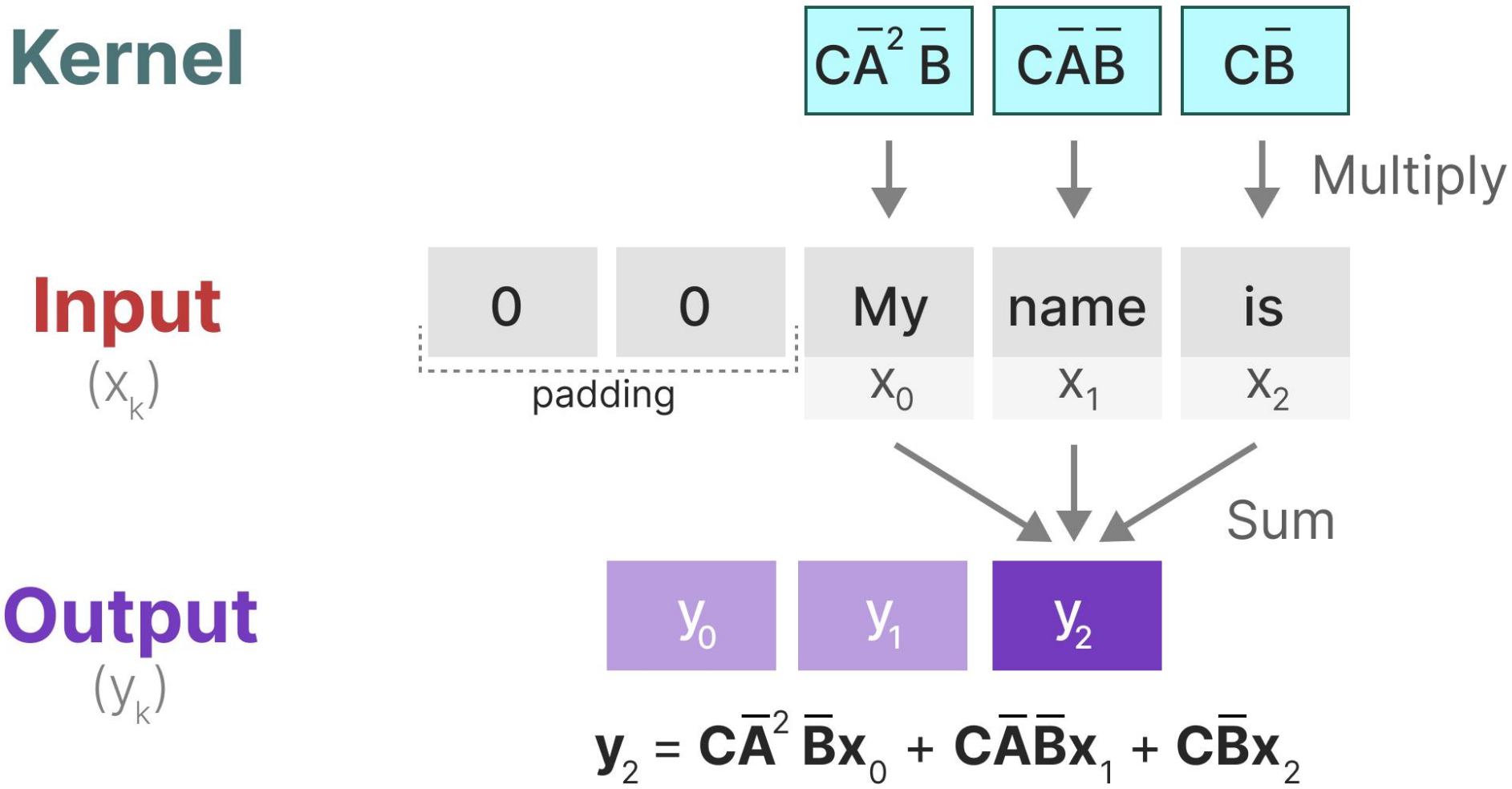
State Space Model : CNN view



State Space Model : CNN view



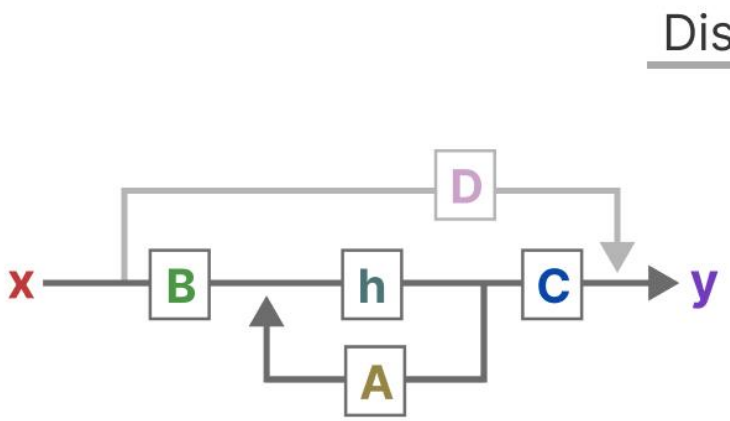
State Space Model : CNN view



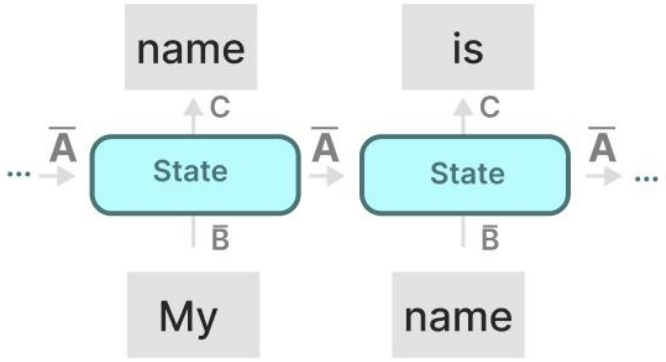
State Space Model : CNN view

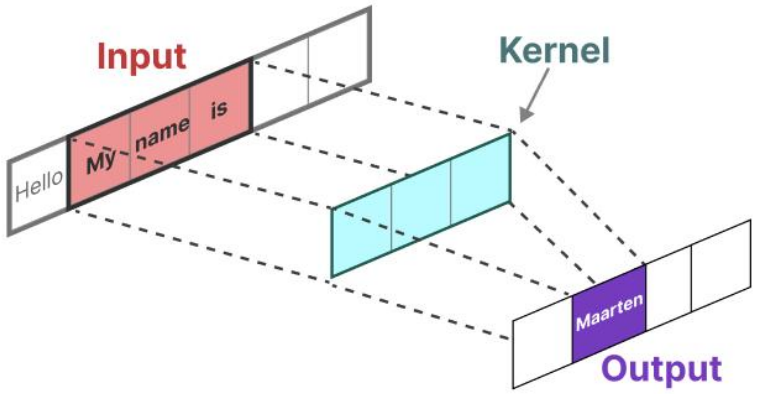
A major benefit of representing the SSM as a convolution is that it can be **trained in parallel like Convolutional Neural Networks (CNNs)**. However, due to the **fixed kernel size**, their inference is **not as fast** and **unbounded as RNNs**.

Continuous-time | **Recurrent** or **Convolutional**



Discretize

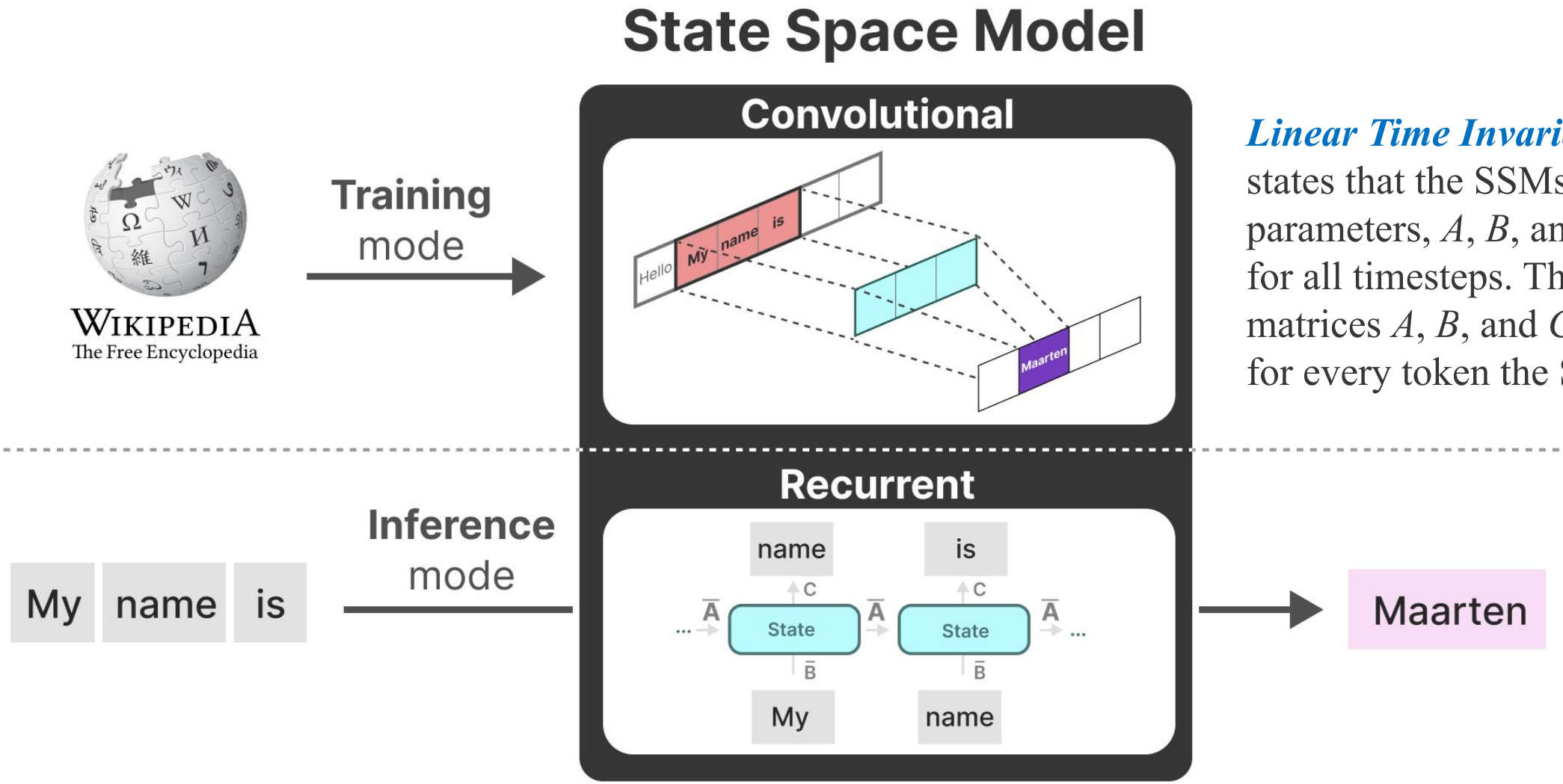




✔ *efficient inference*
✘ *parallelizable training*

✘ *unbounded context*
✔ *parallelizable training*

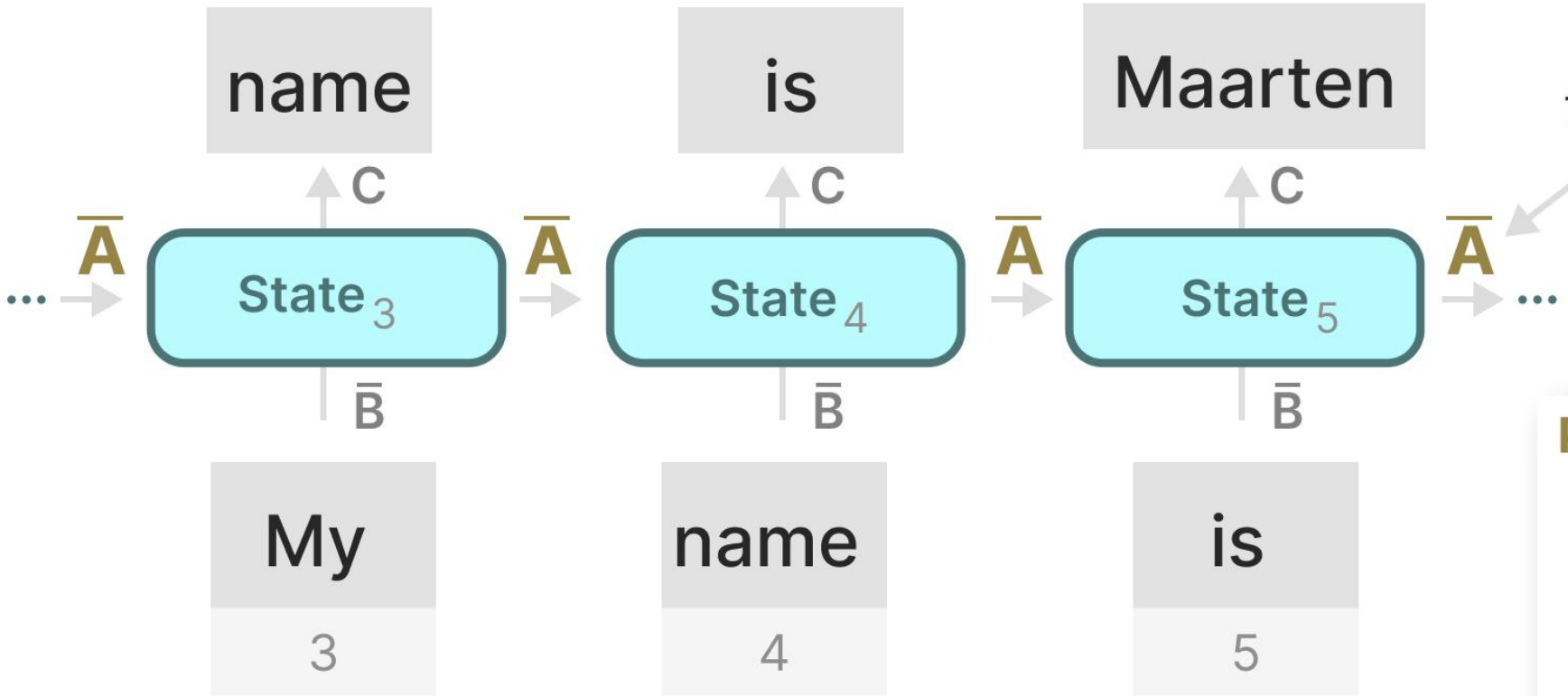
State Space Model : CNN/RNN mode



***Linear Time Invariance* (LTI)**. LTI states that the SSMs parameters, A, B, and C, are fixed for all timesteps. This means that matrices A, B, and C are the same for every token the SSM generates.

Gu, Albert, et al. **"Combining recurrent, convolutional, and continuous-time models with linear state space layers."** *Advances in neural information processing systems* 34 (2021): 572-585.

State Space Model : The Importance of *Matrix A*



SSM
(Recurrent + Unfolded)

Produces hidden state

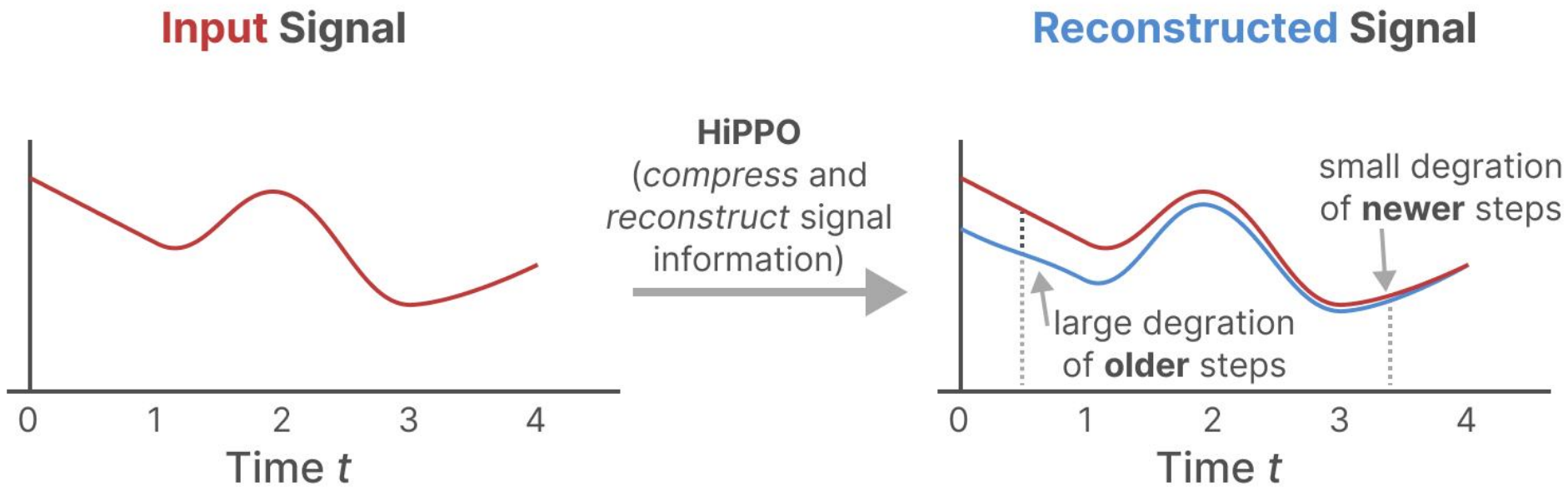
$$h_k = \bar{A}h_{k-1} + \bar{B}x_k$$

$$y_k = Ch_k$$

State Space Model : The Importance of *Matrix A*

Building matrix A using HiPPO was shown to be much better than initializing it as a random matrix. As a result, it more accurately reconstructs newer signals (recent tokens) compared to older signals (initial tokens).

