Demystify Mamba in Vision: A Linear Attention Perspective

Dongchen Han Ziyi Wang Zhuofan Xia Yizeng Han Yifan Pu Chunjiang Ge
Jun Song Shiji Song Bo Zheng Gao Huang

清華大學 Tsinghua University

NEURAL INFORMATION PROCESSING SYSTEMS

Background

paper code

Transformers has a <u>Quadratic Complexity</u> $\mathcal{O}(N^2 d)$ *with respect to sequence length.*

High Resolution Images

Videos

Mamba: a Powerful Selective State Space Model

paper code

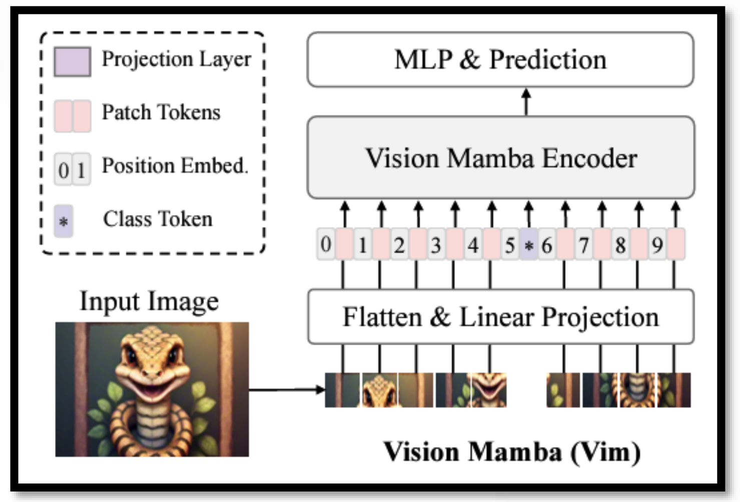
Mamba

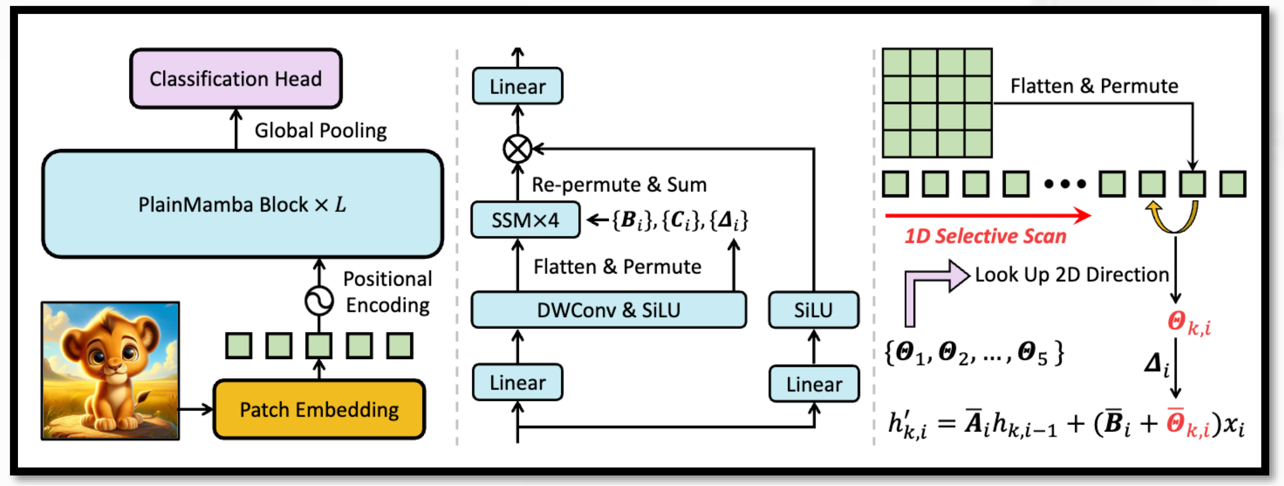
- ✓ ***High expressive capability***
- ✓ ***Linear complexity $\mathcal{O}(Nd^2)$***
- ✓ ***Global modeling***

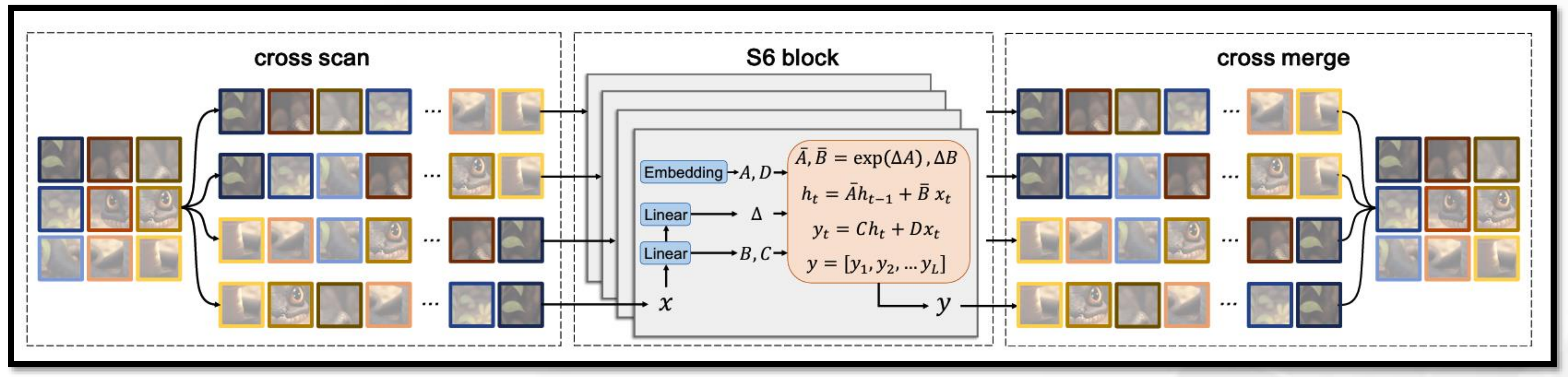
A ***promising*** method to deal with ***high-resolution*** images!

Fork 798 Star 10.2k

Mamba: **Linear-time sequence modeling** with selective state spaces
A Gu, T Dao - arXiv preprint arXiv:2312.00752, 2023 - arxiv.org
… to million-length **sequences**. As a general **sequence model** backbone, **Mamba** achieves state-… On language **modeling**, our **Mamba**-3B **model** outperforms Transformers of the same
Save Cite Cited by 339 Related articles All 4 versions







1. Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces[J]. arXiv preprint arXiv:2312.00752, 2023.

Motivation

paper code

Mamba

- ✓ Linear complexity $\mathcal{O}(N)$
- ✓ Global modeling
- ✓ **High expressive capability**

$$
\begin{aligned}
\boldsymbol{h}_i &= \widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_{i-1} + \boldsymbol{B}_i(\boldsymbol{\Delta}_i \odot \boldsymbol{x}_i), \\
\boldsymbol{y}_i &= \boldsymbol{C}_i \boldsymbol{h}_i \ / \ 1 + \boldsymbol{D} \odot \boldsymbol{x}_i.
\end{aligned}
$$

Linear Attention

- ✓ Linear complexity $\mathcal{O}(N)$
- ✓ Global modeling
- × **Inferior performance**

$$
\boldsymbol{y}_i = \sum_{j=1}^{N} \frac{\boldsymbol{Q}_i \boldsymbol{K}_j^\top}{\sum_{j=1}^{N} \boldsymbol{Q}_i \boldsymbol{K}_j^\top} \boldsymbol{V}_j = \frac{\boldsymbol{Q}_i \left(\sum_{j=1}^{N} \boldsymbol{K}_j^\top \boldsymbol{V}_j \right)}{\boldsymbol{Q}_i \left(\sum_{j=1}^{N} \boldsymbol{K}_j^\top \right)}
$$

1. *Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces[J]. arXiv preprint arXiv:2312.00752, 2023.*
2. *Katharopoulos A, Vyas A, Pappas N, et al. Transformers are rnns: Fast autoregressive transformers with linear attention[C]//International Conference on Machine Learning. PMLR, 2020: 5156-5165.*

