State-Space Models (SSMs)

Rowel Atienza, PhD

University of the Philippines

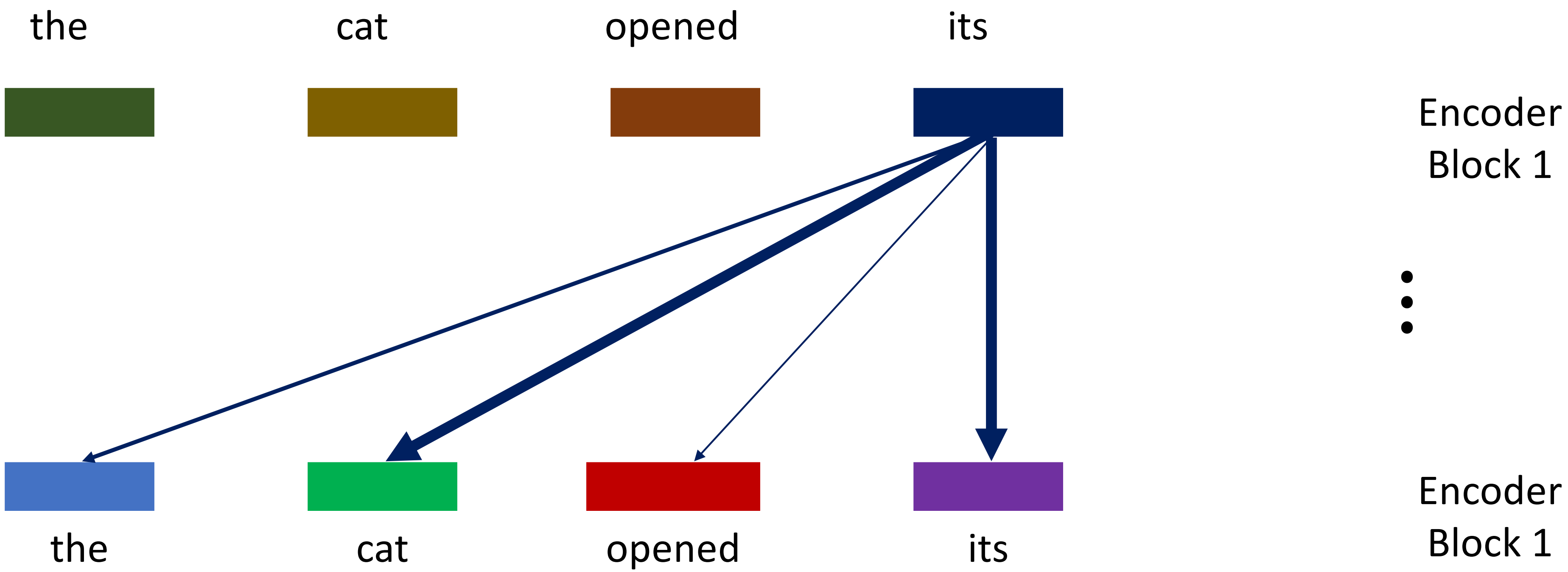
github.com/roatienza

2024

Why SSMs

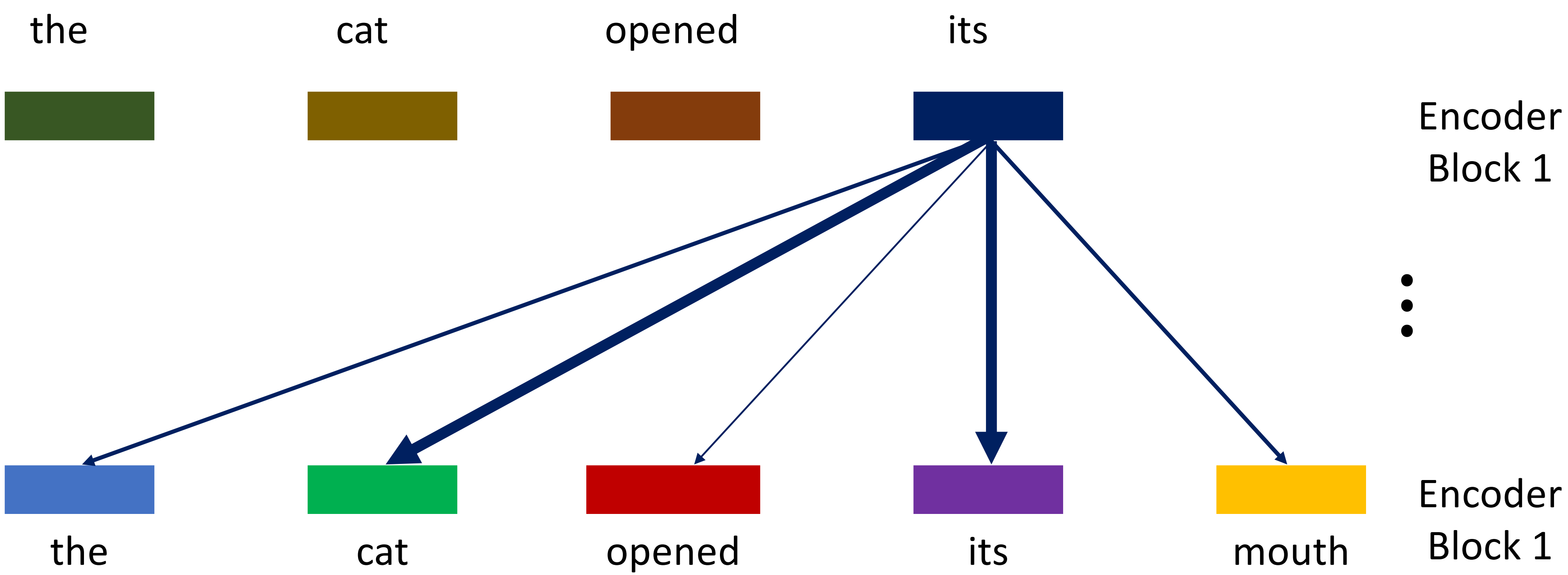
- Transformers has $O(L^2)$ inference complexity with sequence length though $O(1)$ training complexity