The idea behind the HiPPO Matrix is that **it produces a hidden state that memorizes its history.**



State Space Model : The Importance of *Matrix A*

Building matrix A using HiPPO was shown to be much better than initializing it as a random matrix. As a result, it more accurately reconstructs newer signals (recent tokens) compared to older signals (initial tokens).

The idea behind the HiPPO Matrix is that **it produces a hidden state that memorizes its history.**

$$A_{nk} \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{everything below the diagonal} \\ n+1 & \text{the diagonal} \\ 0 & \text{everything above the diagonal} \end{cases}$$

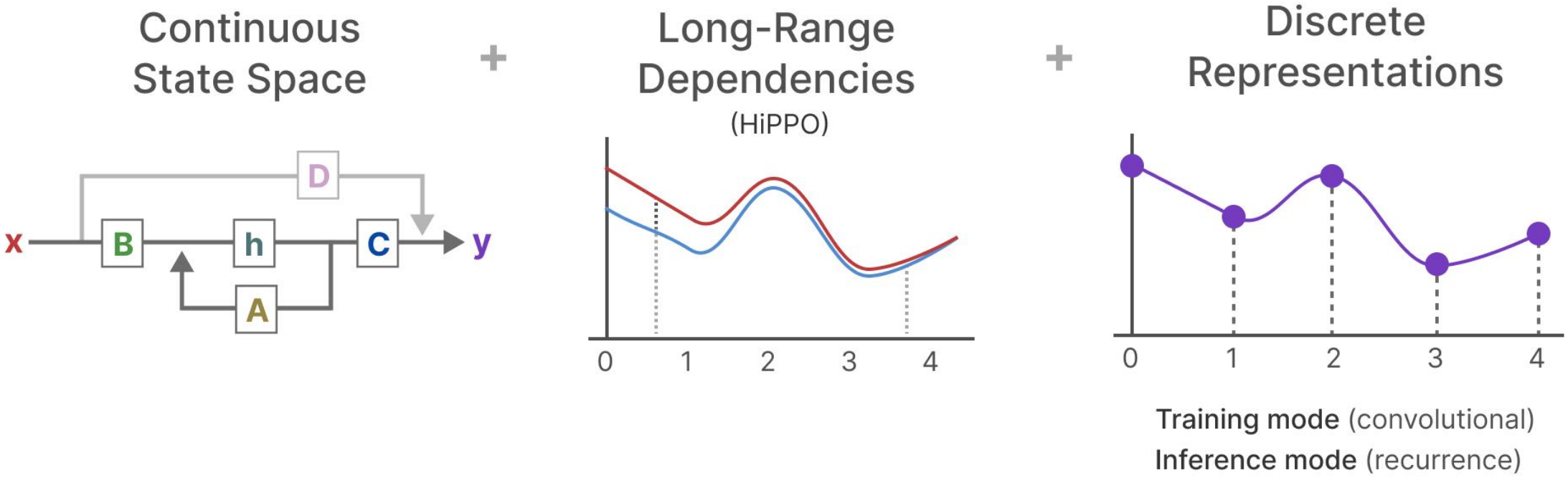
HiPPO Matrix

1	0	0	0
1	2	0	0
1	3	3	0
1	3	5	4

k (rows) × *n* (columns)

State Space Model : The Importance of *Matrix A*

- **Structured State Space for Sequences (S4)**

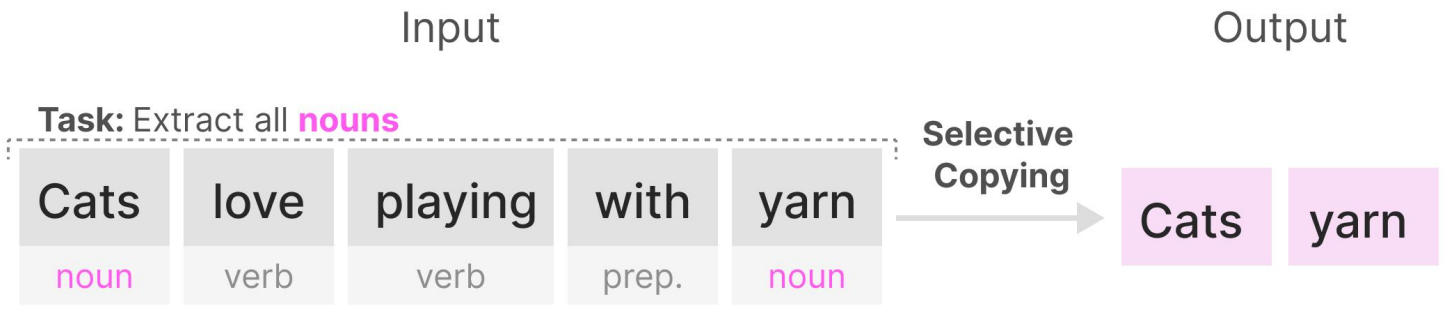


Efficiently Modeling Long Sequences with Structured State Spaces, arXiv:2111.00396, Albert Gu, Karan Goel, Christopher Ré

State Space Model : *Mamba* - A Selective SSM

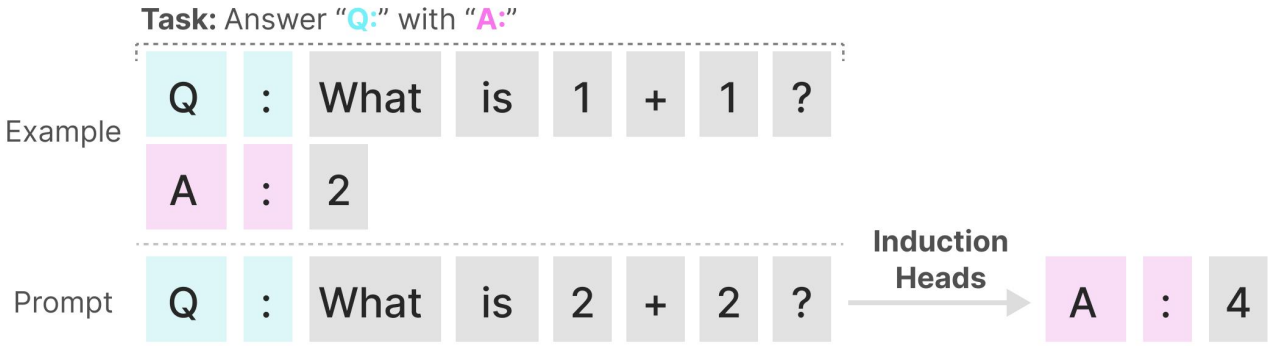
State Space Models, and even the S4 (Structured State Space Model), perform poorly on certain tasks that are vital in language modeling and generation, namely **the ability to focus on or ignore particular inputs.**

- **Selective Copying**



SSM cannot perform content-aware reasoning since it treats each token equally as a result of the fixed A, B, and C matrices.

- **Induction Heads**



SSM cannot select which previous tokens to recall from its history.

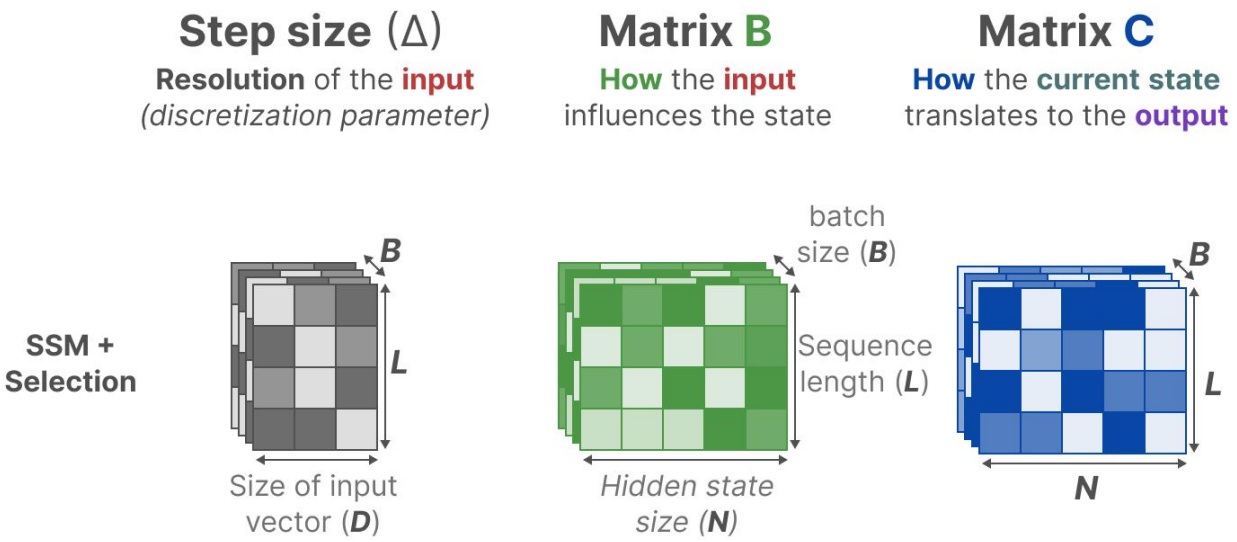
State Space Model : *Mamba* - A Selective SSM

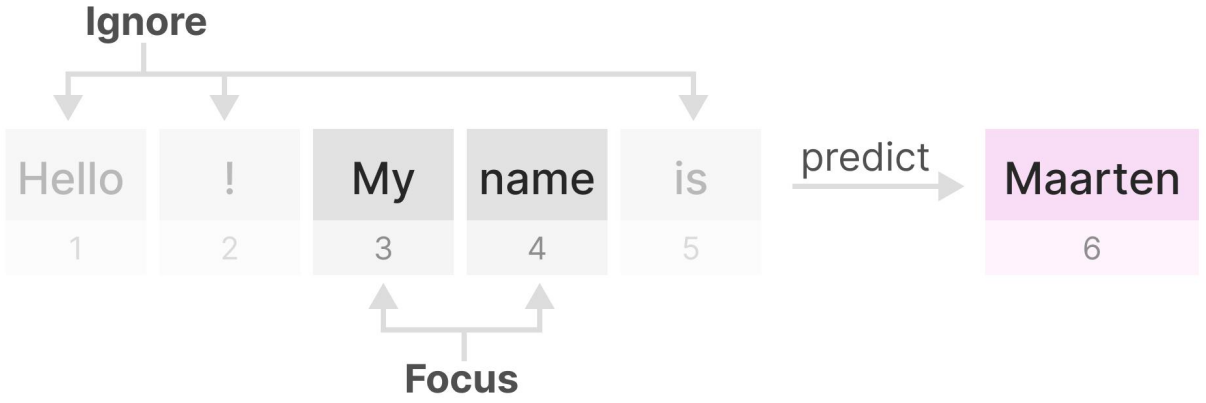
- **From time invariant to dependent**

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.

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.
Matrix *A* remains the same.

A smaller step size Δ results in ignoring specific words and instead using the previous context more whilst a larger step size Δ focuses on the input words more than the context.

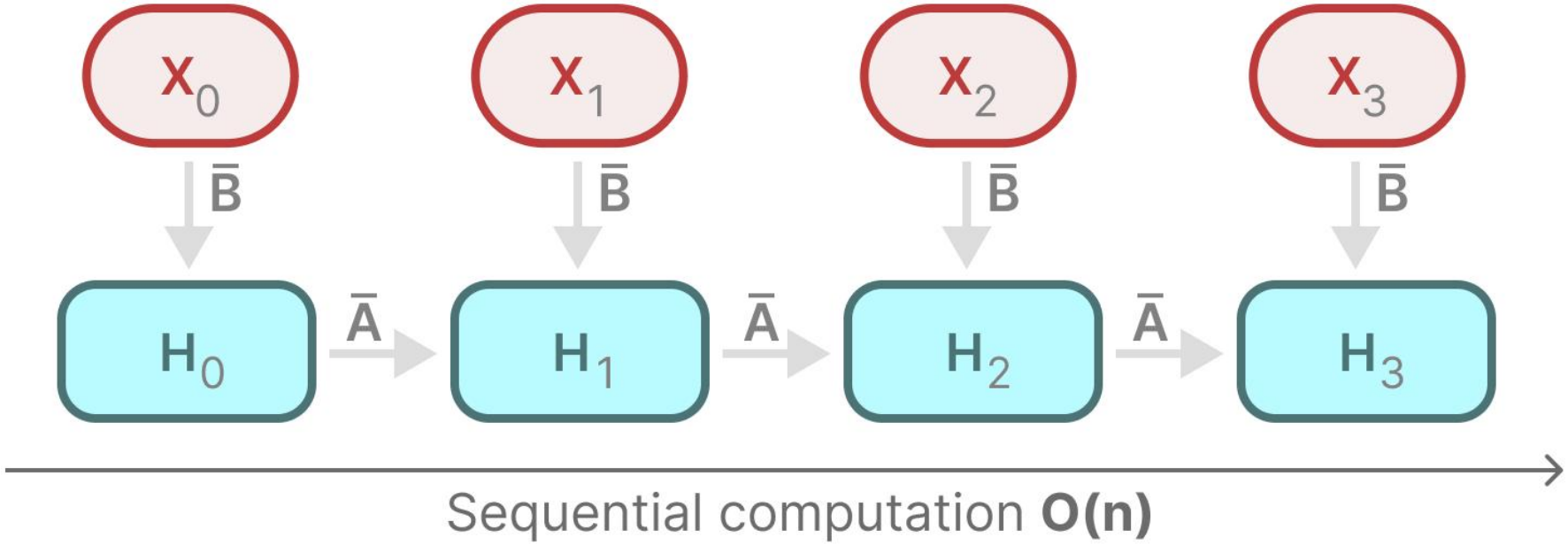




State Space Model : *Mamba* - A Selective SSM

- **From time invariant to dependent**

  Convolution can't be used now, as these matrixes are dynamic but the convolutional kernel is fixed.



Each state is the sum of the previous state (multiplied by A) plus the current input (multiplied by B). This is called a ***scan operation*** and can easily be calculated with a for loop.