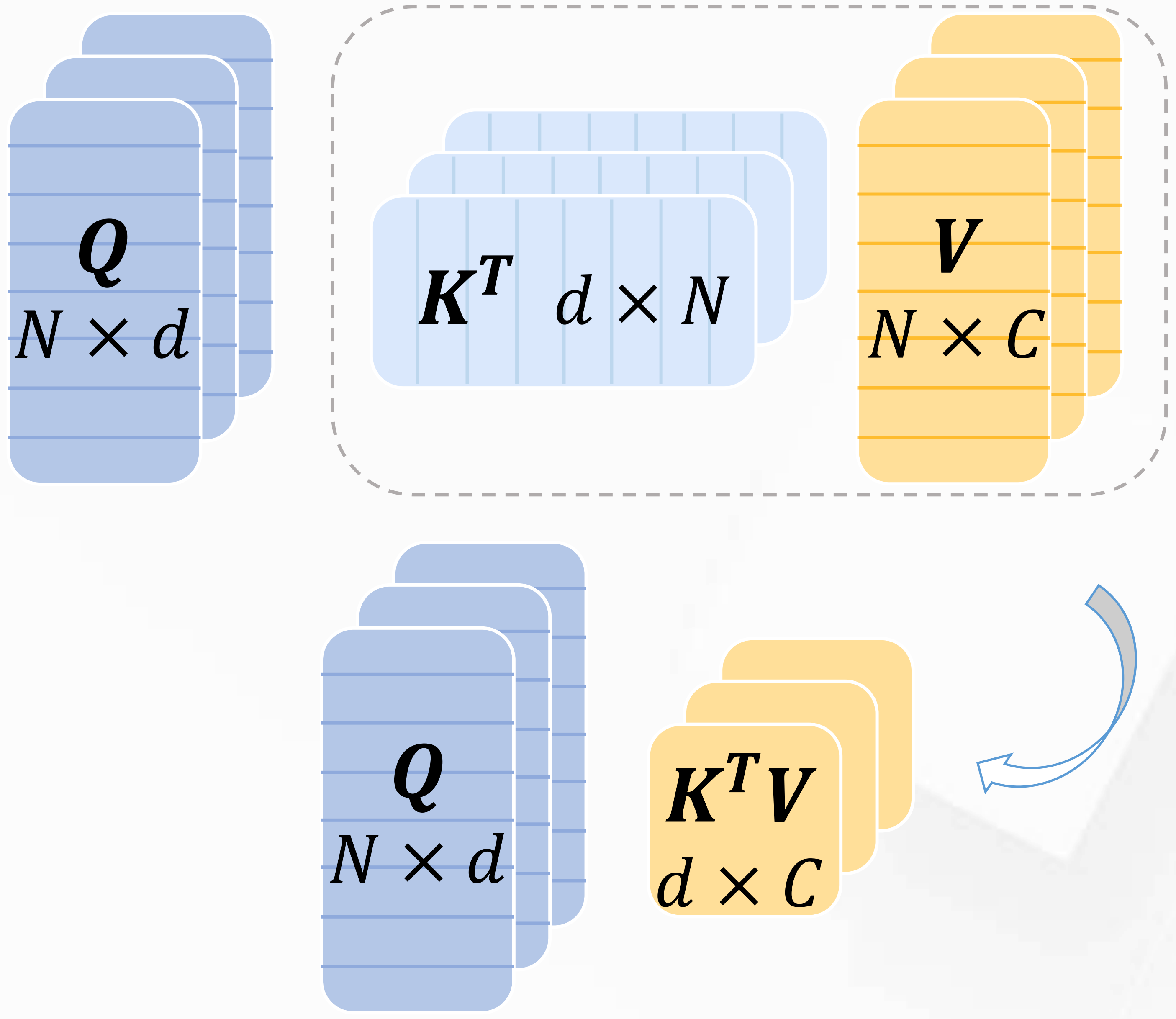
Linear Attention

Linear Attention $\quad O = QK^TV$

Carefully designed kernels are introduced as the approximation of the original similarity function:

$$Q = \phi(xW_Q), K = \phi(xW_K), V = xW_V$$

$$y_i = \sum_{j=1}^{N} \frac{Q_i K_j^\top}{\sum_{j=1}^{N} Q_i K_j^\top} V_j = \frac{Q_i \left(\sum_{j=1}^{N} K_j^\top V_j\right)}{Q_i \left(\sum_{j=1}^{N} K_j^\top\right)}$$



- × **Inferior performance**
- ✓ Linear complexity $\mathcal{O}(Nd^2)$

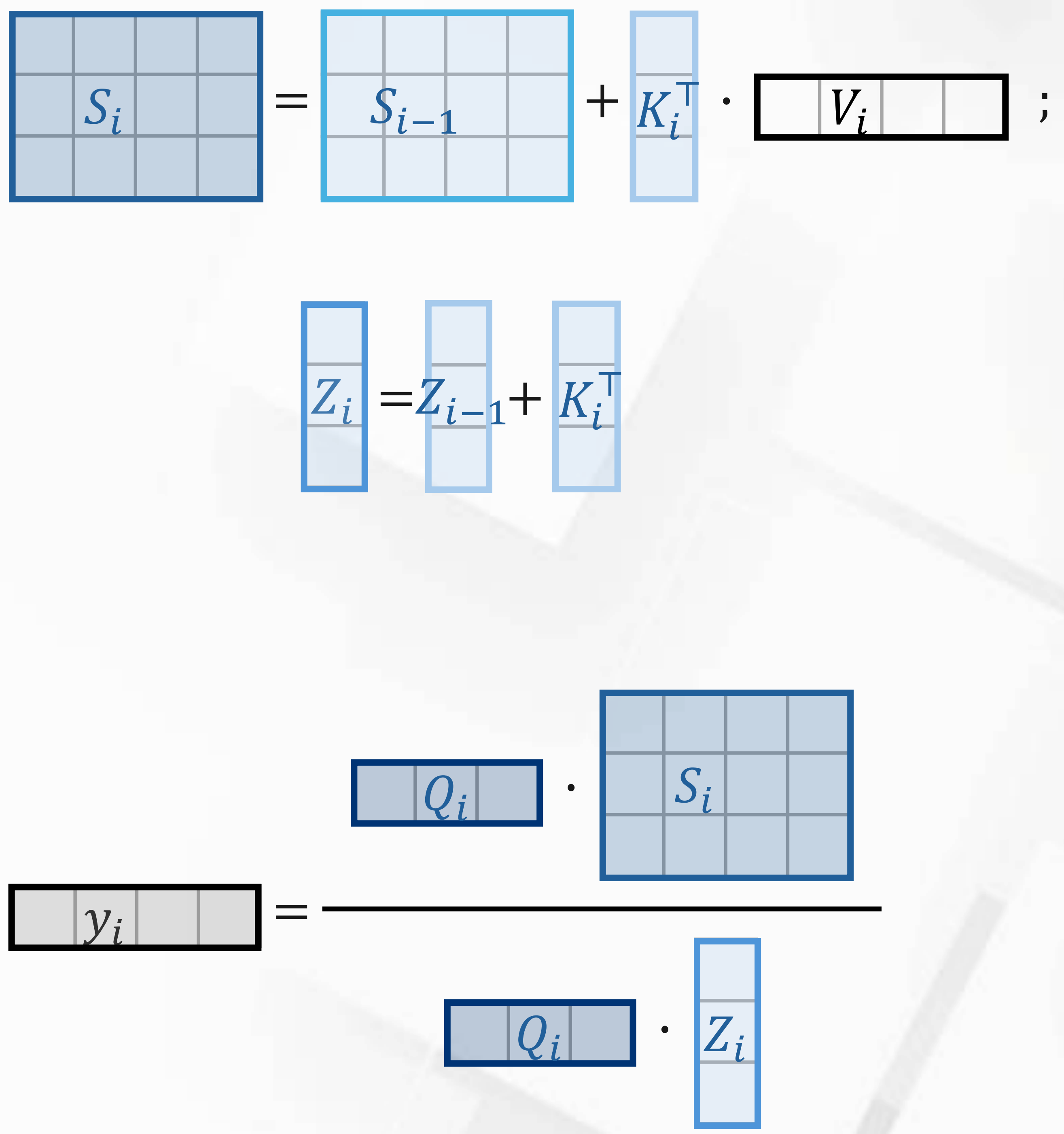
1. *Katharopoulos A, Vyas A, Pappas N, et al. Transformers are rnns: Fast autoregressive transformers with linear attention[C]//International Conference on Machine Learning. PMLR, 2020: 5156-5165.*

paper code

Recurrent Linear Attention

Non-causal linear attention (common linear attention):

$$\boldsymbol{y}_i = \sum_{j=1}^{N} \frac{\boldsymbol{Q}_i \boldsymbol{K}_j^\top}{\sum_{j=1}^{N} \boldsymbol{Q}_i \boldsymbol{K}_j^\top} \boldsymbol{V}_j = \frac{\boldsymbol{Q}_i \left(\sum_{j=1}^{\boxed{N}} \boldsymbol{K}_j^\top \boldsymbol{V}_j \right)}{\boldsymbol{Q}_i \left(\sum_{j=1}^{\boxed{N}} \boldsymbol{K}_j^\top \right)}$$



Causal linear attention:

$$\boldsymbol{y}_i = \frac{\boldsymbol{Q}_i \left(\sum_{j=1}^{\boxed{i}} \boldsymbol{K}_j^\top \boldsymbol{V}_j \right)}{\boldsymbol{Q}_i \left(\sum_{j=1}^{\boxed{i}} \boldsymbol{K}_j^\top \right)} \triangleq \frac{\boldsymbol{Q}_i \boldsymbol{S}_i}{\boldsymbol{Q}_i \boldsymbol{Z}_i}, \quad \boldsymbol{S}_i = \sum_{j=1}^{i} \boldsymbol{K}_j^\top \boldsymbol{V}_j, \ \ \boldsymbol{Z}_i = \sum_{j=1}^{i} \boldsymbol{K}_j^\top$$

Recurrent linear attention form:

$$\boldsymbol{S}_i = \boldsymbol{S}_{i-1} + \boldsymbol{K}_i^\top \boldsymbol{V}_i, \ \ \boldsymbol{Z}_i = \boldsymbol{Z}_{i-1} + \boldsymbol{K}_i^\top, \quad \boldsymbol{y}_i = \boldsymbol{Q}_i S_i / \boldsymbol{Q}_i \boldsymbol{Z}_i.$$