- RNNs has $O(1)$ inference complexity with sequence length but $O(L^2)$ training complexity

Transformers: Any token can attend to any other token



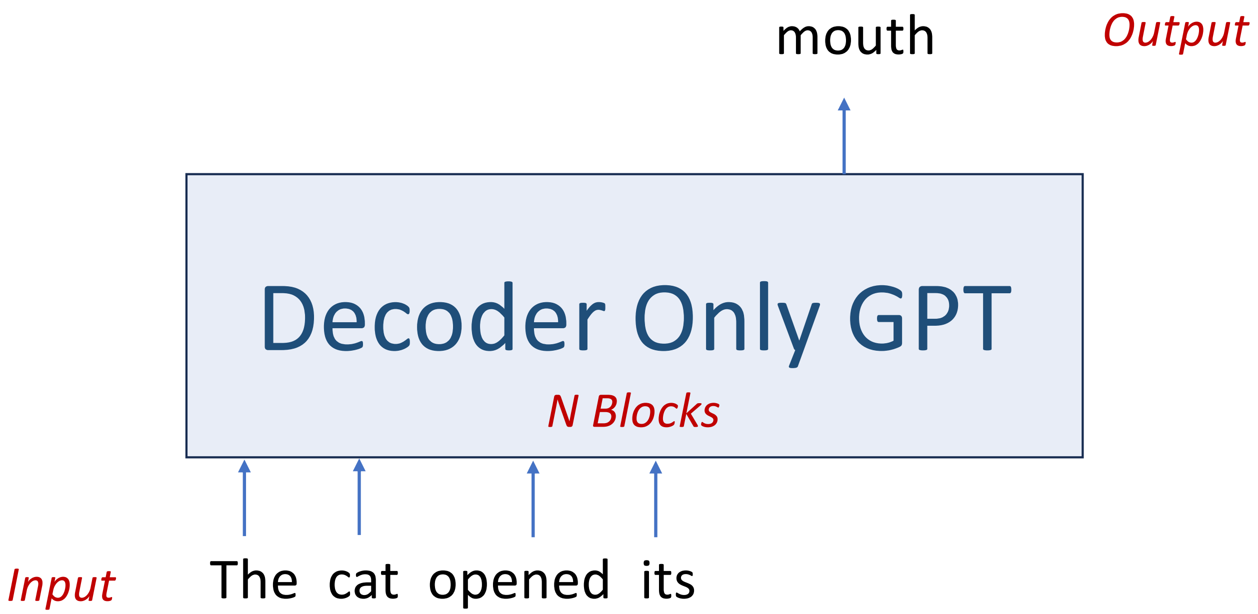
Attention as measured by the width of the arrow