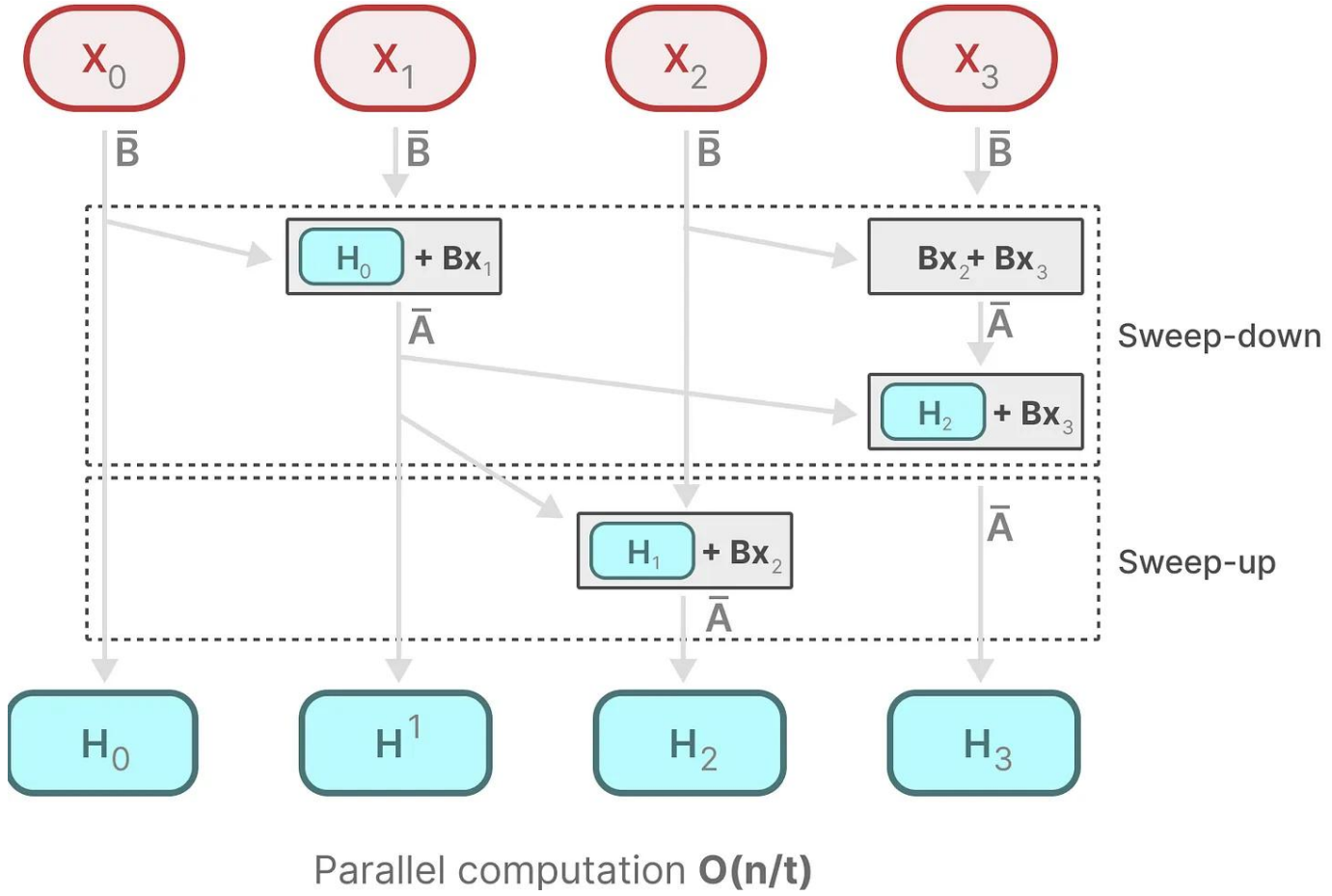
State Space Model : *Mamba* - A Selective SSM

- **From time invariant to dependent**

Parallel scan algorithm adopted in Mamba make this possible.

It assumes the order in which we do operations does not matter through the associate property. As a result, we can calculate the sequences in parts and iteratively combine them.

Thus, dynamic matrices *B* and *C*, and the parallel scan algorithm create the ***selective scan algorithm*** to represent the dynamic and fast nature of using the recurrent representation.



State Space Model : *Mamba* - A Selective SSM

- **From time invariant to dependent**

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

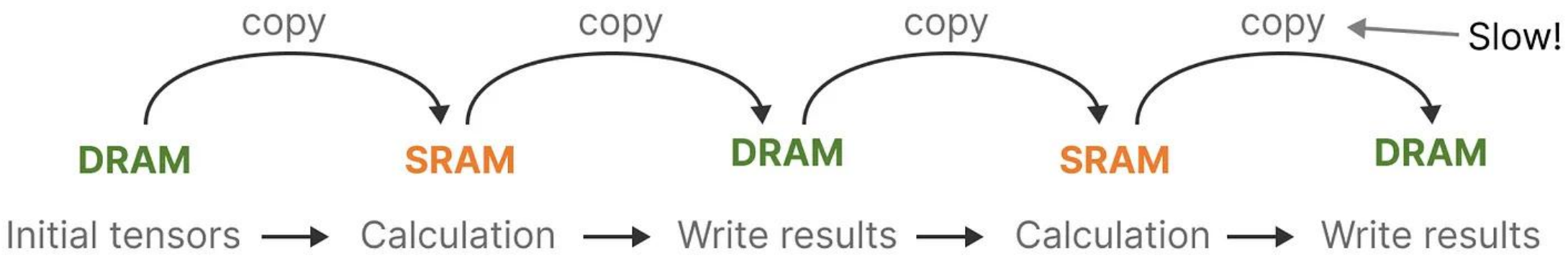
Algorithm 2 SSM + Selection (S6)

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

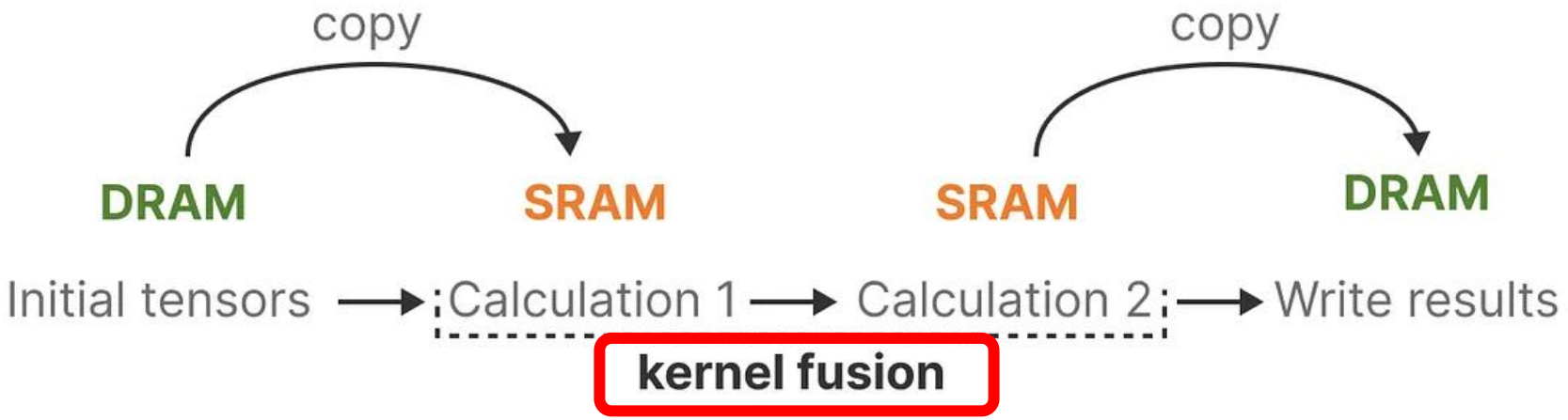
$$s_B(x) = \text{Linear}_N(x),\ s_C(x) = \text{Linear}_N(x),\ s_\Delta(x) = \text{Broadcast}_D(\text{Linear}_1(x)),\ \text{and}\ \tau_\Delta = \text{softplus},$$

State Space Model : *Mamba* - A Selective SSM

- **Hardware-aware Algorithm**



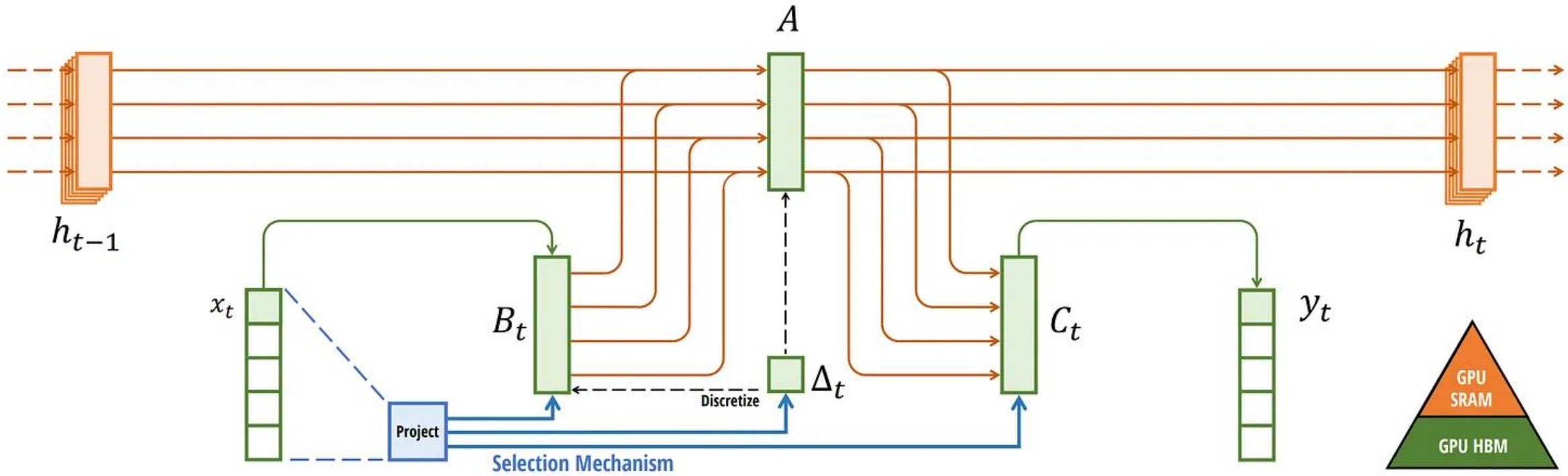
➢ **kernel fusion** used in Mamba allows the model to **prevent writing intermediate results** and **continuously performing computations until it is done**.



State Space Model : *Mamba* - A Selective SSM

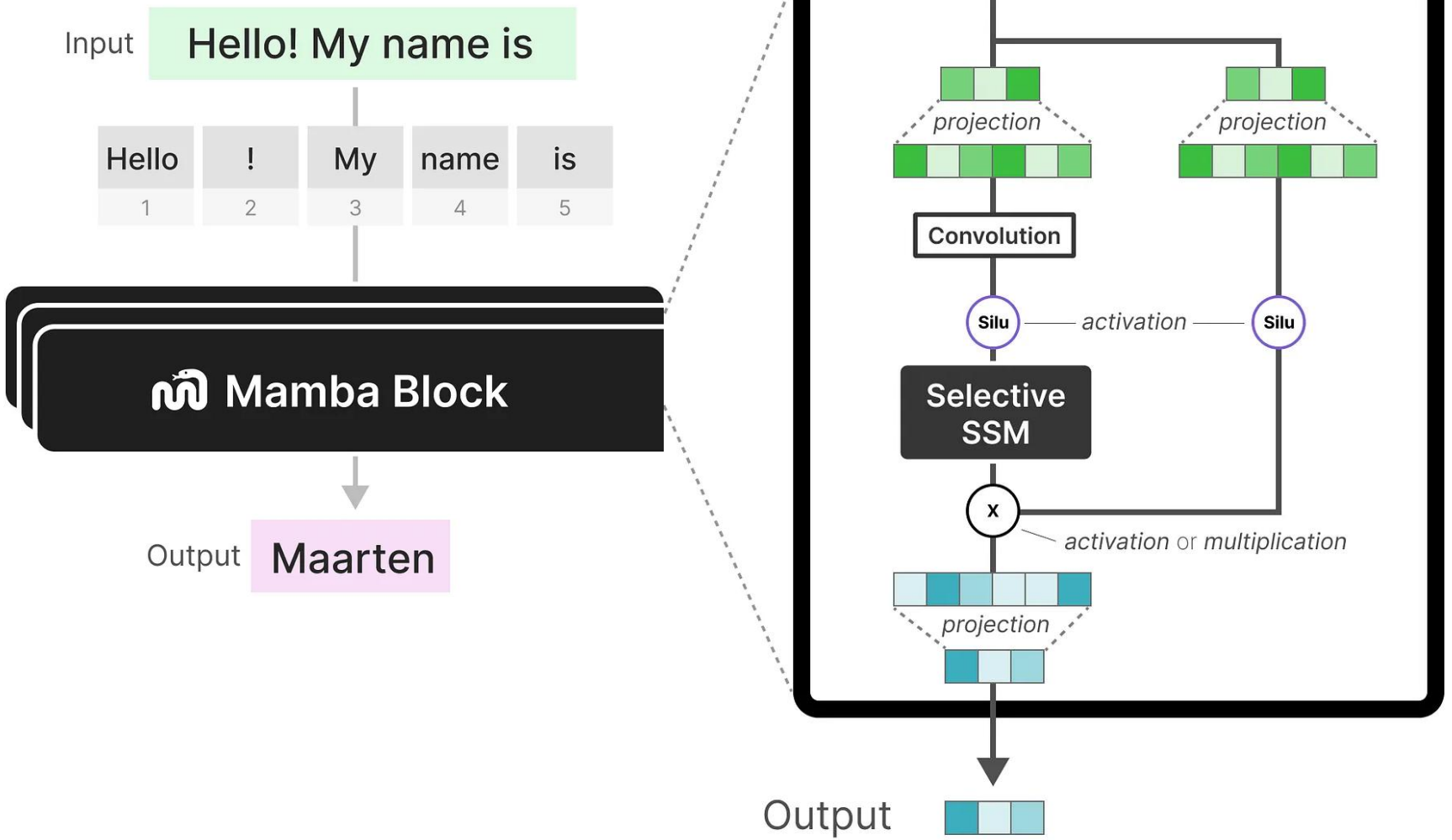
- **Hardware-aware Algorithm**

We can view the specific instances of DRAM and SRAM allocation by visualizing Mamba's base architecture:



State Space Model : *Mamba* - A Selective SSM

- **Mamba Block**



State Space Model : *Mamba* - A Selective SSM

- **Mamba Block**

The Selective SSM has the following properties:

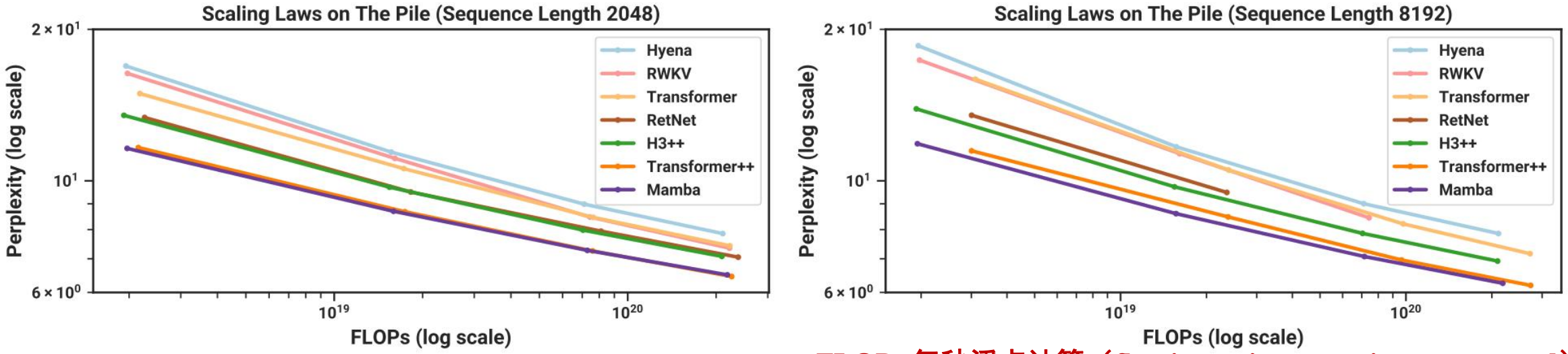
- Recurrent SSM created through discretization
- HiPPO initialization on matrix A to capture long-range dependencies
- Selective scan algorithm to selectively compress information
- Hardware-aware algorithm to speed up computation

	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 + **unbounded** context)

State Space Model： *Mamba* - A Selective SSM

☐ Scaling Laws (From about 125M to about1.3B parameters)

困惑度(perplexity)



FLOPs 每秒浮点计算（floating-point operations per second）

Figure 4: (**Scaling Laws**.) Models of size ≈ 125M to ≈ 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.