Selective State Space Model (Scalar Input)

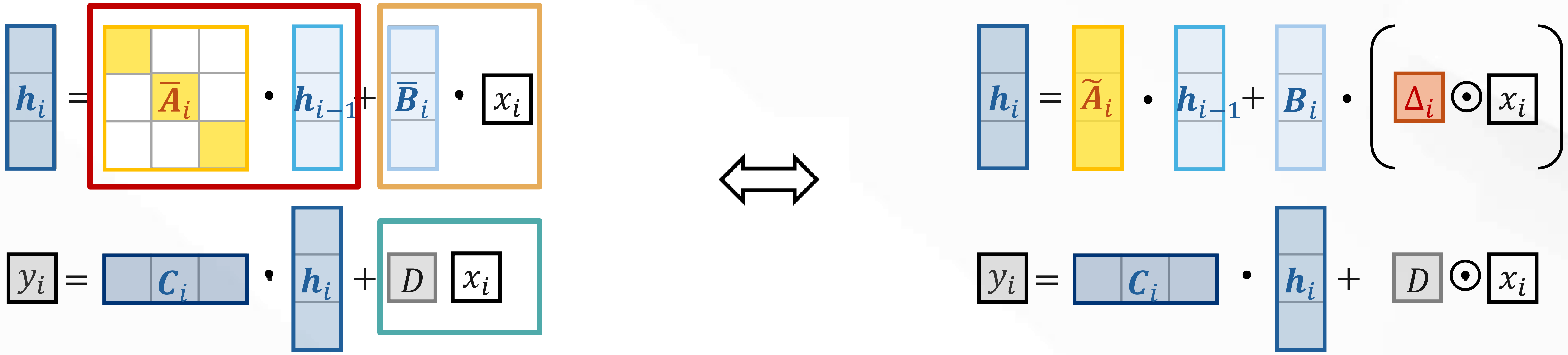
paper code

$$
\begin{aligned}
\boldsymbol{h}_i &= \overline{\boldsymbol{A}}_i \boldsymbol{h}_{i-1} + \overline{\boldsymbol{B}}_i x_i, \\
y_i &= \boldsymbol{C}_i \boldsymbol{h}_i + D x_i,
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{h}_i &= \widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_{i-1} + \boldsymbol{B}_i (\Delta_i \odot x_i), \\
y_i &= \boldsymbol{C}_i h_i + D \odot x_i,
\end{aligned}
$$

$$
x_i \in \mathbb{R}, \quad \overline{\boldsymbol{A}}_i \in \mathbb{R}^{d \times d}, \quad \overline{\boldsymbol{B}}_i, \boldsymbol{h}_{i-1}, \boldsymbol{h}_i \in \mathbb{R}^{d \times 1}, \\
y_i \in \mathbb{R}, \quad \boldsymbol{C}_i \in \mathbb{R}^{1 \times d}, \quad D \in \mathbb{R}.
$$

$$
x_i, \Delta_i \in \mathbb{R}, \quad \widetilde{\boldsymbol{A}}_i, \boldsymbol{B}_i, \boldsymbol{h}_{i-1}, \boldsymbol{h}_i \in \mathbb{R}^{d \times 1}, \\
y_i \in \mathbb{R}, \quad \boldsymbol{C}_i \in \mathbb{R}^{1 \times d}, \quad D \in \mathbb{R}.
$$



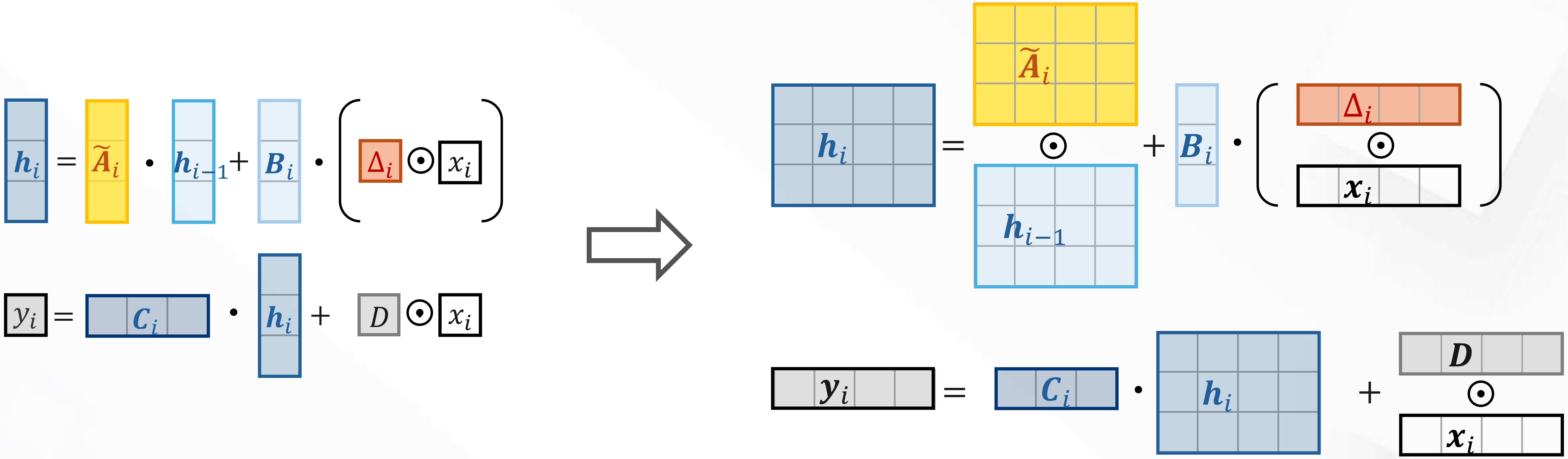
(1) $\overline{\boldsymbol{A}}_i \boldsymbol{h}_{i-1} = \widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_{i-1}$ (2) $\overline{\boldsymbol{B}}_i x_i = \Delta_i \boldsymbol{B}_i x_i = \boldsymbol{B}_i (\Delta_i x_i) = \boldsymbol{B}_i (\Delta_i \odot x_i)$ (3) $D x_i = D \odot x_i$

paper code

Selective State Space Model (Vector Input)

$$\boldsymbol{h}_i = \widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_{i-1} + \boldsymbol{B}_i(\boldsymbol{\Delta}_i \odot \boldsymbol{x}_i), \qquad \boldsymbol{x}_i, \boldsymbol{\Delta}_i \in \mathbb{R}^{1\times C}, \;\; \widetilde{\boldsymbol{A}}_i, \boldsymbol{h}_{i-1}, \boldsymbol{h}_i \in \mathbb{R}^{d\times C}, \;\; \boldsymbol{B}_i \in \mathbb{R}^{d\times 1}$$

$$\boldsymbol{y}_i = \boldsymbol{C}_i \boldsymbol{h}_i + \boldsymbol{D} \odot \boldsymbol{x}_i, \qquad \boldsymbol{y}_i \in \mathbb{R}^{1\times C}, \;\; \boldsymbol{C}_i \in \mathbb{R}^{1\times d}, \;\; \boldsymbol{D} \in \mathbb{R}^{1\times C},$$