What comes after “its”: Next token prediction



Attention as measured by the width of the arrow

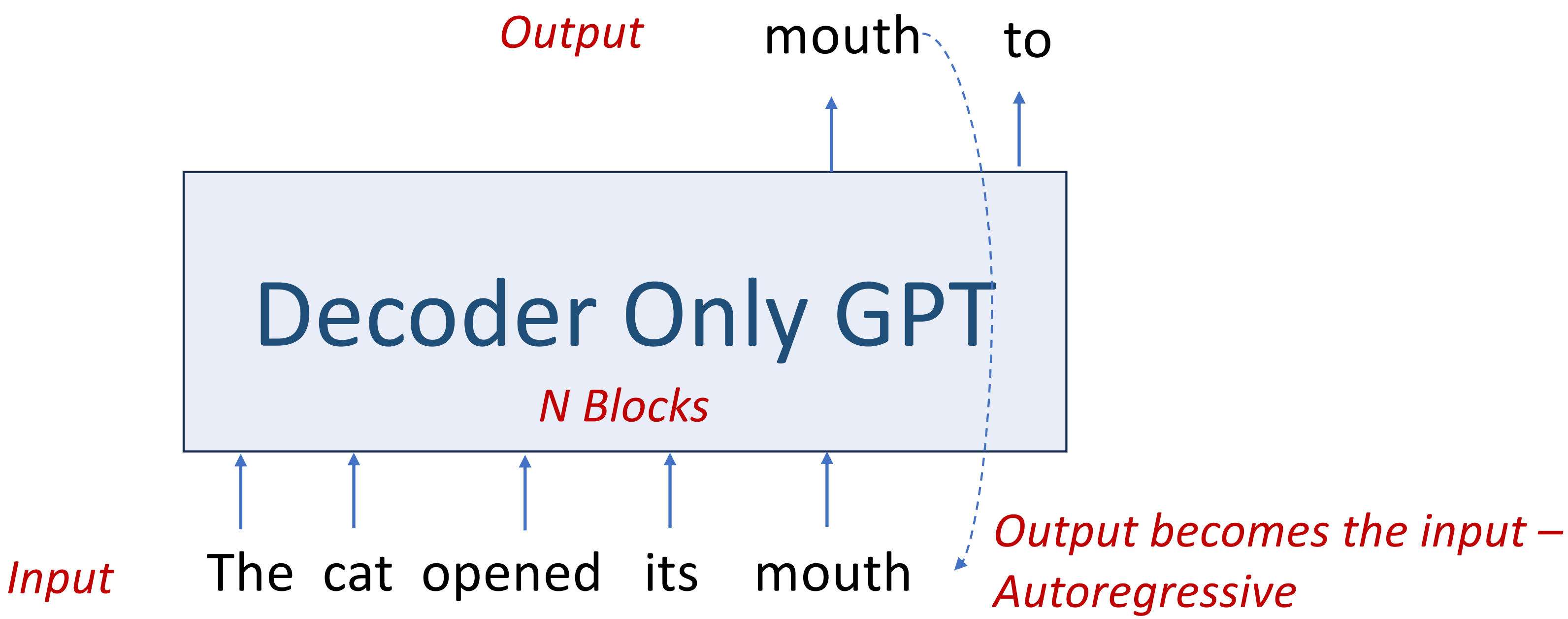
GPT – Generative Pre-trained Transformers