Summary about SSM

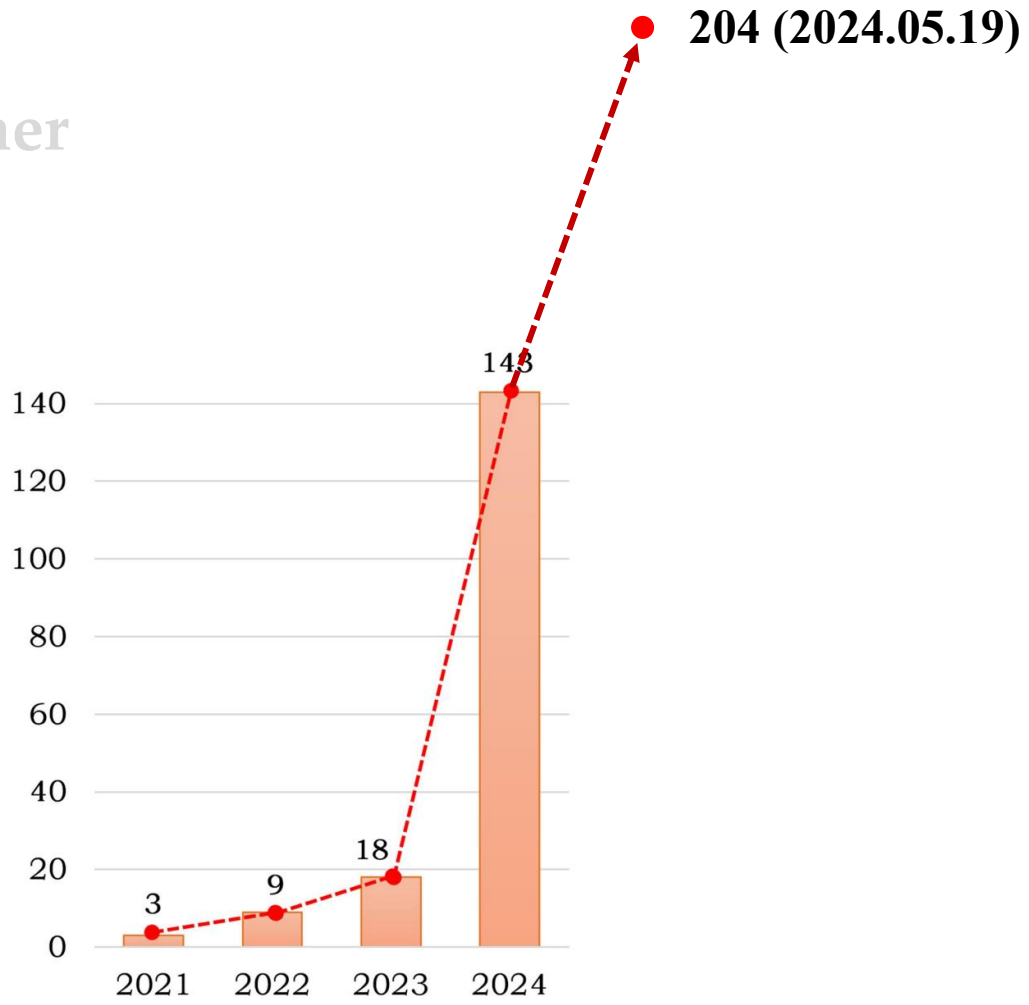
- ✓ From Continuous SSM to Discrete SSM
- ✓ CNN/RNN mode: Linear State-Space Layer (LSSL)
- ✓ Linear Time Invariance
- ✓ S4 model
- ✓ S6 model *(Mamba)*

Overview

- Review: CNN, RNN, Transformer
- **State Space Model**
  - Model Formulation
  - **Applications**
- Future Works
- Discussion

Overview

- Review: CNN, RNN, Transformer
- **State Space Model**
  - Model Formulation
  - **Applications**
- Future Works
- Discussion



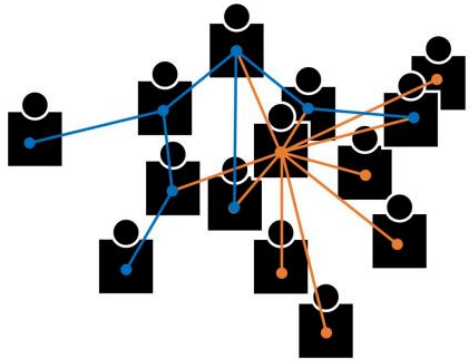
Number of papers released to date
(from year 2021 to year 2024.04)

State Space Model / Mamba : *Applications*

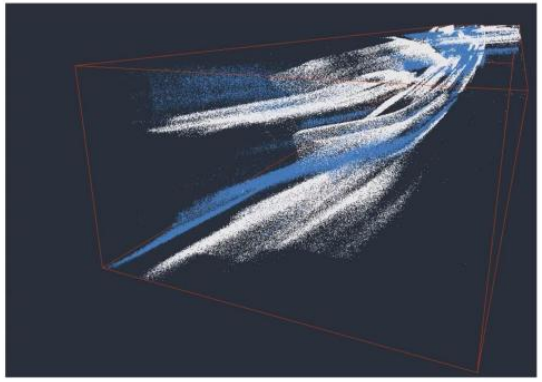
(a). Vision data

1977年，安徽大学数学系设立“计算技术专业”，1984年在“计算技术专业”基础上成立“计算机科学系”，开办“计算机软件”专业。1987年，“计算机应用专业”招收专科生，1988年开始招收本科生。2002年，“计算机应用技术”学科被教育部批准为国家级重点学科。2004年，计算机科学与技术学院正式成立。砥砺奋进谱华章，伴随着我国改革开放和现代化建设的伟大进程,尤其是近年来依靠安徽大学“211工程建设”与“世界一流学科”建设的快速发展，学院也迎来了历史发展的最好时期。

(b). Text data



(c). Graph data

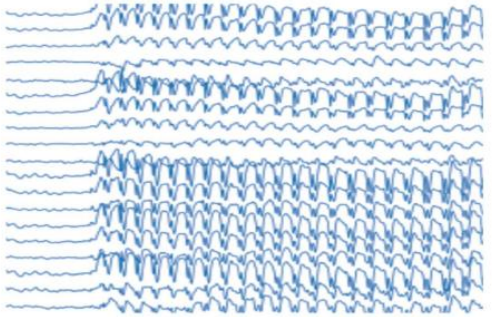


(d). Event/Point data

FINDINGS: No change. No visible active cardiopulmonary disease. Both lungs remain clear and expanded. Heart and pulmonary XXXX are normal. No change in the large hiatus hernia.

(e). Multi-modal/Multi-media data

(f). Audio/Speech data



(g). Time series data

Sample Dataset

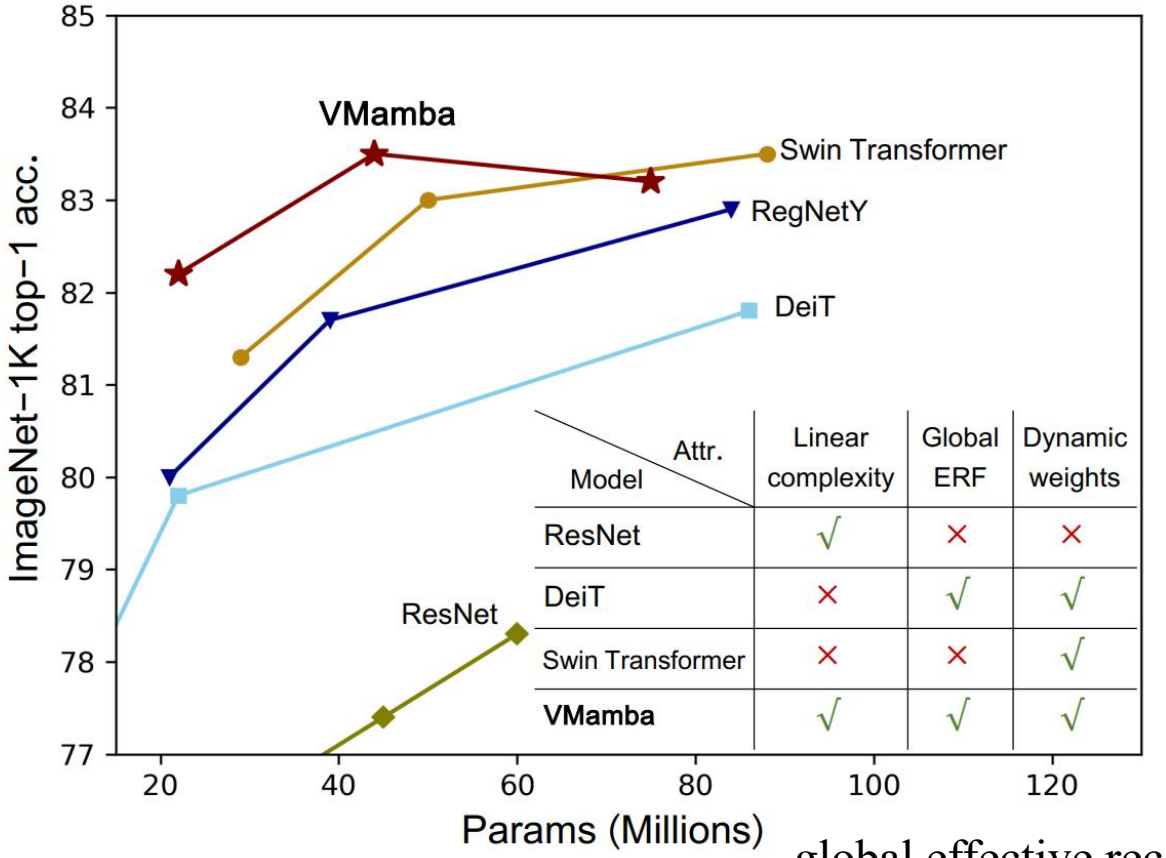
f_1	f_2	f_3	f_4	⋯	f_n	Target
$v_{1,1}$	$v_{1,2}$	$v_{1,3}$	$v_{1,4}$	⋯	$v_{1,n}$	y_1
$v_{2,1}$	$v_{2,2}$	$v_{2,3}$	$v_{2,4}$	⋯	$v_{2,n}$	y_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$v_{m,1}$	$v_{m,2}$	$v_{m,3}$	$v_{m,4}$	⋯	$v_{m,n}$	y_m

(h). Tabular data

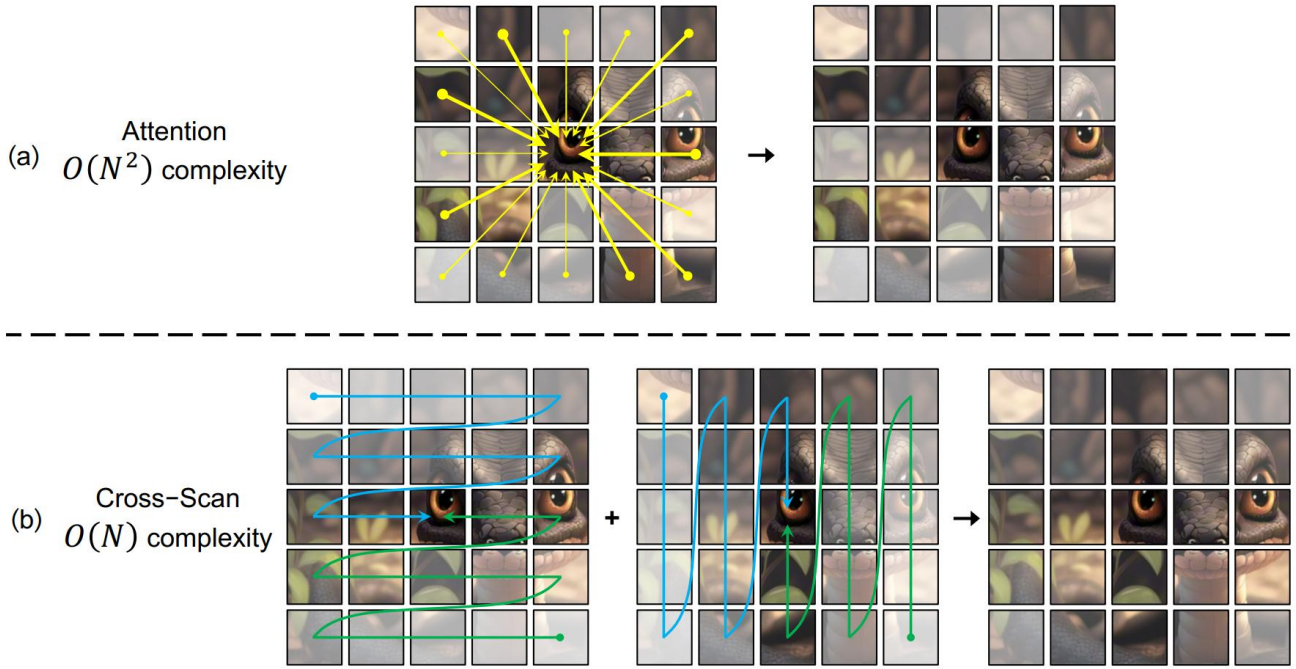
State Space Model / Mamba : *Applications*

VMamba: Visual State Space Model, *https://arxiv.org/abs/2401.10166*

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Yunfan Liu



Model \ Attr.	Linear complexity	Global ERF	Dynamic weights
ResNet	√	×	×
DeiT	×	√	√
Swin Transformer	×	×	√
VMamba	√	√	√



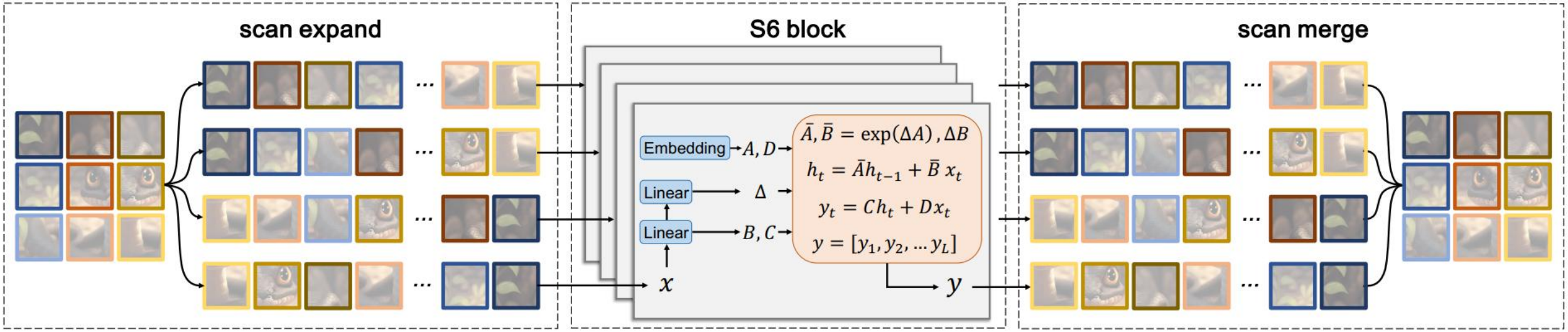
global effective reception field (ERF)

Performance comparison on ImageNet-1K dataset.

State Space Model / Mamba : *Applications*

VMamba: Visual State Space Model, *https://arxiv.org/abs/2401.10166*

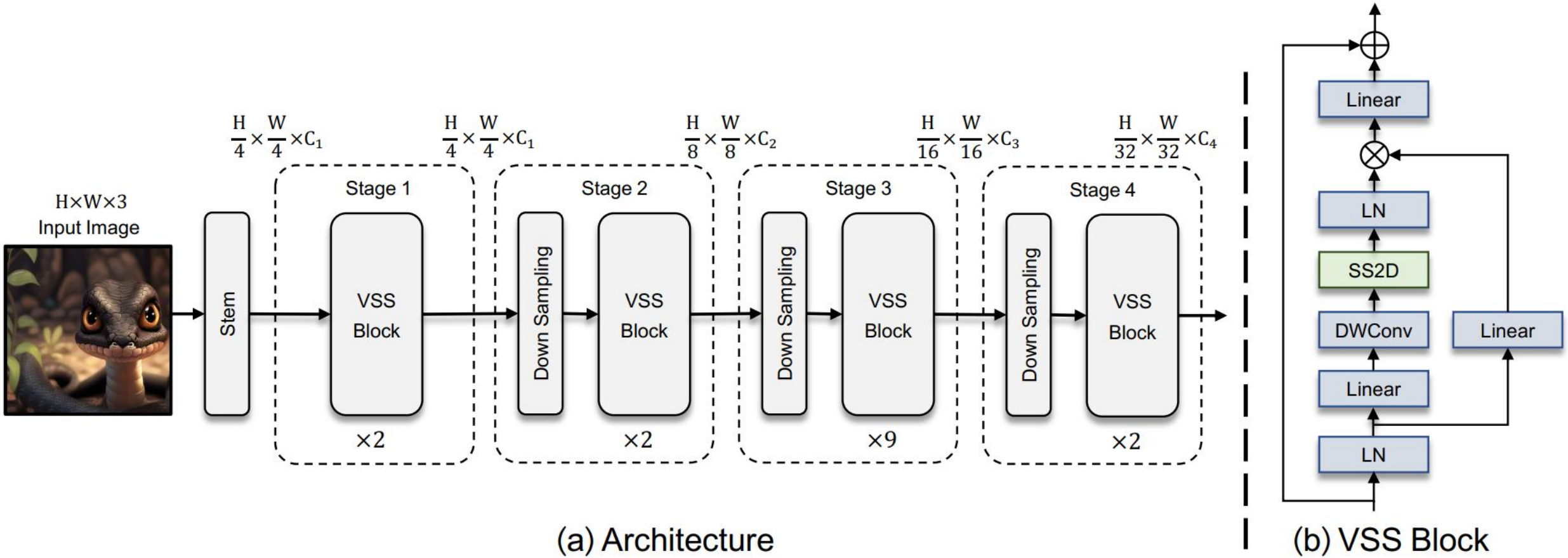
Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Yunfan Liu