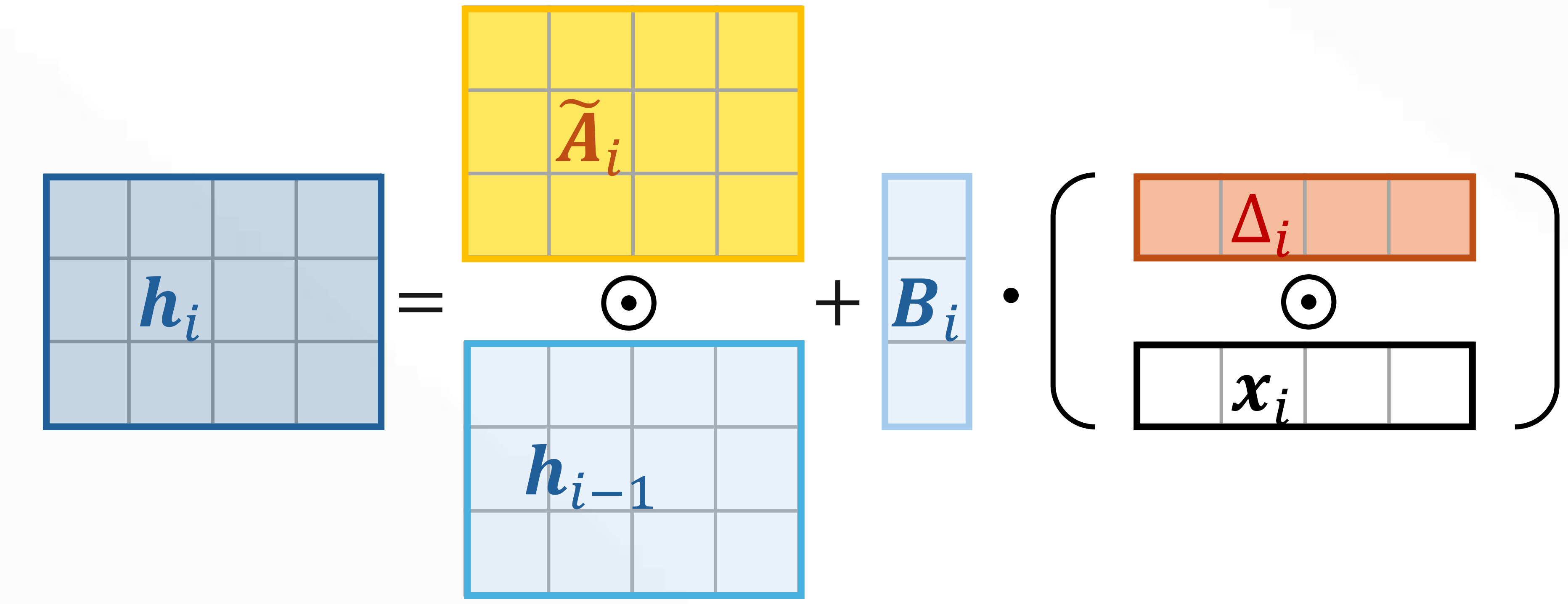


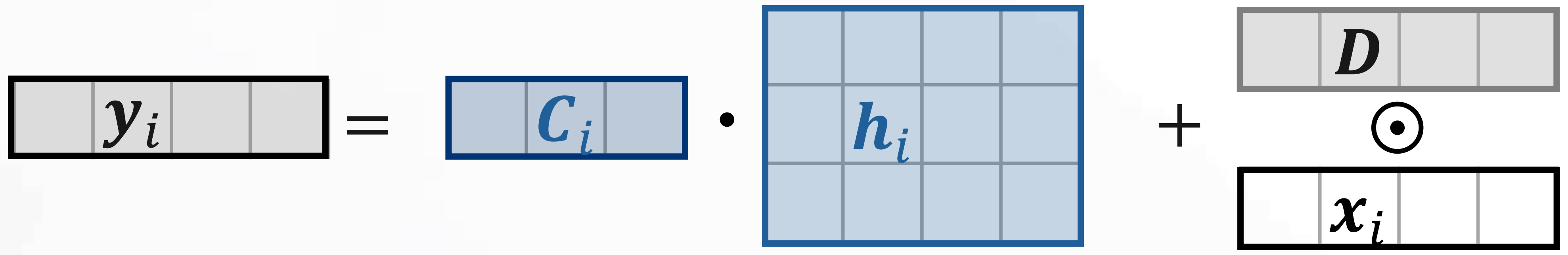
Mamba v.s. Linear Attention Transformer

paper code

Selective SSM in Mamba

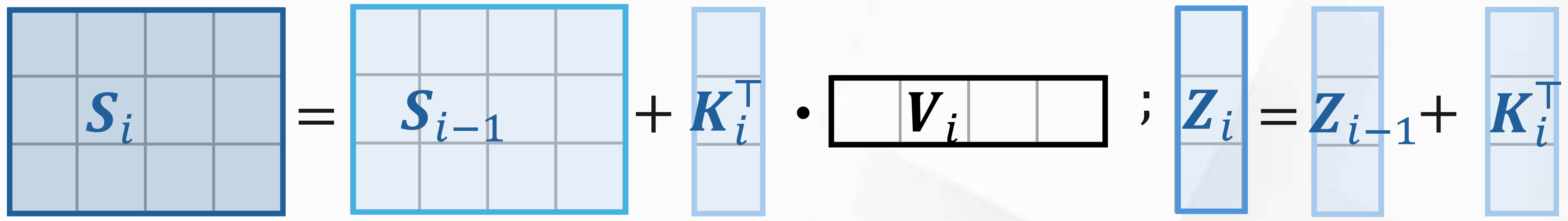
$$\begin{aligned} \boldsymbol{h}_i &= \widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_{i-1} + \boldsymbol{B}_i(\boldsymbol{\Delta}_i \odot \boldsymbol{x}_i), \\ \boldsymbol{y}_i &= \boldsymbol{C}_i \boldsymbol{h}_i \;/\; 1 + \boldsymbol{D} \odot \boldsymbol{x}_i. \end{aligned}$$

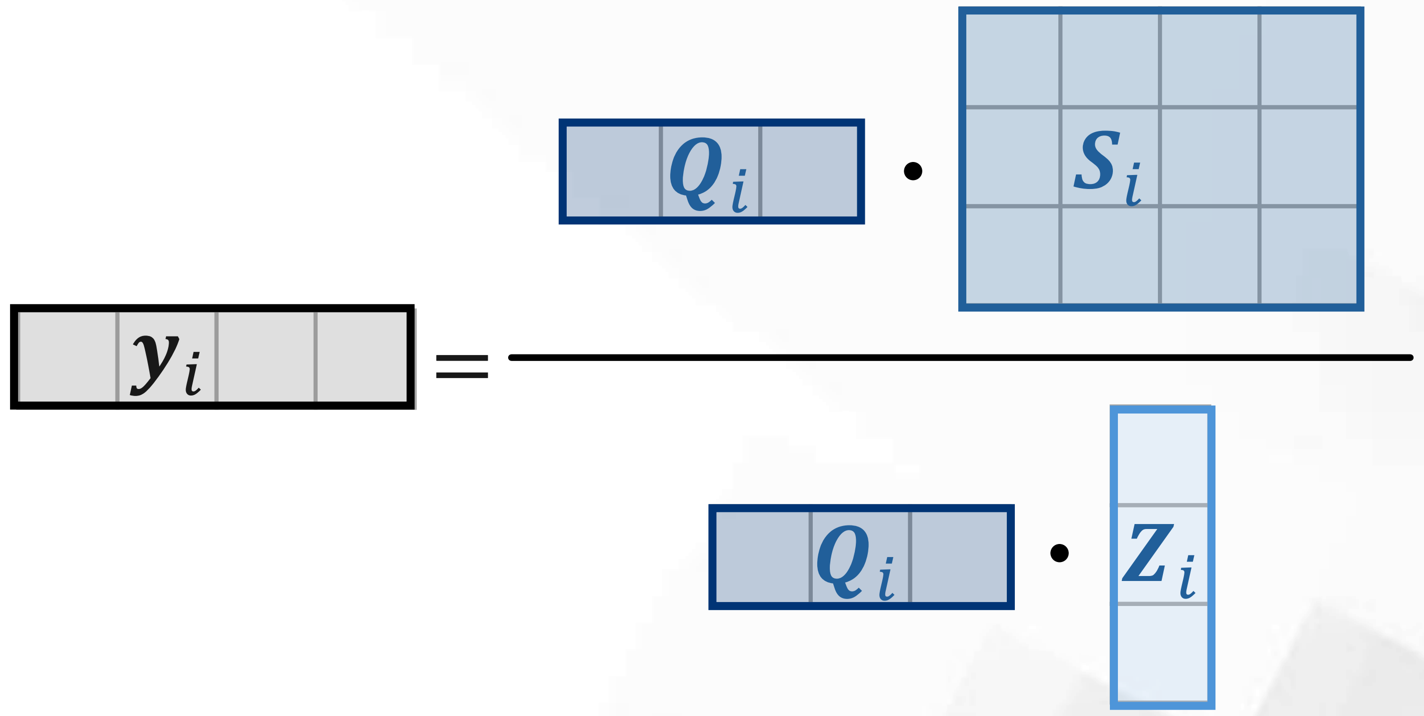




Single-head Linear Attention

$$\begin{aligned} \boldsymbol{S}_i &= \boldsymbol{1} \odot \boldsymbol{S}_{i-1} + \boldsymbol{K}_i^\top(\boldsymbol{1} \odot \boldsymbol{V}_i), \\ \boldsymbol{y}_i &= \boldsymbol{Q}_i S_i \;/\; \boldsymbol{Q}_i \boldsymbol{Z}_i + \boldsymbol{0} \odot \boldsymbol{x}_i. \end{aligned}$$