Attention Matrix

	The	cat	opened	its	mouth
The					
cat					
opened					
its					
mouth					

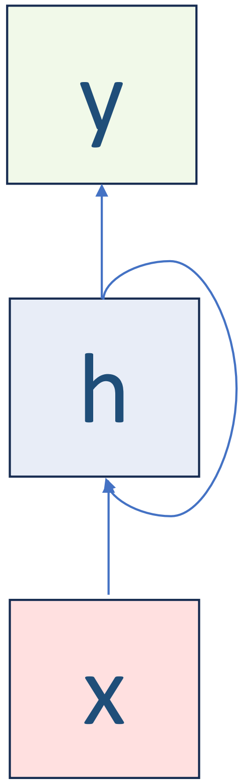
GPT – L^2 complexity



Transformers – Fast Training Slow Inference

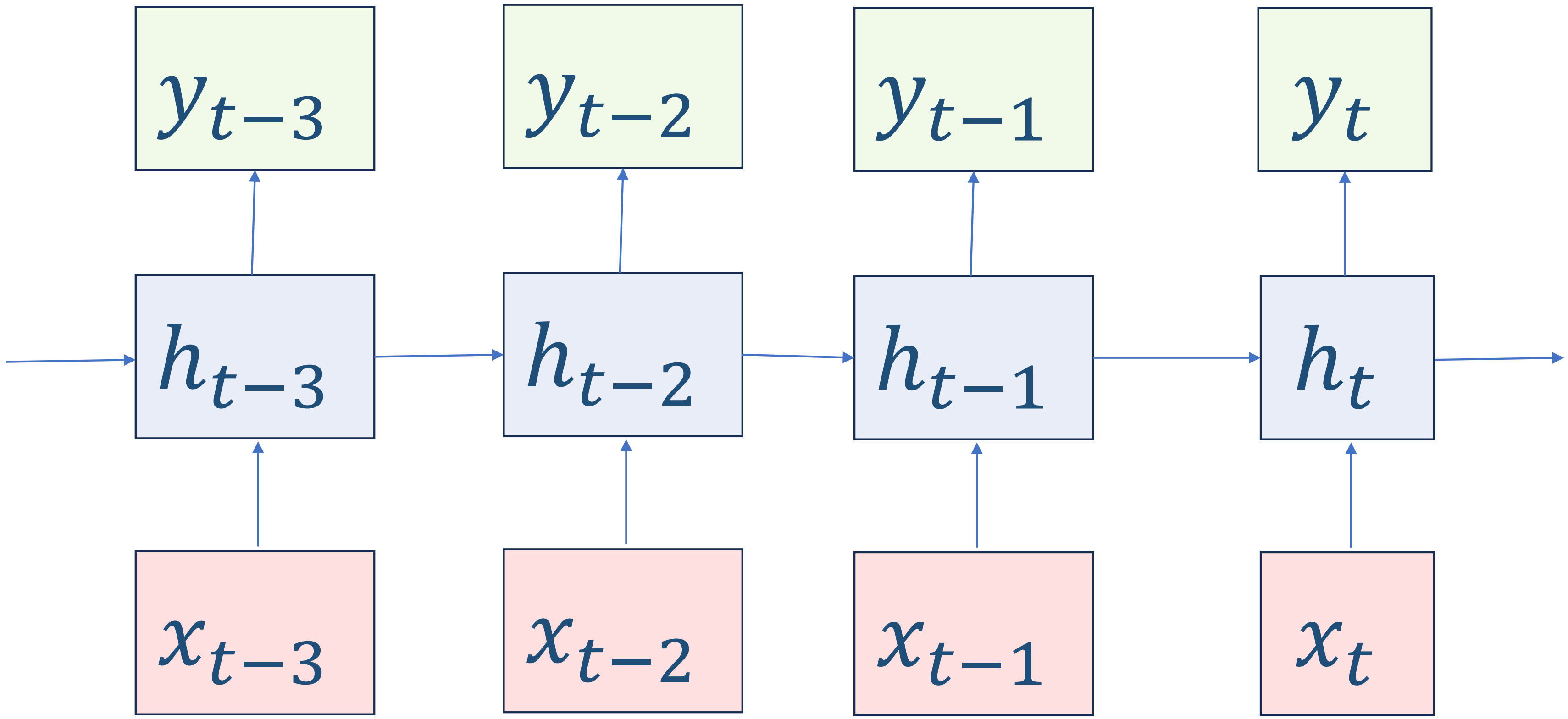
	Training	Inference
Transformers	Fast	Slow