State Space Model / Mamba : *Applications*

VMamba: Visual State Space Model, *https://arxiv.org/abs/2401.10166*

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Yunfan Liu



State Space Model / Mamba : *Applications*

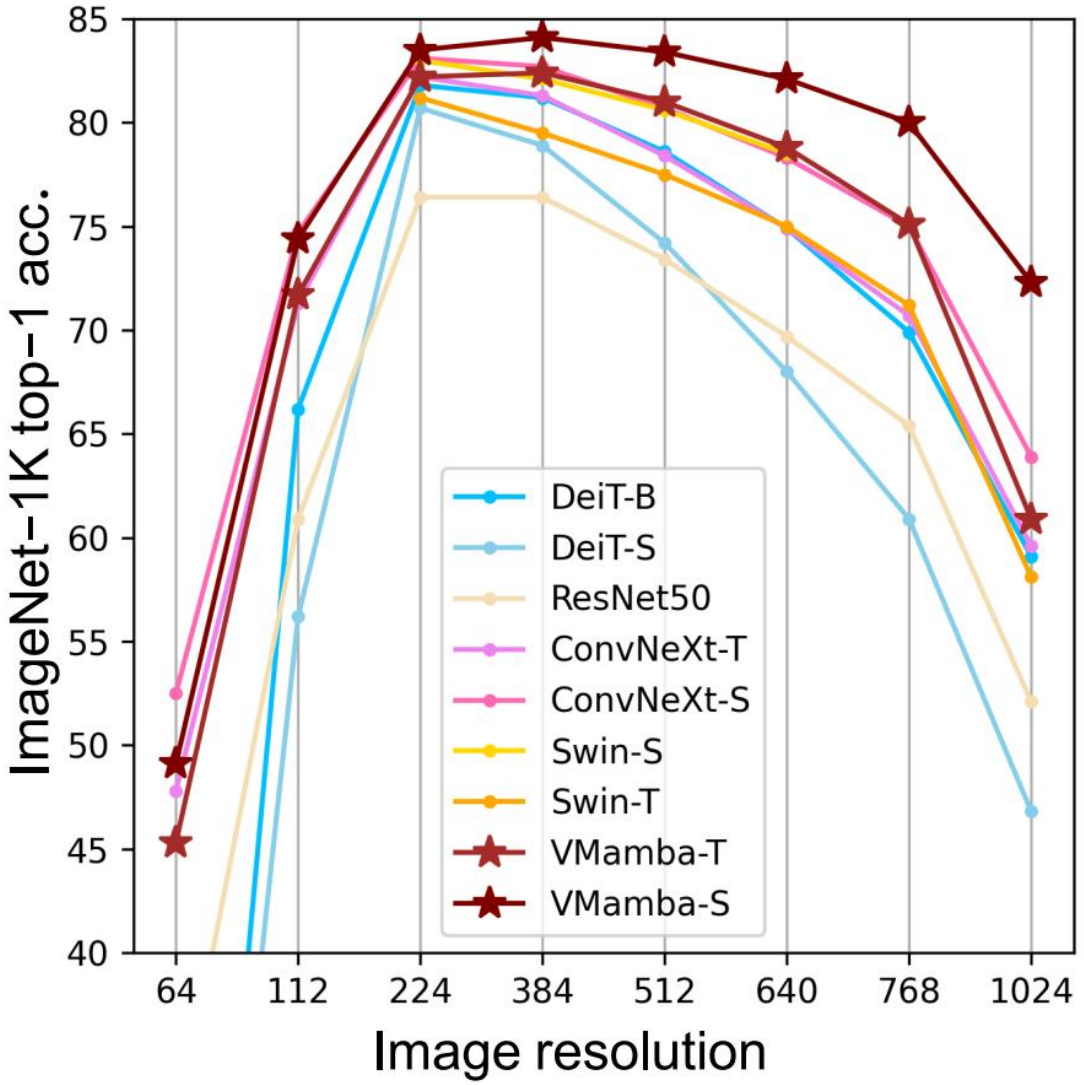
Method	Image size	#Param.	FLOPs	Throughput	Train Throughput	ImageNet top-1 acc.
RegNetY-4G [41]	224^2	21M	4.0G	–	–	80.0
RegNetY-8G [41]	224^2	39M	8.0G	–	–	81.7
RegNetY-16G [41]	224^2	84M	16.0G	–	–	82.9
EffNet-B3 [47]	300^2	12M	1.8G	–	–	81.6
EffNet-B4 [47]	380^2	19M	4.2G	–	–	82.9
EffNet-B5 [47]	456^2	30M	9.9G	–	–	83.6
EffNet-B6 [47]	528^2	43M	19.0G	–	–	84.0
ViT-B/16 [12]	384^2	86M	55.4G	–	–	77.9
ViT-L/16 [12]	384^2	307M	190.7G	–	–	76.5
DeiT-S [50]	224^2	22M	4.6G	1759	2397	79.8
DeiT-B [50]	224^2	86M	17.5G	500	1024	81.8
DeiT-B [50]	384^2	86M	55.4G	498	344	83.1
ConvNeXt-T [33]	224^2	29M	4.5G	1189	701	82.1
ConvNeXt-S [33]	224^2	50M	8.7G	682	444	83.1
ConvNeXt-B [33]	224^2	89M	15.4G	435	334	83.8
HiViT-T [64]	224^2	19M	4.6G	1391	1300	82.1
HiViT-S [64]	224^2	38M	9.1G	711	697	83.5
HiViT-B [64]	224^2	66M	15.9G	456	541	83.8
Swin-T [32]	224^2	28M	4.6G	1247	985	81.3
Swin-S [32]	224^2	50M	8.7G	719	640	83.0
Swin-B [32]	224^2	88M	15.4G	457	494	83.5
S4ND-ConvNeXt-T [40]	224^2	30M	-	684	331	82.2
S4ND-ViT-B [40]	224^2	89M	-	404	340	80.4
ViM-S [68]	224^2	26M	-	811	232†	80.5
VMamba-T	224^2	31M	4.9G	1335	464	82.5
VMamba-S	224^2	50M	8.7G	874	313	83.6
VMamba-B	224^2	89M	15.4G	645	246	83.9

Table 1: **Performance comparison on ImageNet-1K**. Throughput values are measured with an A100 GPU and an AMD EPYC 7542 CPU, using the toolkit released by [56], following the protocol proposed in [32]. † denotes that the training process only supports float32 datatype.

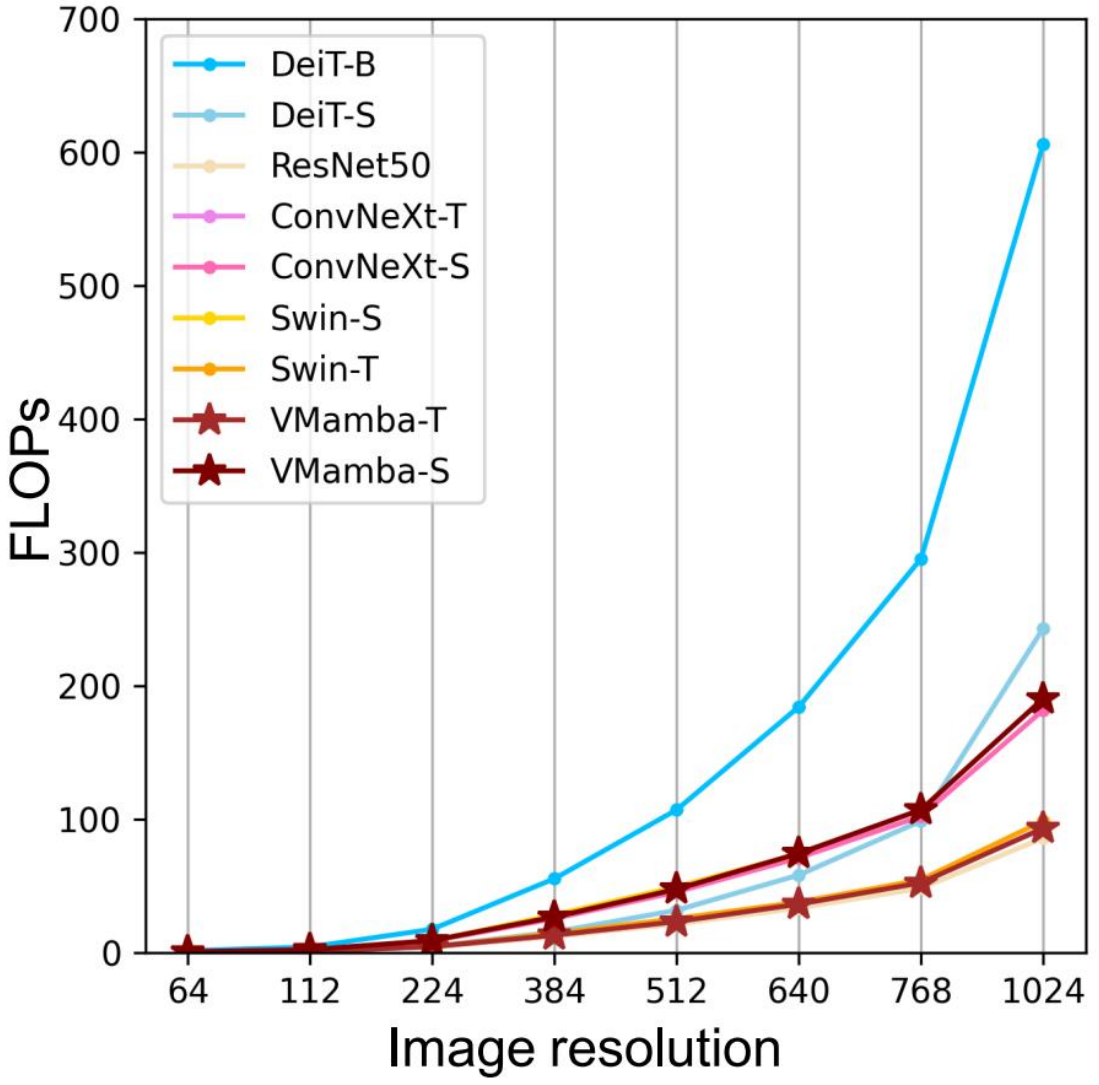
Mask R-CNN 1× schedule								
Backbone	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	#param.	FLOPs
ResNet-50	38.2	58.8	41.4	34.7	55.7	37.2	44M	260G
Swin-T	42.7	65.2	46.8	39.3	62.2	42.2	48M	267G
ConvNeXt-T	44.2	66.6	48.3	40.1	63.3	42.8	48M	262G
PVTv2-B2	45.3	67.1	49.6	41.2	64.2	44.4	45M	309G
ViT-Adapter-S	44.7	65.8	48.3	39.9	62.5	42.8	48M	403G
VMamba-T	47.4	69.5	52.0	42.7	66.3	46.0	50M	270G
ResNet-101	38.2	58.8	41.4	34.7	55.7	37.2	63M	336G
Swin-S	44.8	66.6	48.9	40.9	63.2	44.2	69M	354G
ConvNeXt-S	45.4	67.9	50.0	41.8	65.2	45.1	70M	348G
PVTv2-B3	47.0	68.1	51.7	42.5	65.7	45.7	65M	397G
VMamba-S	48.7	70.0	53.4	43.7	67.3	47.0	64M	357G
Swin-B	46.9	–	–	42.3	–	–	107M	496G
ConvNeXt-B	47.0	69.4	51.7	42.7	66.3	46.0	108M	486G
PVTv2-B5	47.4	68.6	51.9	42.5	65.7	46.0	102M	557G
ViT-Adapter-B	47.0	68.2	51.4	41.8	65.1	44.9	102M	557G
VMamba-B	49.2	70.9	53.9	43.9	67.7	47.6	108M	485G

Object detection and instance segmentation on COCO dataset.

State Space Model / Mamba : *Applications*



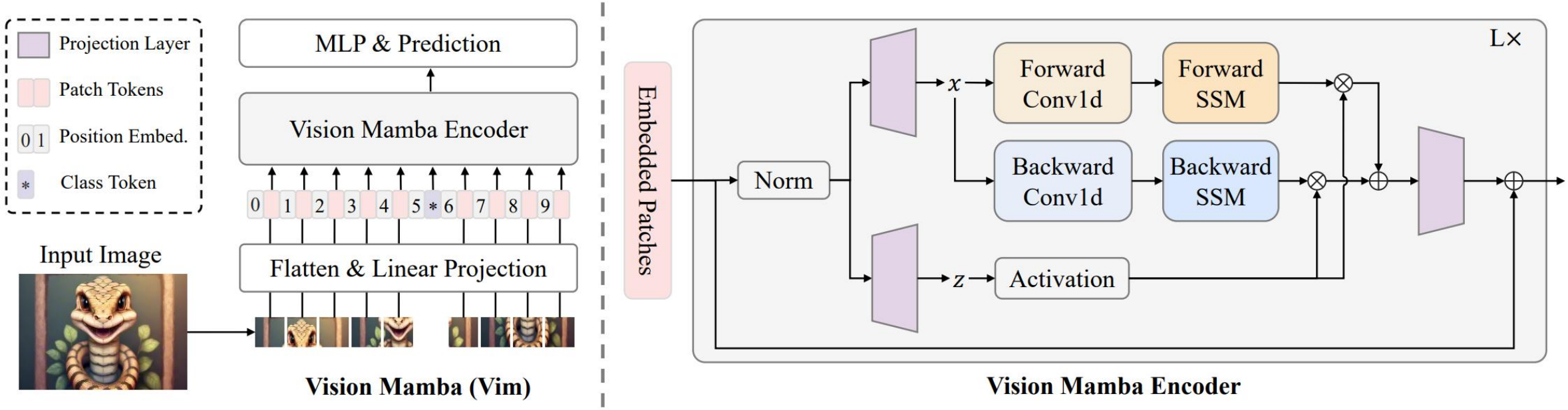
(a)



(b)

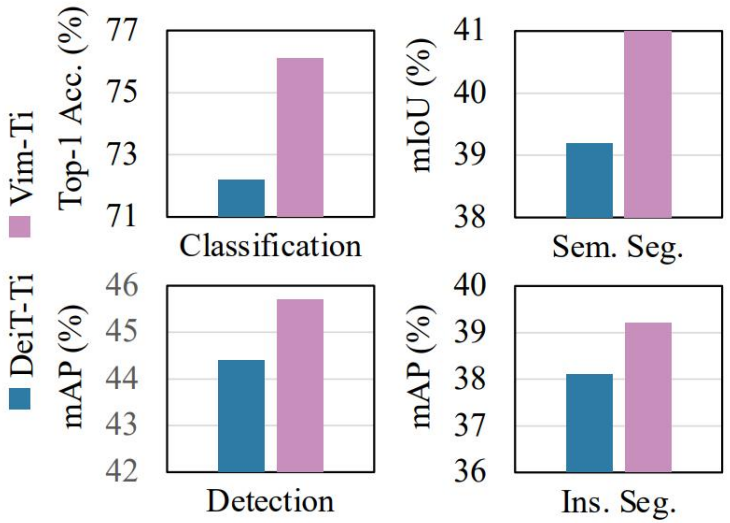
State Space Model / Mamba : *Applications*

Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model,
Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, Xinggang Wang, ICML 2024

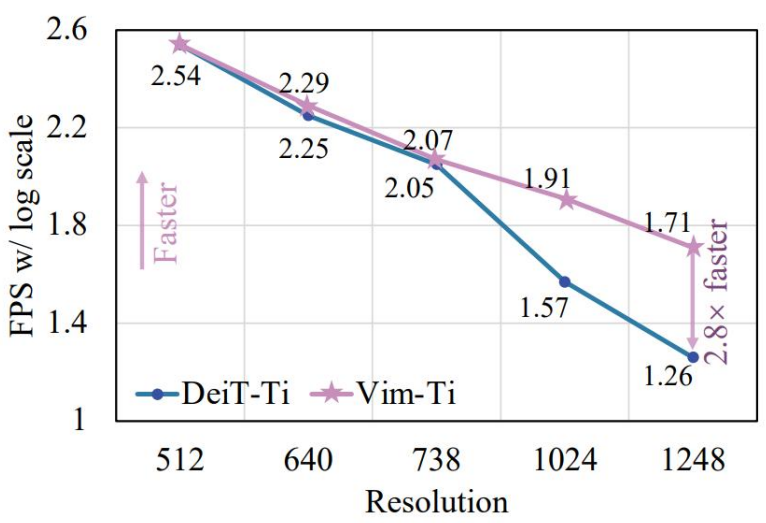


State Space Model / Mamba : *Applications*

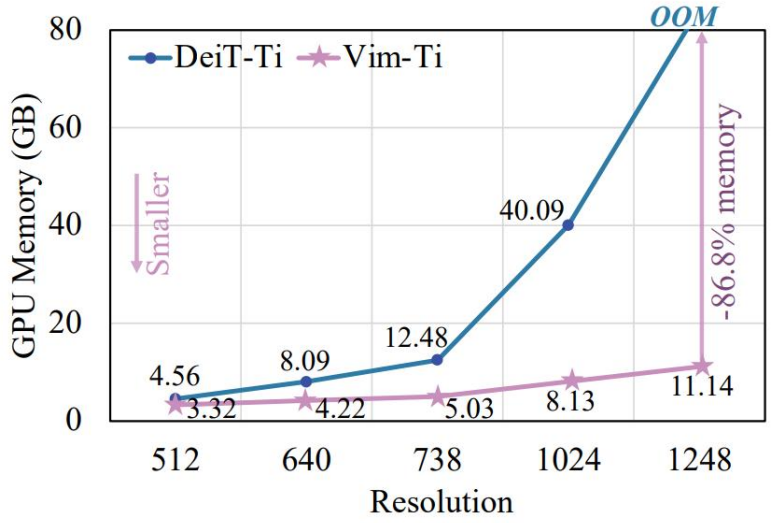
Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model,
Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, Xinggang Wang, ICML 2024