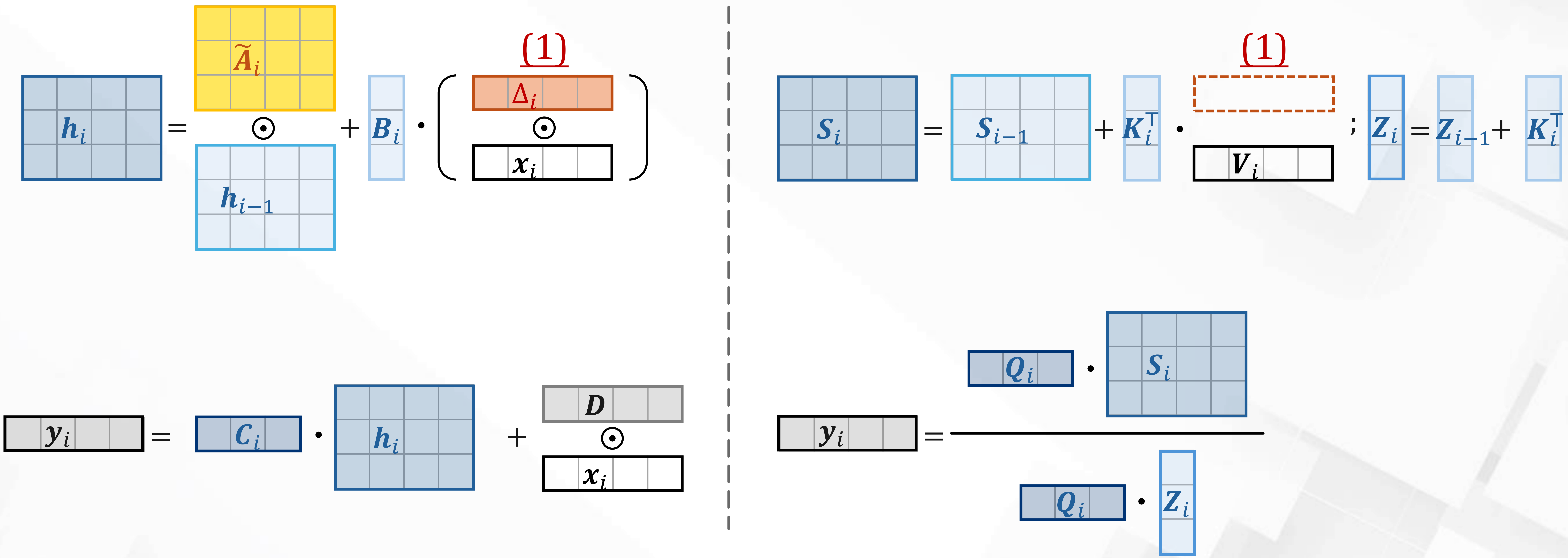




Mamba v.s. Linear Attention Transformer

paper code

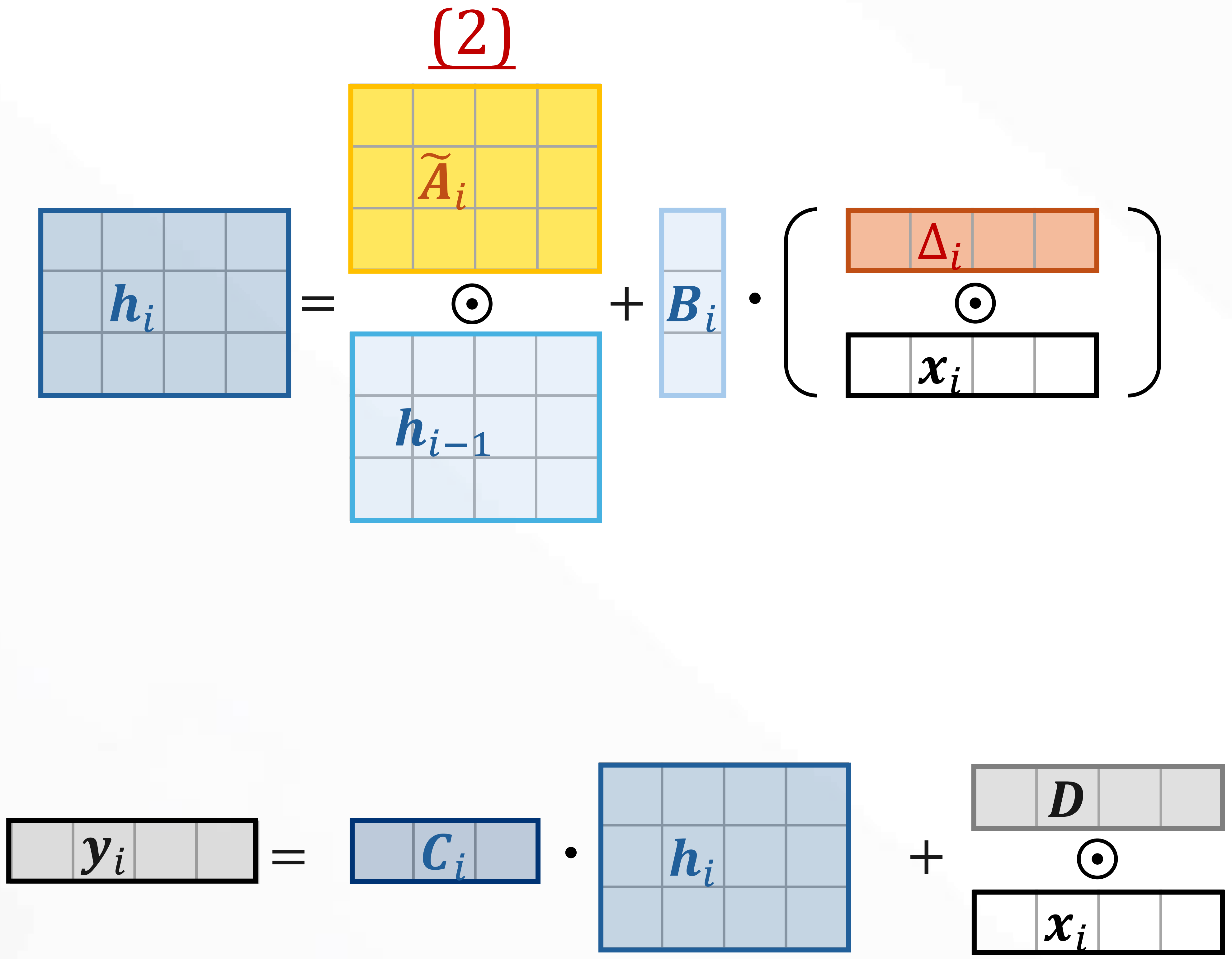
Four differences: (1) Δ_i : input gate

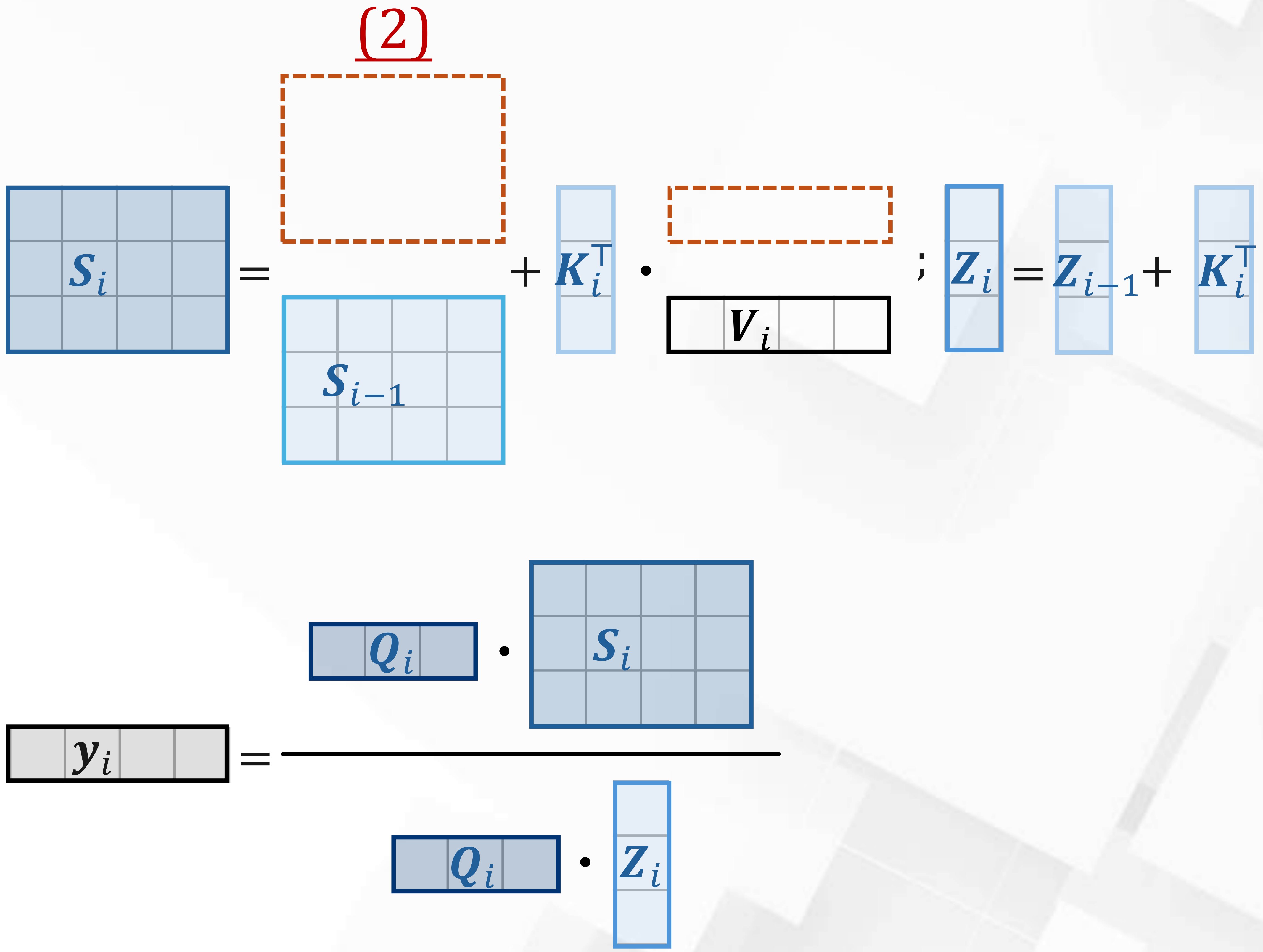


Mamba v.s. Linear Attention Transformer

paper code

Four differences: (1) Δ_i : input gate (2) $\tilde{A}_i$: forget gate





Mamba v.s. Linear Attention Transformer

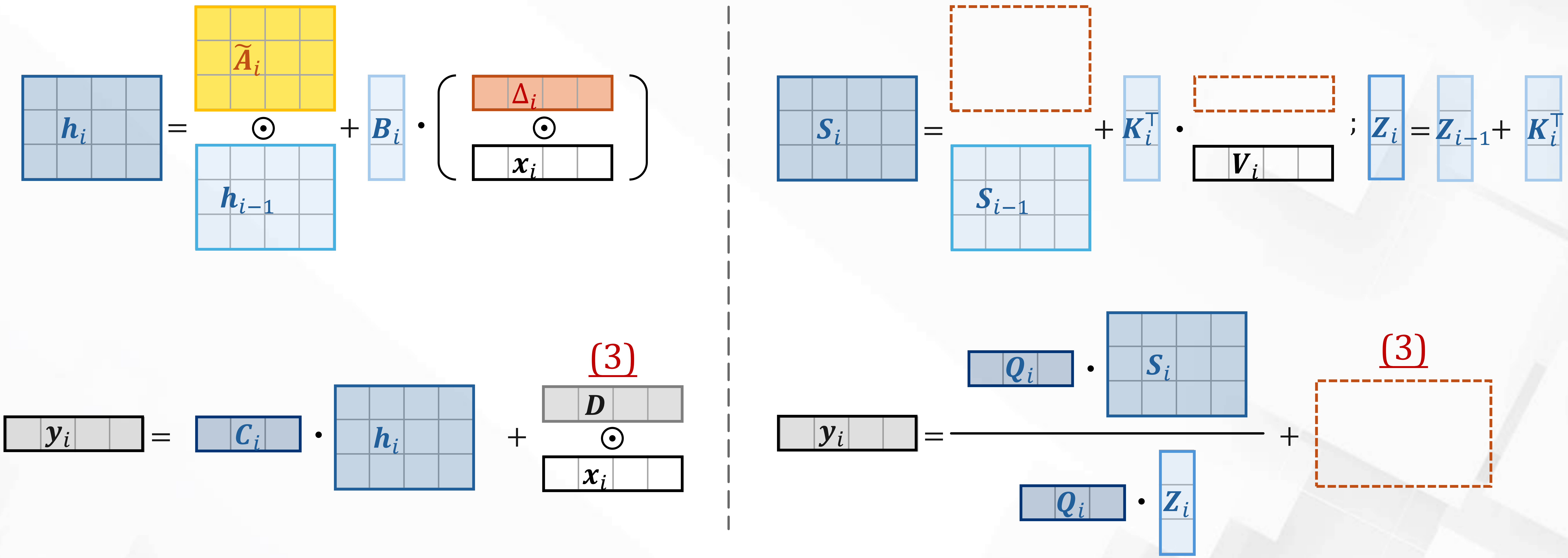
paper code

Four differences:

(1) Δ_i : input gate

(2) $\tilde{A}_i$: forget gate

(3) $D \odot x_i$: shortcut



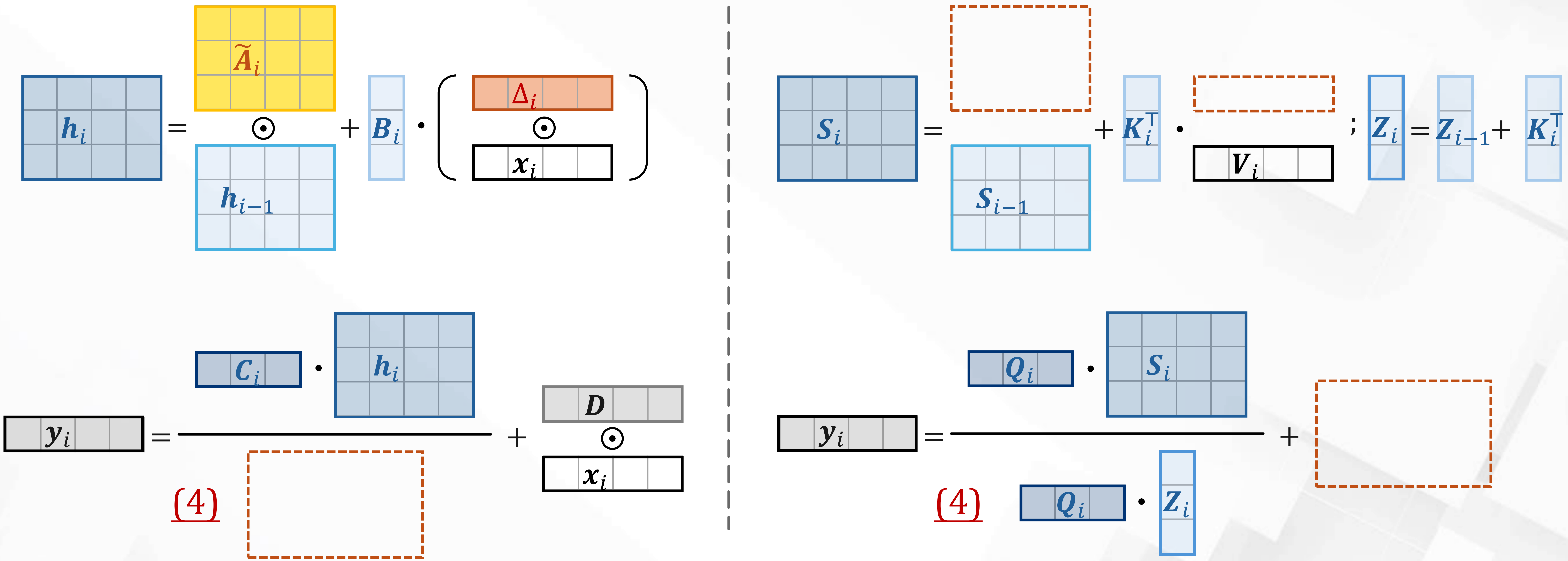
Mamba v.s. Linear Attention Transformer

paper code

Four differences:

(1) Δ_i : input gate (2) $\tilde{A}_i$: forget gate

(3) $D \odot x_i$: shortcut (4) $Q_i Z_i$: attention normalization



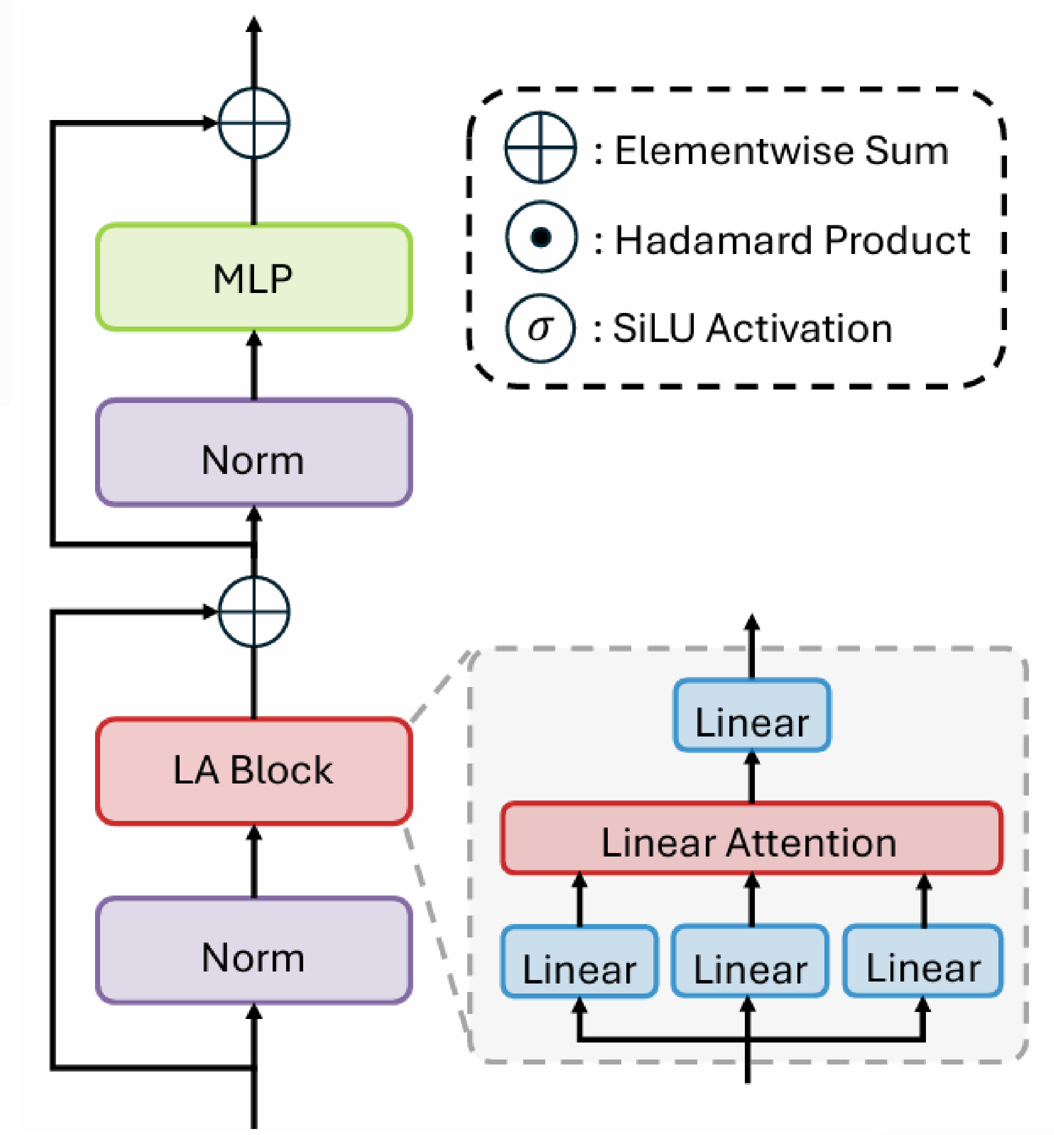
Mamba v.s. Linear Attention Transformer

paper code

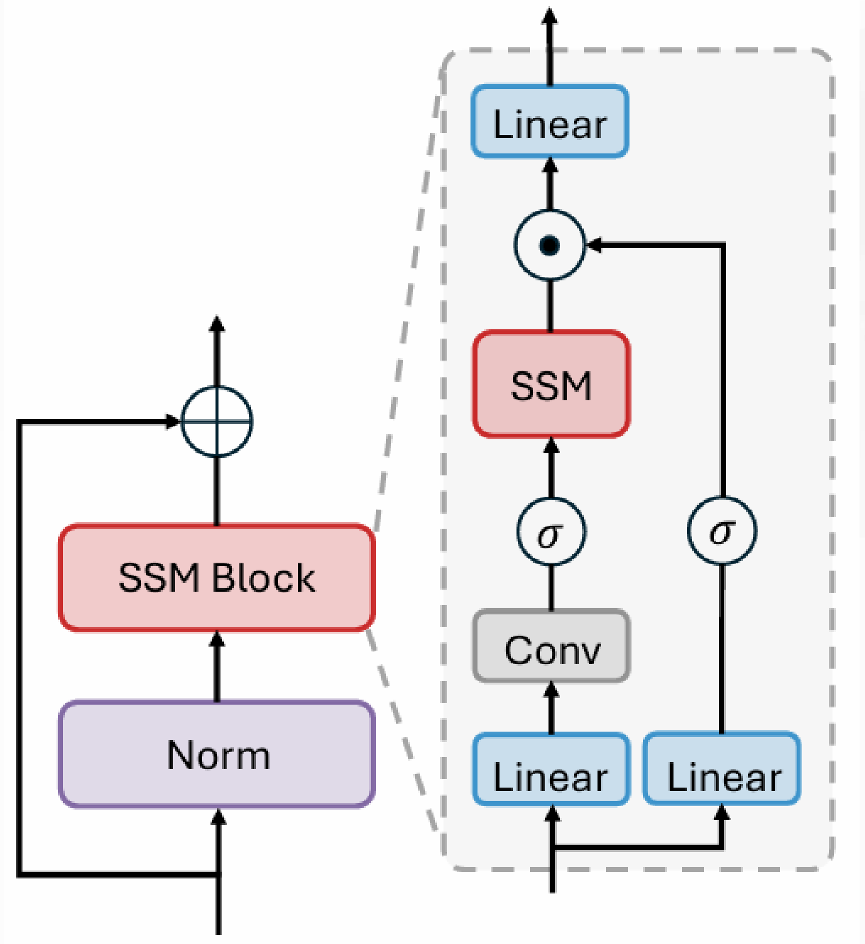
(5) Multi-head design:

- Selective SSM resembles single-head attention
- Linear attention commonly employ multi-head design

(6) Different macro design:



Linear Attention Transformer



Mamba

Mamba v.s. Linear Attention Transformer

paper code

Mamba can be viewed as linear attention Transformer with **six special designs**:

(1) ***input gate***

(2) ***forget gate***

(3) ***shortcut***

(4) ***no attention normalization***

(5) ***single-head design***

(6) ***modified block structure***

Empirical Study

paper code

	#Params	FLOPs	Throughput	Top-1
Baseline	28M	4.5G	1152	77.6
(1) + Input Gate	29M	4.5G	1069	77.8
(2) + Forget Gate	29M	4.8G	743	78.4
(3) + Shortcut	28M	4.5G	1066	77.8
(4) − Normalization	28M	4.5G	1215	72.4
(5) − Multi-head Design	24M	3.9G	1540	73.5
(6) + Block Design	31M	4.8G	1010	80.9

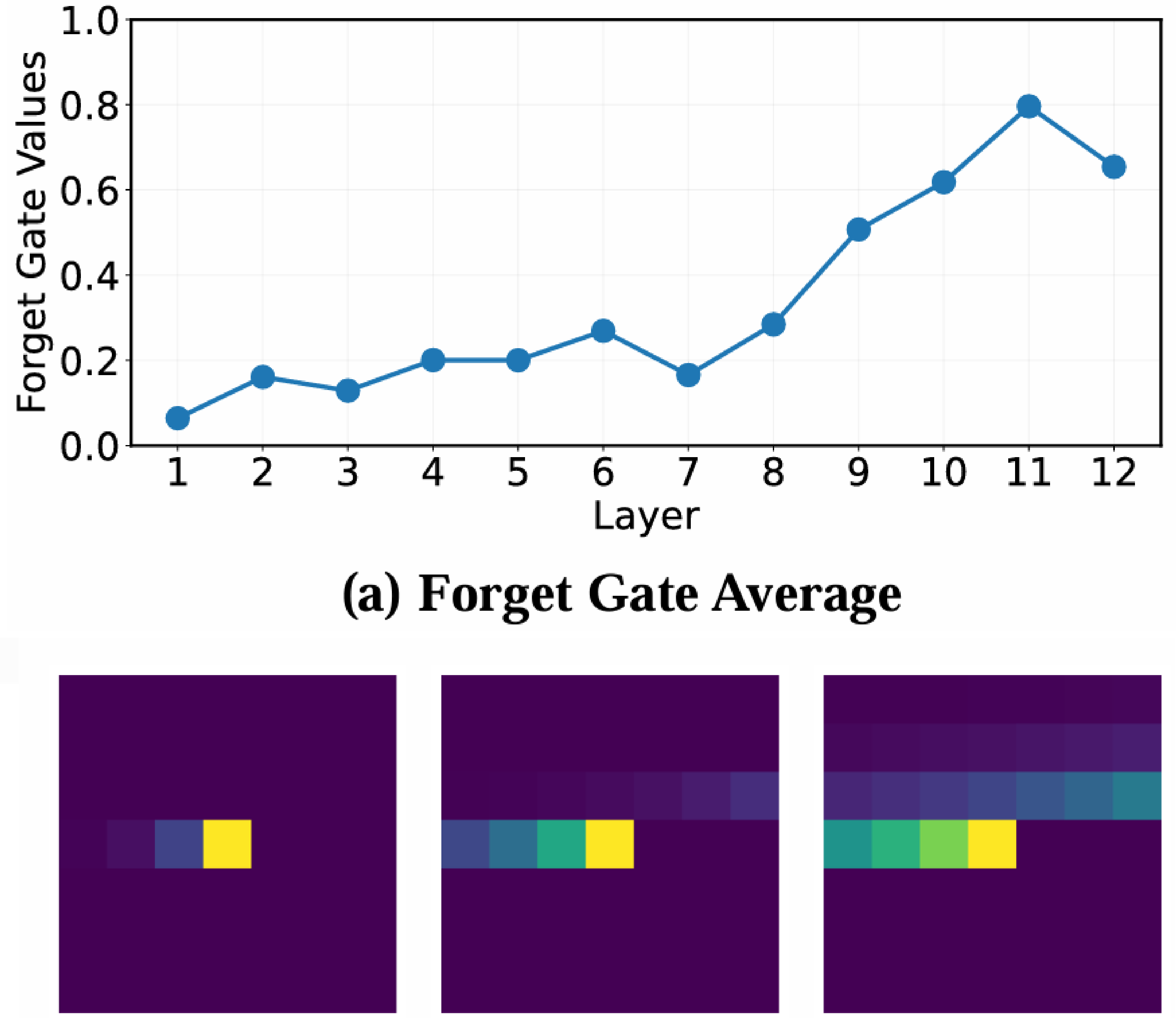
The **forget gate** and **block design** tend to be the core contributors!

Empirical Study

paper code

- The **forget gate** needs *recurrent calculation*, which is not ideal for vision models.
- Proper **positional encoding** can function as the forget gate in vision tasks, while preserving *parallelizable computation*.

	#Params	FLOPs	Throughput	Top-1
Baseline	28M	4.5G	1152	77.6
+ Forget Gate	29M	4.8G	743	78.4
+ APE [8]	30M	4.5G	1132	80.0
+ LePE [7]	28M	4.5G	1074	81.6
+ CPE [4]	28M	4.5G	1099	81.7
+ RoPE [33]	28M	4.5G	1113	80.0



(a) Forget Gate Average

$\tilde{A}_i = 0.2$ $\tilde{A}_i = 0.6$ $\tilde{A}_i = 0.8$

(b) Forget Gate Illustration

Empirical Study

paper code

Based on these findings, we propose a

Mamba-Inspired Linear Attention (MILA) model

by incorporating the merits of Mamba's two key designs

into linear attention.