(a) Accuracy Comparison



(b) Speed Comparison



(c) GPU Memory Comparison

State Space Model / Mamba : *Applications*

Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model,
Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, Xinggang Wang, ICML 2024

Method	image size	#param.	ImageNet top-1 acc.
Convnets			
ResNet-18	224^2	12M	69.8
ResNet-50	224^2	25M	76.2
ResNet-101	224^2	45M	77.4
ResNet-152	224^2	60M	78.3
ResNeXt50-32×4d	224^2	25M	77.6
RegNetY-4GF	224^2	21M	80.0
Transformers			
ViT-B/16	384^2	86M	77.9
ViT-L/16	384^2	307M	76.5
DeiT-Ti	224^2	6M	72.2
DeiT-S	224^2	22M	79.8
DeiT-B	224^2	86M	81.8
SSMs			
S4ND-ViT-B	224^2	89M	80.4
Vim-Ti	224^2	7M	76.1
Vim-Ti†	224^2	7M	78.3 +2.2
Vim-S	224^2	26M	80.5
Vim-S†	224^2	26M	81.6 +1.1

Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{box}_{s}	AP^{box}_{m}	AP^{box}_{l}
DeiT-Ti	44.4	63.0	47.8	26.1	47.4	61.8
Vim-Ti	45.7	63.9	49.6	26.1	49.0	63.2
Backbone	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}_{s}	AP^{mask}_{m}	AP^{mask}_{l}
DeiT-Ti	38.1	59.9	40.5	18.1	40.5	58.4
Vim-Ti	39.2	60.9	41.7	18.2	41.8	60.2

Results of object detection and instance segmentation on the COCO val set using Cascade Mask R-CNN framework.

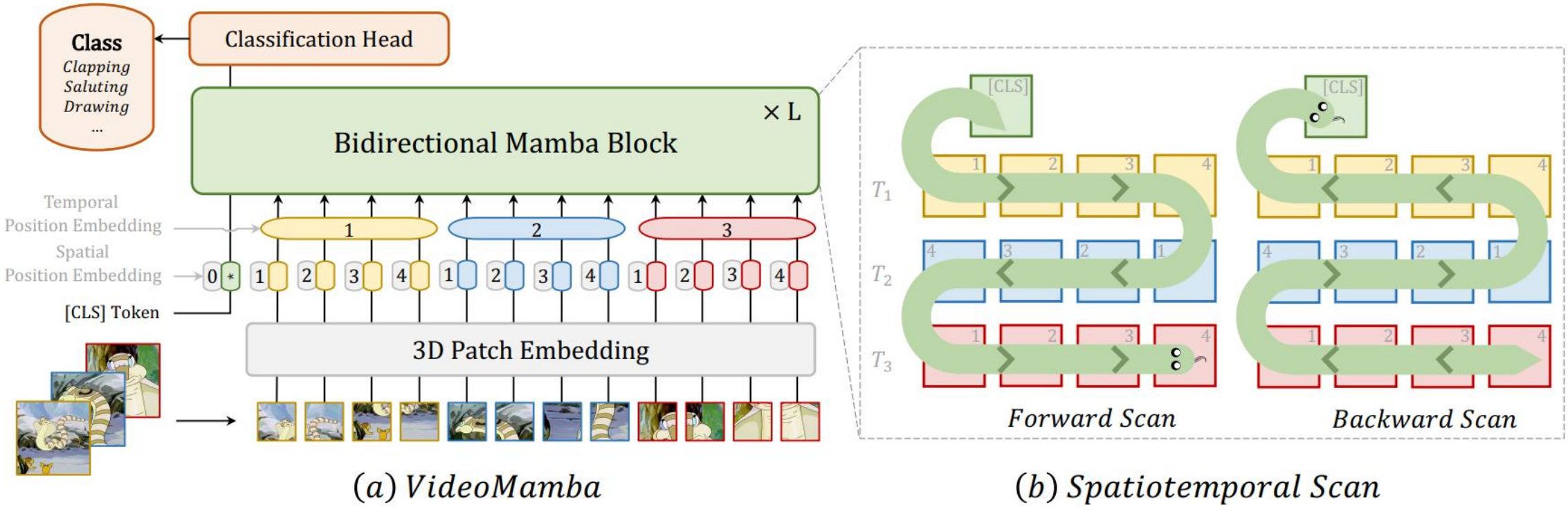
Method	Backbone	image size	#param.	*val* mIoU
DeepLab v3+	ResNet-101	512^2	63M	44.1
UperNet	ResNet-50	512^2	67M	41.2
UperNet	ResNet-101	512^2	86M	44.9
UperNet	DeiT-Ti	512^2	11M	39.2
UperNet	DeiT-S	512^2	43M	44.0
UperNet	Vim-Ti	512^2	13M	41.0
UperNet	Vim-S	512^2	46M	44.9

Results of semantic segmentation on the ADE20K val set.

State Space Model / Mamba : *Applications*

VideoMamba: State Space Model for Efficient Video Understanding,

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, Yu Qiao



(a) VideoMamba (b) Spatiotemporal Scan

State Space Model / Mamba : *Applications*

VideoMamba: State Space Model for Efficient Video Understanding,

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, Yu Qiao

Arch.	Model	*iso.*	Input Size	#Param (M)	FLOPs (G)	IN-1K Top-1
CNN	ConvNeXt-T [53]	✗	224^2	29	4.5	82.1
	ConvNeXt-S [53]	✗	224^2	50	8.7	83.1
	ConvNeXt-B [53]	✗	224^2	89	15.4	**83.8**
Trans.	SwinT-T [51]	✗	224^2	28	4.5	81.3
	Swin-S [51]	✗	224^2	50	8.7	83.0
	Swin-B [51]	✗	224^2	88	15.4	83.5
CNN+ SSM	VMamba-T [50]	✗	224^2	22	5.6	82.2
	VMamba-S [50]	✗	224^2	44	11.2	83.5
	VMamba-B [50]	✗	224^2	75	18.0	83.7
CNN	ConvNeXt-S [53]	✓	224^2	22	4.3	79.7
	ConvNeXt-B [53]	✓	224^2	87	16.9	82.0
Trans.	DeiT-Ti [75]	✓	224^2	6	1.3	72.2
	DeiT-S [75]	✓	224^2	22	4.6	79.8
	DeiT-B [75]	✓	224^2	87	17.6	81.8
	DeiT-B [75]	✓	384^2	87	55.5	83.1
SSM	S4ND-ViT-B [58]	✓	224^2	89	-	80.4
	Vim-Ti [91]	✓	224^2	7	1.1	76.1
	Vim-S [91]	✓	224^2	26	4.3	80.5
	VideoMamba-Ti	✓	224^2	7	1.1	76.9
	VideoMamba-Ti	✓	448^2	7	4.3	79.3
	VideoMamba-Ti	✓	576^2	7	7.1	79.6
	VideoMamba-S	✓	224^2	26	4.3	81.2
	VideoMamba-S	✓	448^2	26	16.9	83.2
	VideoMamba-S	✓	576^2	26	28.0	83.5
	VideoMamba-M	✓	224^2	74	12.7	82.8
	VideoMamba-M	✓	448^2	75	50.4	83.8
	VideoMamba-M	✓	576^2	75	83.1	**84.0**

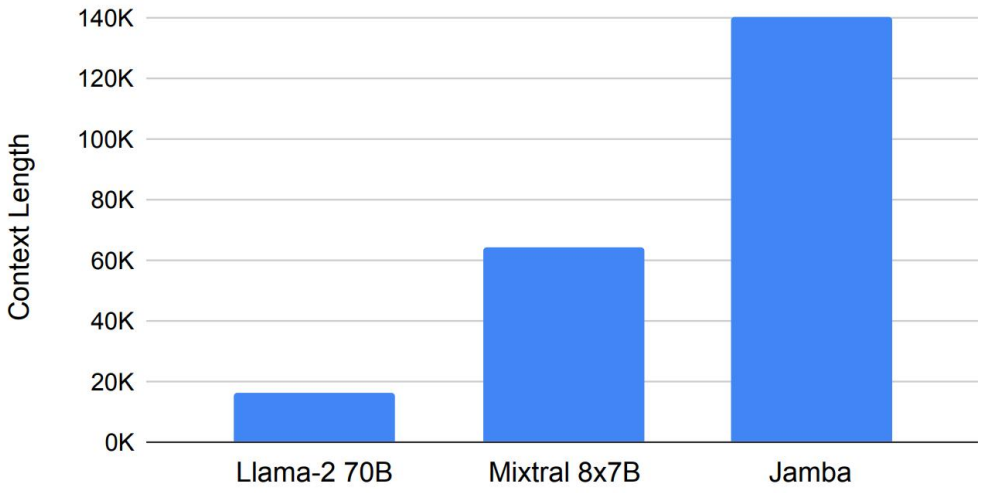
Arch.	Model	*iso.*	Extra Data	Input Size	#Param (M)	FLOPs (G)	K400 Top-1	K400 Top-5
Supervised: *Those models with extra data are under supervised training.*								
CNN	SlowFast$_{R101+NL}$ [19]	✗		80×224^2	60	234×3×10	79.8	93.9
	X3D-M [17]	✗		16×224^2	4	6×3×10	76.0	92.3
	X3D-XL [17]	✗		16×312^2	20	194×3×10	80.4	94.6
Trans.	Swin-T [52]	✗	IN-1K	32×224^2	28	88×3×4	78.8	93.6
	Swin-B [52]	✗	IN-1K	32×224^2	88	88×3×4	80.6	94.5
	Swin-B [52]	✗	IN-21K	32×224^2	88	282×3×4	82.7	**95.5**
CNN+ Trans.	MViTv1-B [16]	✗		32×224^2	37	70×1×5	80.2	94.4
	MViTv2-S [45]	✗		16×224^2	35	64×1×5	81.0	94.6
	UniFormer-S [44]	✗	IN-1K	16×224^2	21	42×1×4	80.8	94.7
	UniFormer-B [44]	✗	IN-1K	16×224^2	50	97×1×4	82.0	95.1
	UniFormer-B [44]	✗	IN-1K	32×224^2	50	259×3×4	**83.0**	95.4
Trans.	STAM [63]	✓	IN-21K	64×224^2	121	1040×1×1	79.2	-
	TimeSformer-L [4]	✓	IN-21K	96×224^2	121	2380×3×1	80.7	94.7
	ViViT-L [2]	✓	IN-21K	16×224^2	311	3992×3×4	81.3	94.7
	Mformer-HR [59]	✓	IN-21K	16×336^2	311	959×3×10	81.1	95.2
SSM	VideoMamba-Ti	✓	IN-1K	16×224^2	7	17×3×4	78.1	93.5
	VideoMamba-Ti	✓	IN-1K	32×224^2	7	34×3×4	78.8	93.9
	VideoMamba-Ti	✓	IN-1K	64×384^2	7	202×3×4	80.3	94.8
	VideoMamba-S	✓	IN-1K	16×224^2	26	68×3×4	80.8	94.8
	VideoMamba-S	✓	IN-1K	32×224^2	26	135×3×4	81.5	95.2
	VideoMamba-S	✓	IN-1K	64×384^2	26	395×3×4	82.7	95.6
	VideoMamba-M	✓	IN-1K	16×224^2	74	202×3×4	81.9	95.4
	VideoMamba-M	✓	IN-1K	32×224^2	74	403×3×4	82.4	95.7
	VideoMamba-M	✓	IN-1K	64×384^2	74	2368×3×4	**83.3**	**96.1**

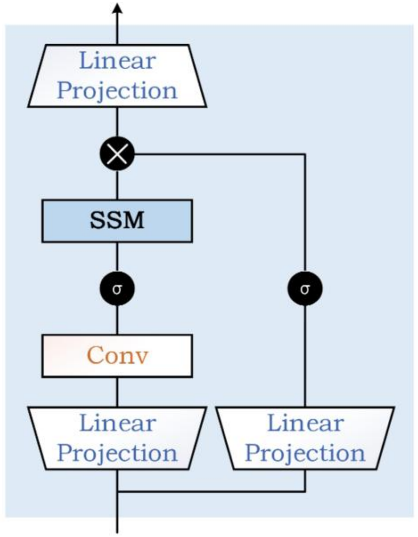
State Space Model / Mamba : *Applications*

Jamba: A Hybrid Transformer-Mamba Language Model

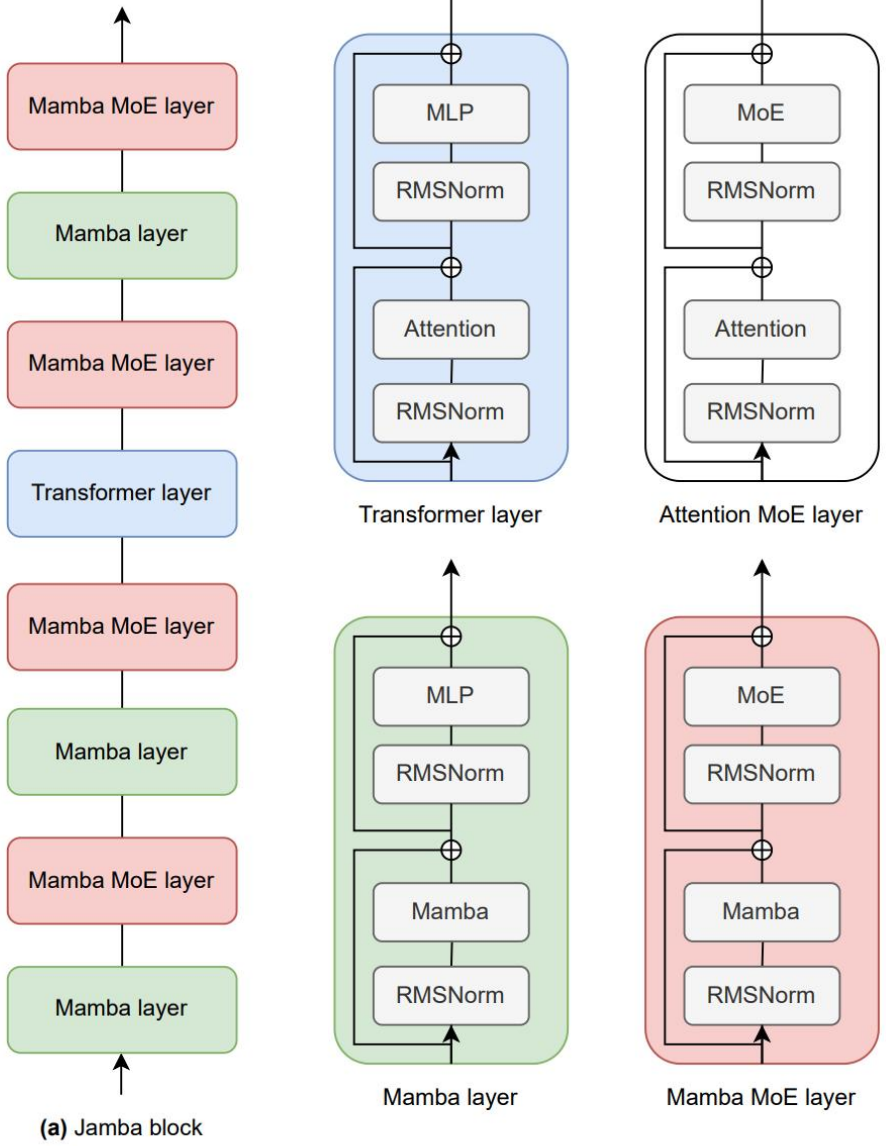
	Available params	Active params	KV cache (256K context, 16bit)
LLAMA-2	6.7B	6.7B	128GB
Mistral	7.2B	7.2B	32GB
Mixtral	46.7B	12.9B	32GB
Jamba	52B	12B	4GB