Empirical Study: ImageNet Classification

paper code

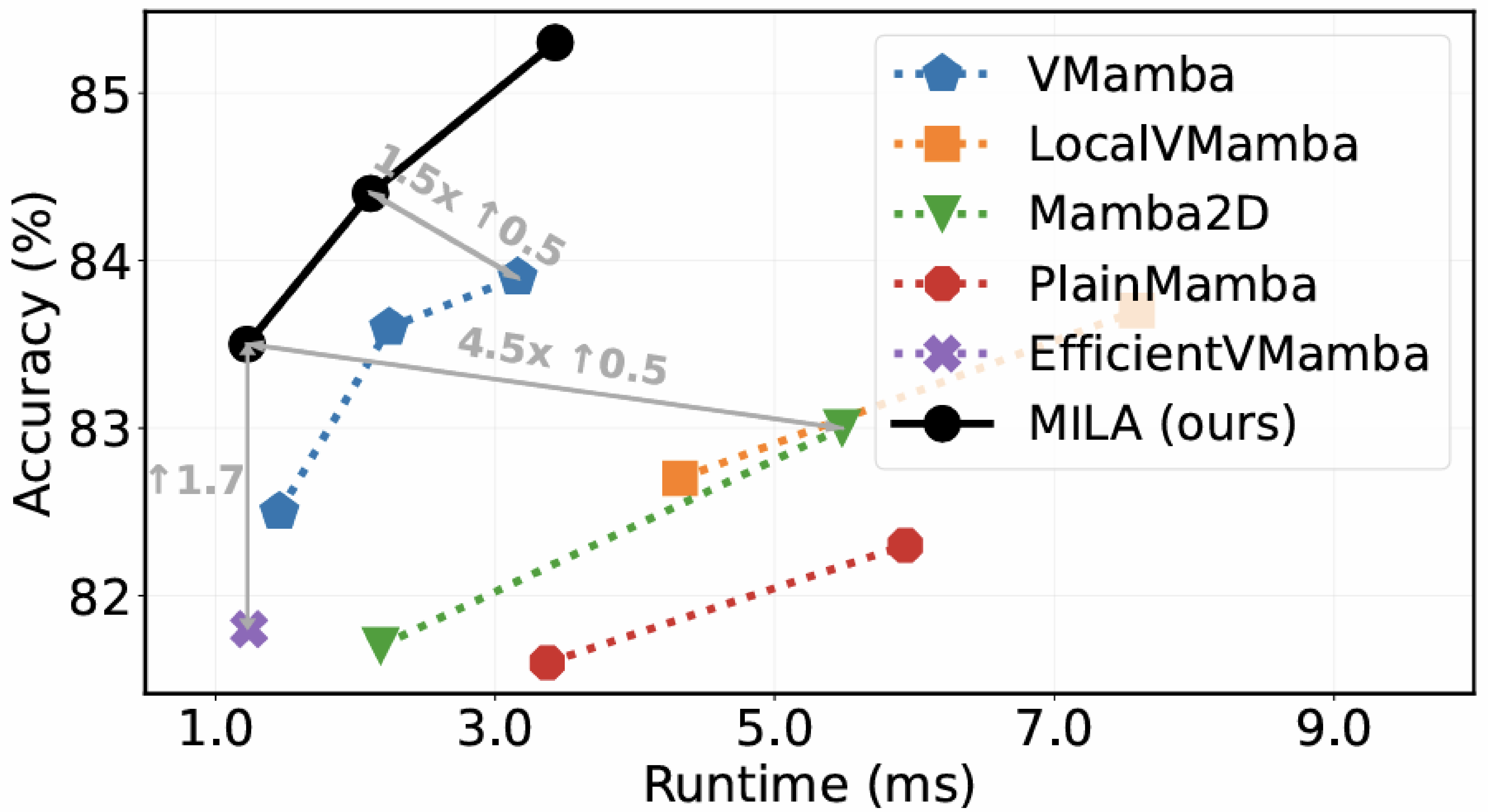
Method	Type	#Params	FLOPs	Top-1
ConvNeXt-T [33]	CNN	29M	4.5G	82.1
MambaOut-T [51]	CNN	27M	4.5G	82.7
Swin-T [32]	Transformer	29M	4.5G	81.3
PVTv2-B2 [44]	Transformer	25M	4.0G	82.0
Focal-T [50]	Transformer	29M	4.9G	82.2
MViTv2-T [28]	Transformer	24M	4.7G	82.3
CSwin-T [9]	Transformer	23M	4.3G	82.7
DiNAT-T [19]	Transformer	28M	4.3G	82.7
NAT-T [20]	Transformer	28M	4.3G	83.2
PlainMamba-L1 [49]	Mamba	7M	3.0G	77.9
Vim-S [57]	Mamba	26M	5.1G	80.3
LocalVim-S [25]	Mamba	28M	4.8G	81.2
PlainMamba-L2 [49]	Mamba	25M	8.1G	81.6
Mamba2D-S [27]	Mamba	24M	–	81.7
EfficientVMamba-B [38]	Mamba	33M	4.0G	81.8
VMamba-T [31]	Mamba	31M	4.9G	82.5
LocalVMamba-T [25]	Mamba	26M	5.7G	82.7
MILA-T	MILA	25M	4.2G	83.5

Method	Type	#Params	FLOPs	Top-1
ConvNeXt-S [33]	CNN	50M	8.7G	83.1
MambaOut-S [51]	CNN	48M	9.0G	84.1
PVTv2-B3 [44]	Transformer	45M	7.9G	83.2
CSwin-S [9]	Transformer	35M	6.9G	83.6
Focal-S [50]	Transformer	51M	9.4G	83.6
MViTv2-S [28]	Transformer	35M	7.0G	83.6
VMamba-S [31]	Mamba	50M	8.7G	83.6
LocalVMamba-S [25]	Mamba	50M	11.4G	83.7
MILA-S	MILA	43M	7.3G	84.4
ConvNeXt-B [33]	CNN	89M	15.4G	83.8
MambaOut-B [51]	CNN	85M	15.8G	84.2
PVTv2-B5 [44]	Transformer	82M	11.8G	83.8
Focal-B [50]	Transformer	90M	16.4G	84.0
CSwin-B	Transformer	78M	15.0G	84.2
NAT-B [20]	Transformer	90M	13.7G	84.3
PlainMamba-L3 [49]	Mamba	50M	14.4G	82.3
Mamba2D-B [27]	Mamba	94M	–	83.0
VMamba-B [31]	Mamba	89M	15.4G	83.9
MILA-B	MILA	96M	16.2G	85.3

Empirical Study: Efficiency

paper

code



Empirical Study: Object Detection

paper code

(b) Mask R-CNN 3x on COCO

Method	Type	#Params	FLOPs	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ConvNeXt-T [33]	CNN	48M	262G	46.2	67.9	50.8	41.7	65.0	44.9
Swin-T [32]	Transformer	48M	267G	46.0	68.1	50.3	41.6	65.1	44.9
PVTv2-B2 [44]	Transformer	45M	309G	47.8	69.7	52.6	43.1	66.8	46.7
FocalNet-T [50]	Transformer	49M	268G	48.0	69.7	53.0	42.9	66.5	46.1
Vmamba-T [31]	Mamba	50M	270G	48.9	70.6	53.6	43.7	67.7	46.8
LocalVMamba-T [25]	Mamba	45M	291G	48.7	70.1	53.0	43.4	67.0	46.4
MILA-T	MILA	44M	255G	48.8	71.0	53.6	43.8	68.0	46.8
ConvNeXt-S [33]	CNN	70M	348G	47.9	70.0	52.7	42.9	66.9	46.2
Swin-S [32]	Transformer	69M	354G	48.2	69.8	52.8	43.2	67.0	46.1
PVTv2-B3 [44]	Transformer	65M	397G	48.4	69.8	53.3	43.2	66.9	46.7
FocalNet-S [50]	Transformer	72M	365G	49.3	70.7	54.2	43.8	67.9	47.4
CSWin-S [9]	Transformer	54M	342G	50.0	71.3	54.7	44.5	68.4	47.7
Vmamba-S [31]	Mamba	70M	384G	49.9	70.9	54.7	44.2	68.2	47.7
LocalVMamba-S [25]	Mamba	69M	414G	49.9	70.5	54.4	44.1	67.8	47.4
MILA-S	MILA	63M	319G	50.5	71.8	55.2	44.9	69.1	48.2

paper code

Take-away Messages

- ✓ We reveal the *surprisingly close relationship* between the powerful Mamba and subpar linear attention Transformer
- ✓ We identify that the *forget gate* and *block design* are the core factors behind Mamba’s success
- ✓ We propose ***Mamba-Inspire Linear Attention (MILA)*** model, enjoying *high performance* while maintaining *parallel computation* and *fast inference speed*.

paper

code

Thank you!

Contact: hdc23@mails.tsinghua.edu.cn