Context length fitting a single 80GB A100 GPU





Mamba

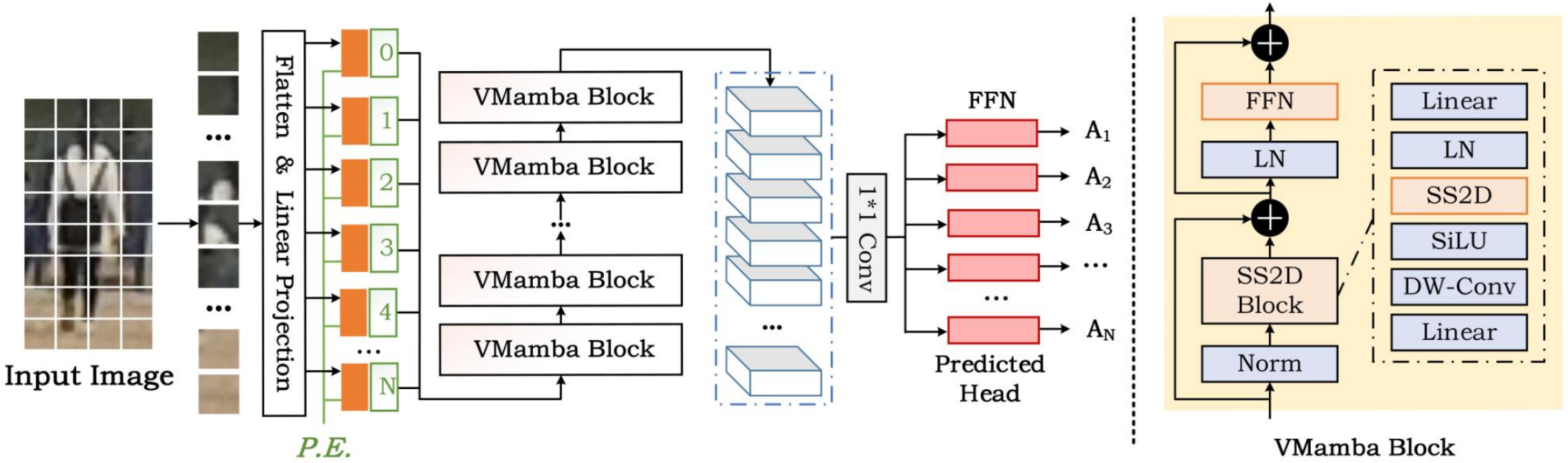


(a) Jamba block

(b) Different types of layers

Mamba for Pedestrian Attribute Recognition

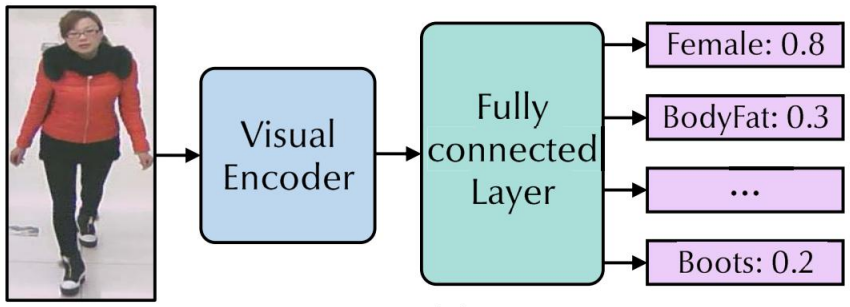
➢ Mamba based Multi-label Classification Framework

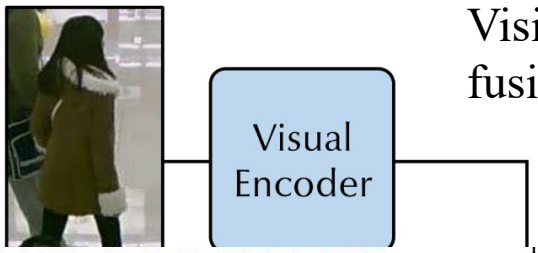


Methods	Backbone	PETA					PA100K				
		mA	Acc	Prec	Recall	F1	mA	Acc	Prec	Recall	F1
VTB (TCSVT 2022) [4]	ViT-B/16	85.31	79.60	86.76	87.17	86.71	83.72	80.89	87.88	89.30	88.21
VTB* (TCSVT 2022) [4]	ViT-L/14	86.34	79.59	86.66	87.82	86.97	85.30	81.76	87.87	90.67	88.86
Baseline (only VMamba)	VMamba-B	86.28	80.54	87.45	87.95	87.45	83.63	81.11	87.59	89.98	88.40

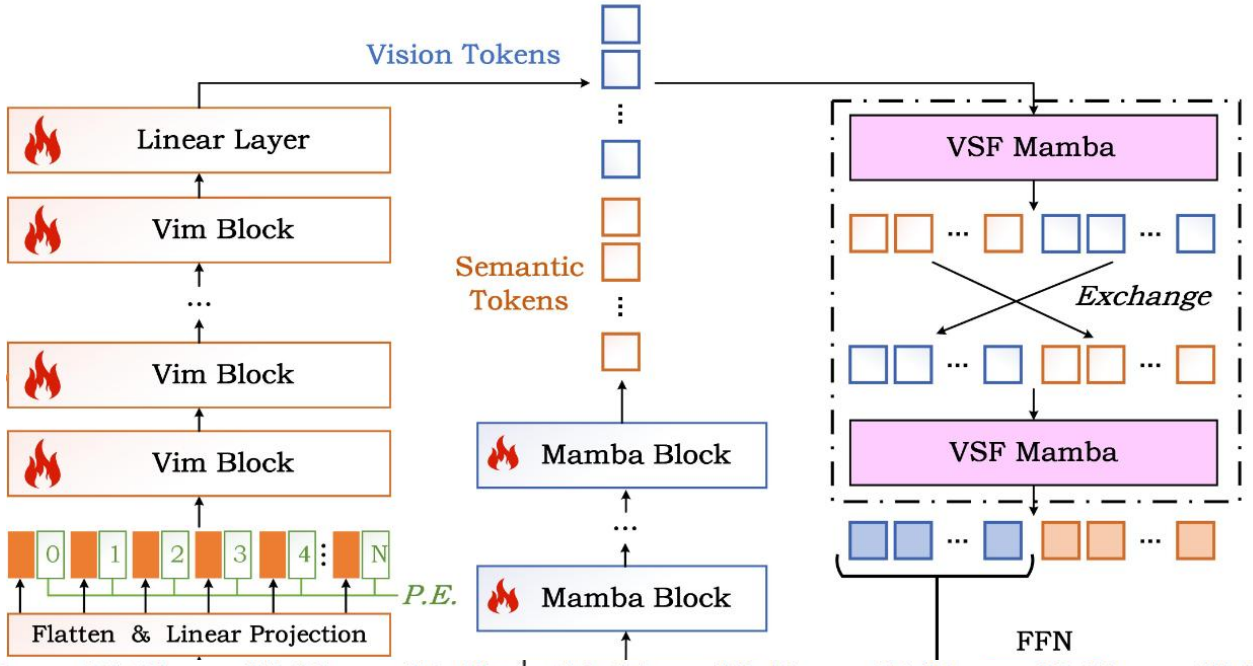
Mamba for Pedestrian Attribute Recognition

➢ Mamba based Multi-label Classification Framework





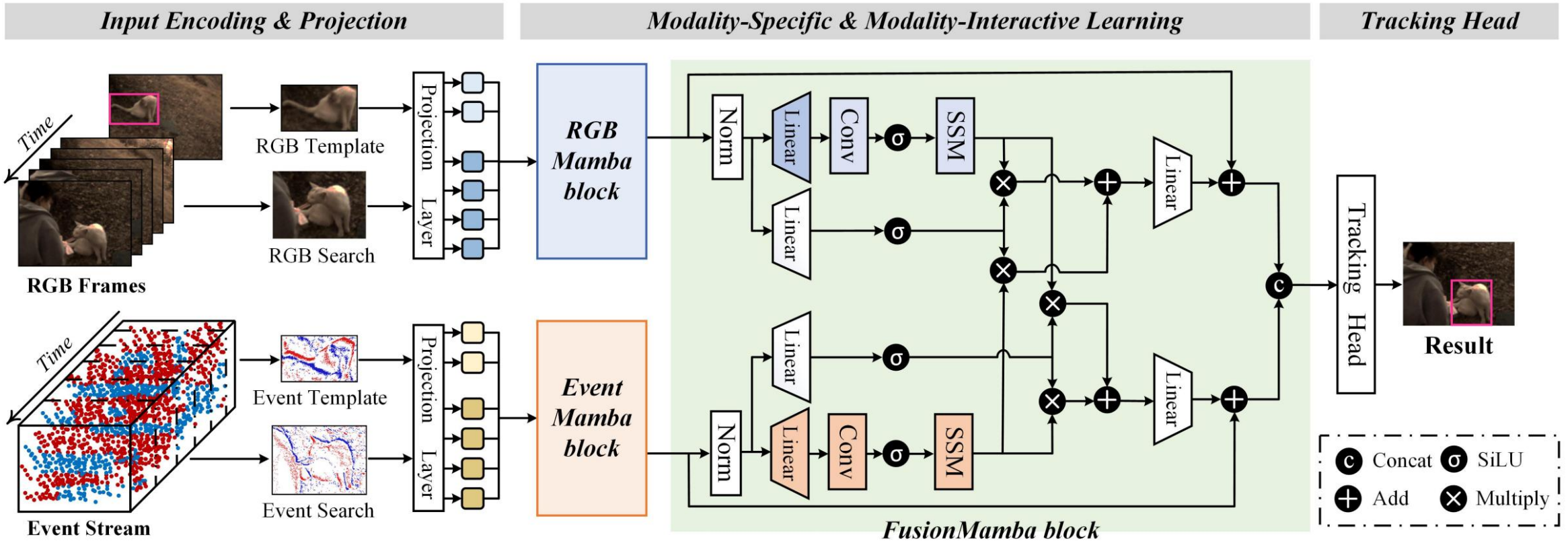
Vision + Language fusion for PAR



DAFL (AAAI 2022) [20]	ResNet50	87.07	78.88	85.78	87.03	86.40	83.54	80.13	87.01	89.19	88.09
CGCN (TMM 2022) [6]	ResNet	87.08	79.30	83.97	89.38	86.59	-	-	-	-	-
CAS-SAL-FR (IJCV 2022) [39]	ResNet50	86.40	79.93	87.03	87.33	87.18	82.86	79.64	86.81	87.79	85.18
VTB (TCSVT 2022) [4]	ViT-B/16	85.31	79.60	86.76	87.17	86.71	83.72	80.89	87.88	89.30	88.21
VTB* (TCSVT 2022) [4]	ViT-L/14	86.34	79.59	86.66	87.82	86.97	85.30	81.76	87.87	90.67	88.86
Baseline (only VMamba)	VMamba-B	86.28	80.54	87.45	87.95	87.45	83.63	81.11	87.59	89.98	88.40
MambaPAR (V+L)	Vim-S	84.25	76.07	82.73	87.20	84.56	80.87	79.33	86.48	88.79	87.21
MambaPAR (V+L)	VMamba-B	85.01	78.47	85.12	87.49	86.00	81.50	79.55	87.54	88.03	87.35

Frame-Event Single Object Tracking

Mamba-FETrack: Frame-Event Tracking via State Space Model, arXiv:2404.18174,
Ju Huang, Shiao Wang, Shuai Wang, Zhe Wu, Xiao Wang, Bo Jiang
https://github.com/Event-AHU/Mamba_FETrack



Frame-Event Single Object Tracking

Table 2: Tracking results on FELT SOT dataset (SR/PR). Note that, OSTrack* indicates that we first perform image fusion at the input level on the two modalities.

Trackers	Publish	RGB	Event	RGB+Event	FPS
01. STARK [8]	ICCV21	45.6/58.2	39.3/50.8	45.7/59.4	42
02. GRM [58]	CVPR23	44.7/55.9	39.2/48.9	44.5/55.9	45
03. PrDiMP [57]	CVPR20	43.8/54.7	34.9/44.5	43.8/55.2	30
04. DiMP [56]	ICCV19	43.3/54.5	37.8/48.5	43.5/55.1	43
05. SuperDiMP [56]	-	43.0/53.5	37.8/47.8	43.0/54.2	-
06. TransT [7]	CVPR21	42.2/53.0	35.2/45.1	34.6/44.3	50
07. TOMP50 [59]	CVPR22	41.2/51.8	37.7/49.2	43.4/55.2	25
08. ATOM [55]	CVPR19	37.5/47.0	22.3/28.4	36.2/45.9	30
09. KYS [53]	ECCV20	35.9/45.0	22.5/29.5	33.1/42.4	20
10. OSTrcak-B [48]	ECCV22	-	-	45.0/56.6	63
11. OSTrack-S [48]	ECCV22	-	-	40.0/50.9	84
12. OSTrack* [48]	ECCV22	45.2/56.3	37.4/46.9	32.5/40.3	105
13. AFNet [39]	CVPR23	-	-	28.9/36.6	36
14. Mamba-FETrack	-	43.3/55.1	38.3/48.3	43.5/55.6	24

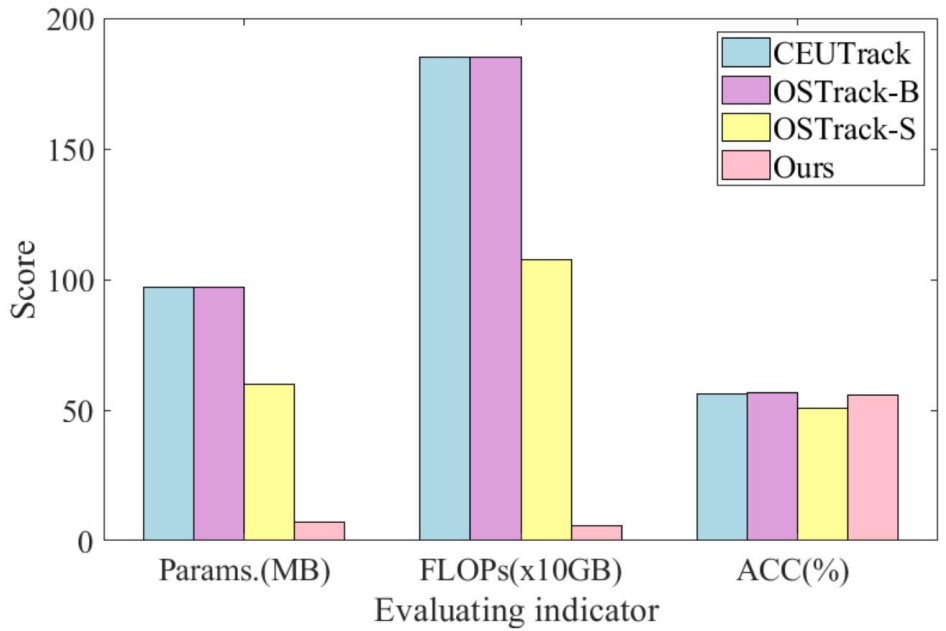


Table 1: Experimental results (SR/PR) on FE108 dataset.

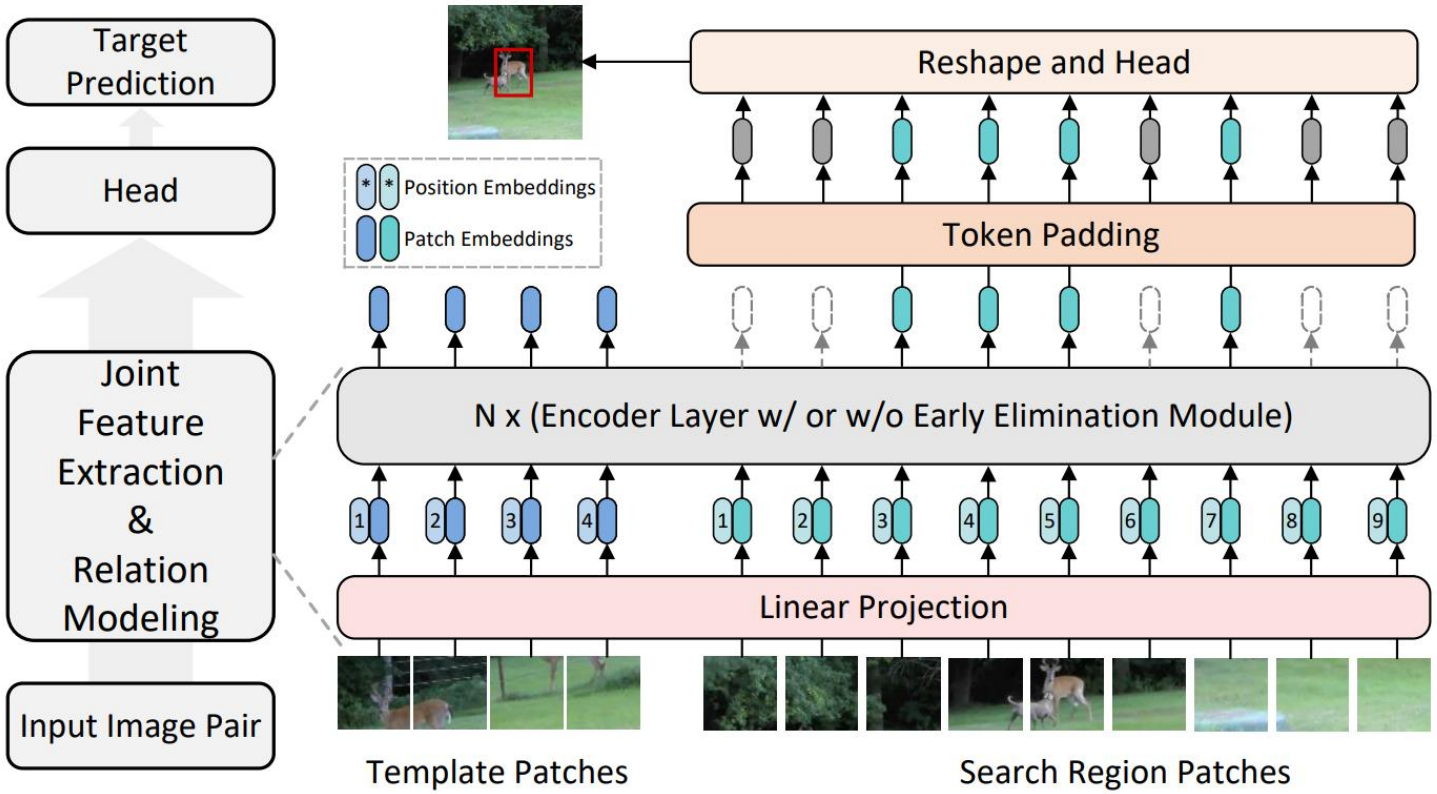
Tracker	SiamRPN [51]	SiamBAN [52]	SiamFC++ [4]	KYS [53]	CLNet [54]	CMT-MDNet [15]
AUC/PR	21.8/33.5	22.5/37.4	23.8/39.1	26.6/41.0	34.4/55.5	35.1/57.8
Tracker	**ATOM [55]**	**DiMP [56]**	**PrDiMP [57]**	**CMT-ATOM [15]**	**CEUTrack [16]**	**Ours -**
AUC/PR	46.5/71.3	52.6/79.1	53.0/80.5	54.3/79.4	55.58/84.46	58.71/90.95

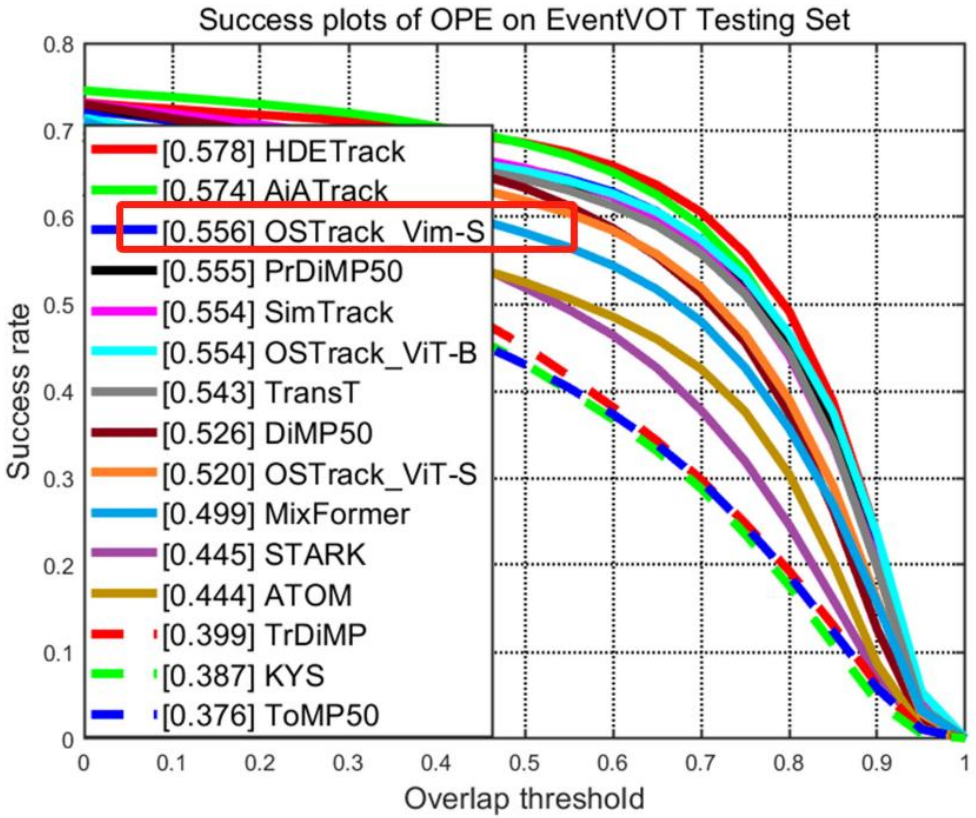
Table 7: Comparison on Tracking Speed, Parameters, and FLOPs.

Tracker	Ours -	CEUTrack [16]	AMTTrack [37]	OSTrack-B [48]	OSTrack-S [48]
FPS	24	75	61	63	84
FLOPs (GB)	59	1850	2070	1850	1076
Parameters (MB)	7	97	108	97	60

Mamba for Event-based Tracking

- **OSTrack + Vim for Event Stream based Tracking**



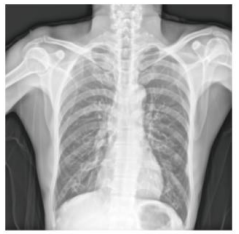


Mamba for Image-to-Text Generation

➢ Medical Report Generation

Comparison between the performance of R2Gen-GPT-Vim-Small and with other methods on IU-Xray dataset. R2Gen-GPT-Vim-S* and R2GenGPT-Vim-S denote the Vim-S are initialized with and without pre-trained parameters, respectively.

Methods	Backbone	CIDEr	BLEU-4	ROUGE-L
R2Gen [226]	CNN	0.398	0.165	0.371
KERP [227]	CNN	0.280	0.162	0.339
HRGP [228]	CNN	0.343	0.151	0.322
MKG [229]	CNN	0.304	0.147	0.367
PPKED [230]	CNN	0.351	0.168	0.376
MGSK [231]	CNN	0.382	0.178	0.381
CA [232]	ResNet-50	-	0.169	0.381
CMCL [233]	CNN	-	0.162	0.378
DCL [234]	CNN	0.586	0.163	0.383
R2GenGPT	Swin-B	0.524	0.152	0.352
R2GenGPT	Vim-S	0.388	0.152	0.355
R2GenGPT	Vim-S*	0.382	0.171	0.371

Image	Ground Truth
	胸廓对称，气管居中；两侧肋骨、肋间隙正常；两下肺纹理增多增粗，其间见斑点、小斑片状密度增高影，两侧肺门和纵隔影未见明显异常；主动脉结突出伴有钙化影，心影横径稍增大；膈肌平滑，双侧肋膈角锐利。
	两侧胸廓对称，两肺未见明显实质性病变，两侧膈面光滑，两侧肋膈角锐利。心影形态、大小未见明显异常。
	两肺纹理稍增多，可见散在分布斑点状高密度影，边界尚清，心影大小形态大致正常范围内，双侧膈肌光滑，肋膈角锐利。

Mamba for Re-ID

➢ Person / Vehicle Re-Identification

Comparison with methods based on CNN and Transformer on Person Re-identification and Vehicle Re-identification datasets.

Backbone	Method	MSMT17		Market1501		DukeMTMC		Occluded-Duke		Method	VeRi-776		VehicleID	
		mAP	R1	mAP	R1	mAP	R1	mAP	R1		mAP	R1	R1	R5
CNN	CBN [235]	42.9	72.8	77.3	91.3	67.3	82.5	-	-	PRReID [236]	72.5	93.3	72.6	88.6
	OSNet [237]	52.9	78.7	84.9	94.8	73.5	88.6	-	-	SAN [238]	72.5	93.3	79.7	94.3
	MGN [239]	52.1	76.9	86.9	95.7	78.4	88.7	-	-	UMTS [240]	75.9	95.8	80.9	87.0
	RGA-SC [241]	57.5	80.3	88.4	96.1	-	-	-	-	VANet [242]	66.3	89.8	83.3	96.0
	SAN [243]	55.7	79.2	88.0	96.1	75.7	87.9	-	-	SPAN [244]	68.9	94.0	-	-
	SCSN [245]	58.5	83.8	88.5	95.7	79.0	91.0	-	-	PGAN [246]	79.3	96.5	78.0	93.2
	ABDNet [247]	60.8	82.3	88.3	95.6	78.6	89.0	-	-	PVEN [248]	79.5	95.6	84.7	97.0
	PGFA [249]	-	-	76.8	91.2	65.5	82.6	37.3	51.4	SAVER [250]	79.6	96.4	79.9	95.2
	HOReID [251]	-	-	84.9	94.2	75.6	86.9	43.8	55.1	CFVMNet [252]	77.1	95.3	81.4	94.1
	ISP [253]	-	-	88.6	95.3	80.0	89.6	52.3	62.8	GLAMOR [254]	80.3	96.5	78.6	93.6
Transformer	DeiT-B/16 [255]	61.4	81.9	86.6	94.4	78.9	89.3	53.1	60.6	DeiT-B/16 [255]	78.4	95.9	83.1	96.8
	ViT-B/16 [255]	61.0	81.8	86.8	94.7	79.3	88.8	53.1	60.5	ViT-B/16 [255]	78.2	96.5	82.3	96.1
	VehicleMAE [256]	-	-	-	-	-	-	-	-	VehicleMAE [256]	85.6	97.9	-	-
Mamba	Vim-T/16	40.1	62.6	75.7	89.4	66.5	81.8	35.4	45.1	Vim-T/16	62.9	89.2	67.0	88.2
	Vim-S/16	42.2	66.2	77.5	89.7	67.4	83.0	40.8	51.3	Vim-S/16	61.6	89.6	78.2	94.8
	VMamba-T/16	51.0	75.6	83.3	92.8	74.9	87.3	49.4	58.3	VMamba-T/16	77.3	95.9	78.5	93.5
	VMamba-B/16	51.1	75.3	84.3	93.2	77.4	88.0	48.1	57.4	VMamba-B/16	77.5	95.6	82.5	96.1

Future Works

- Current SSMs model still performs inferior to the mainstream of Transformer networks.
- The advantages of the SSMs in GPU usage are worth further exploration and research.
- To further explore its advantages in high-resolution or long-term vision data is a direction worthy of attention and research.
- Pre-trained big models using SSMs architecture.
- Multi-modal learning using SSMs architecture.
- Developing novel scan operators for the SSMs.
- The generalization performance of SSMs still deserves attention and further research and improvement.
- Use the latest SSM model to empower the existing deep neural network model.

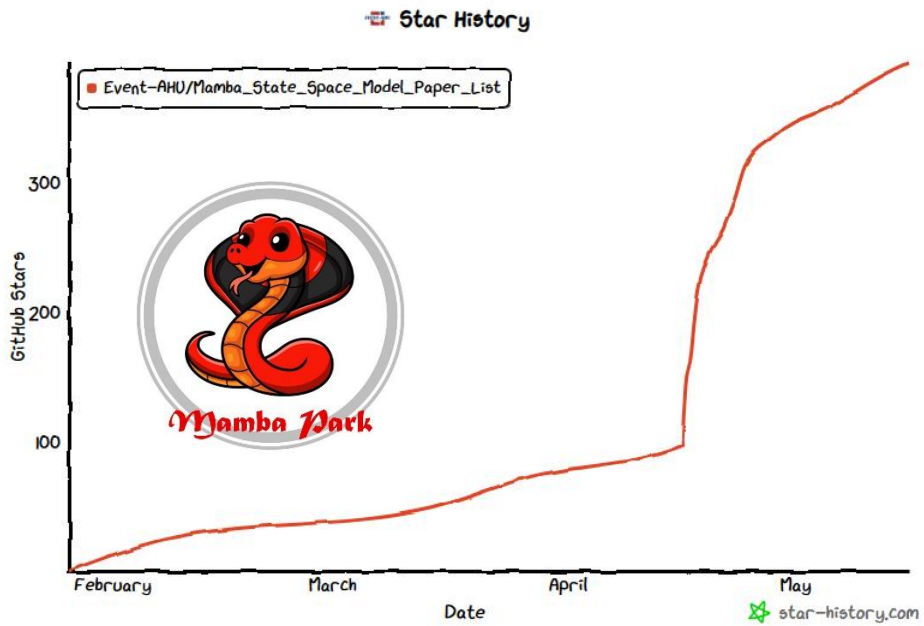
Computer Science > Machine Learning

[Submitted on 15 Apr 2024]

State Space Model for New-Generation Network Alternative to Transformers: A Survey

Xiao Wang, Shiao Wang, Yuhe Ding, Yuehang Li, Wentao Wu, Yao Rong, Weizhe Kong, Ju Huang, Shihao Li, Haoxiang Yang, Ziwen Wang, Bo Jiang, Chenglong Li, Yaowei Wang, Yonghong Tian, Jin Tang

In the post-deep learning era, the Transformer architecture has demonstrated its po
downstream tasks. However, the enormous computational demands of this architec
complexity of attention models, numerous efforts have been made to design more
possible replacement for the self-attention based Transformer model, has drawn m
comprehensive review of these works and also provide experimental comparisons
SSM. Specifically, we first give a detailed description of principles to help the reade
reviews of existing SSMs and their various applications, including natural language
point cloud/event stream, time series data, and other domains. In addition, we give
helps the readers to understand the effectiveness of different structures on various
better promote the development of the theoretical model and application of SSM. M
GitHub: this https URL.



Comments: The First review of State Space Model (SSM)/Mamba and their applications in artific

Access Paper:

- View PDF
- HTML (experimental)
- TeX Source
- Other Formats

view license

Current browse context:
cs.LG

< prev | next >
new | recent | 2404

Change to browse by:
cs
cs.AI
cs.CL
cs.CV
cs.MM

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

Export BibTeX Citation

Bookmark

More Surveys

① **Modeling sequences with structured state spaces**, Responsibility: Albert Gu, Publication: [Stanford, California] : [Stanford University], 2023 [Thesis (330 pages)] [PDF]

② **State Space Model for New-Generation Network Alternative to Transformers: A Survey**, Xiao Wang, Shiao Wang, Yuhe Ding, Yuehang Li, Wentao Wu, Yao Rong, Weizhe Kong, Ju Huang, Shihao Li, Haoxiang Yang, Ziwen Wang, Bo Jiang, Chenglong Li, Yaowei Wang, Yonghong Tian, Jin Tang, 2024 [PDF] [arXiv]

③ **State Space Models as Foundation Models: A Control Theoretic Overview**, arXiv:2403.16899, Carmen Amo Alonso, Jerome Sieber, Melanie N. Zeilinger [Paper]

④ **A Survey on Visual Mamba**, arXiv:2404.15956, Hanwei Zhang, Ying Zhu, Dan Wang, Lijun Zhang, Tianxiang Chen, Zi Ye [Paper]

⑤ **Mamba-360: Survey of State Space Models as Transformer Alternative for Long Sequence Modelling: Methods, Applications, and Challenges,** Badri Narayana Patro, Vijay Srinivas Agneeswaran [Paper] [Github]

⑥ **A Survey on Vision Mamba: Models, Applications and Challenges**, Rui Xu, Shu Yang, Yihui Wang, Bo Du, Hao Chen [Paper] [Paper List]

⑦ **Vision Mamba: A Comprehensive Survey and Taxonomy**, arXiv:2405.03978, Xiao Liu, Chenxu Zhang, Lei Zhang [Paper] [Github]

Thanks for your attention!

Q & A

安徽大学—结构模式与视觉学习研究组
Anhui University-Structural Patterns and Visual Learning (AHU-SPVL) Group

个人主页
https://wangxiao5791509.github.io/