Recurrent Neural Networks - RNNs



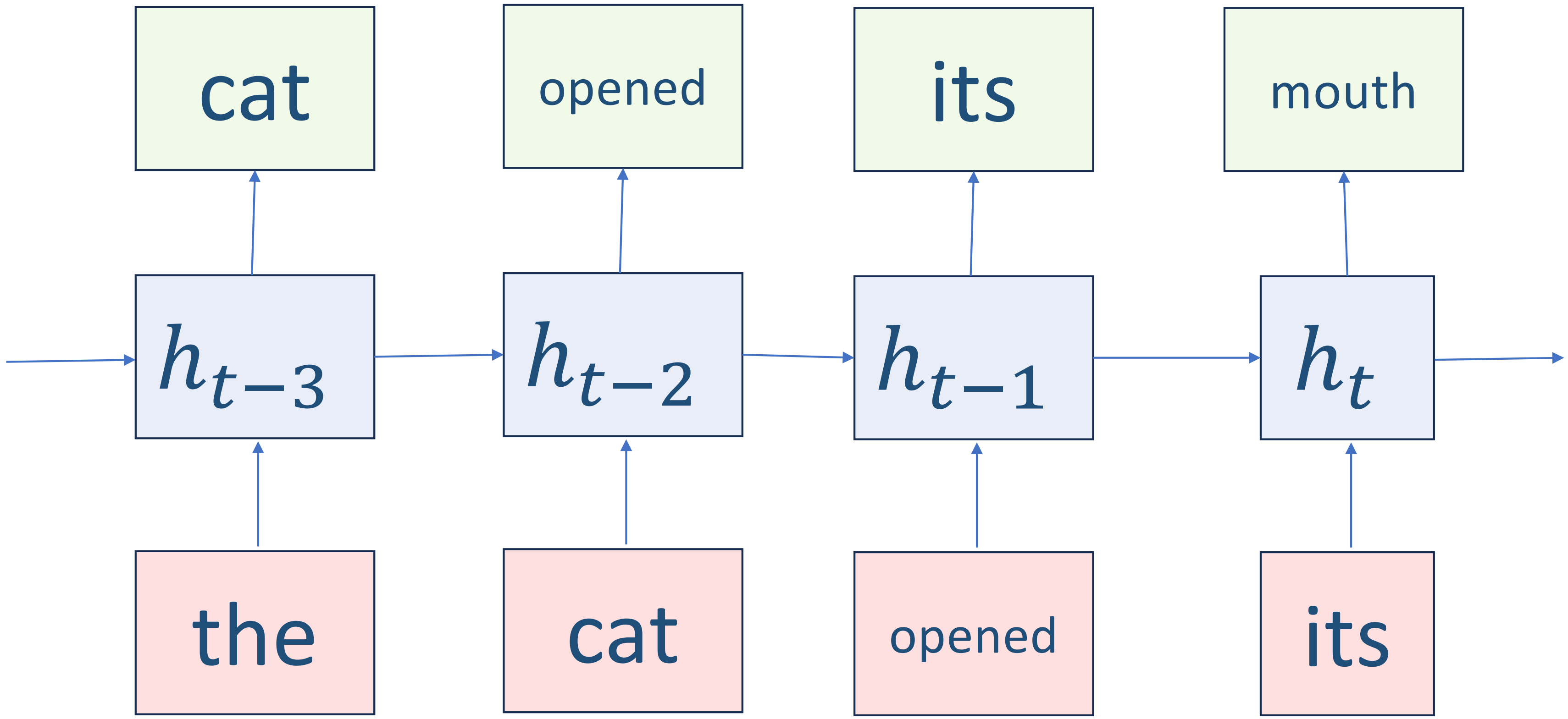
RNNs Unfolded

Output is dependent on previous state only



RNNs Unfolded

Output is dependent on previous state only



Transformers vs RNNs

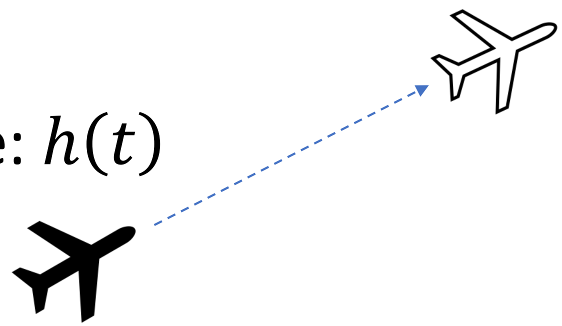
	Training	Inference
Transformers	Fast	Slow
RNNs	Slow	Fast

State Space Models

SSMs

	Training	Inference
SSMs	Fast	Fast

Key Idea – Output is a function of Input and Hidden State

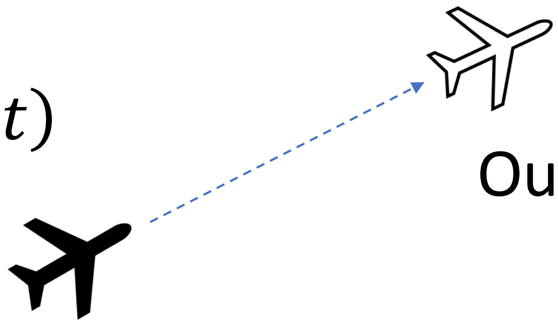


SSM in Continuous Time

State Space Equations

State Update: $h'(t) = Ah(t) + Bx(t)$

State: $h(t)$



Output: $y(t) = Ch(t) + Bx(t)$

Input: $x(t)$

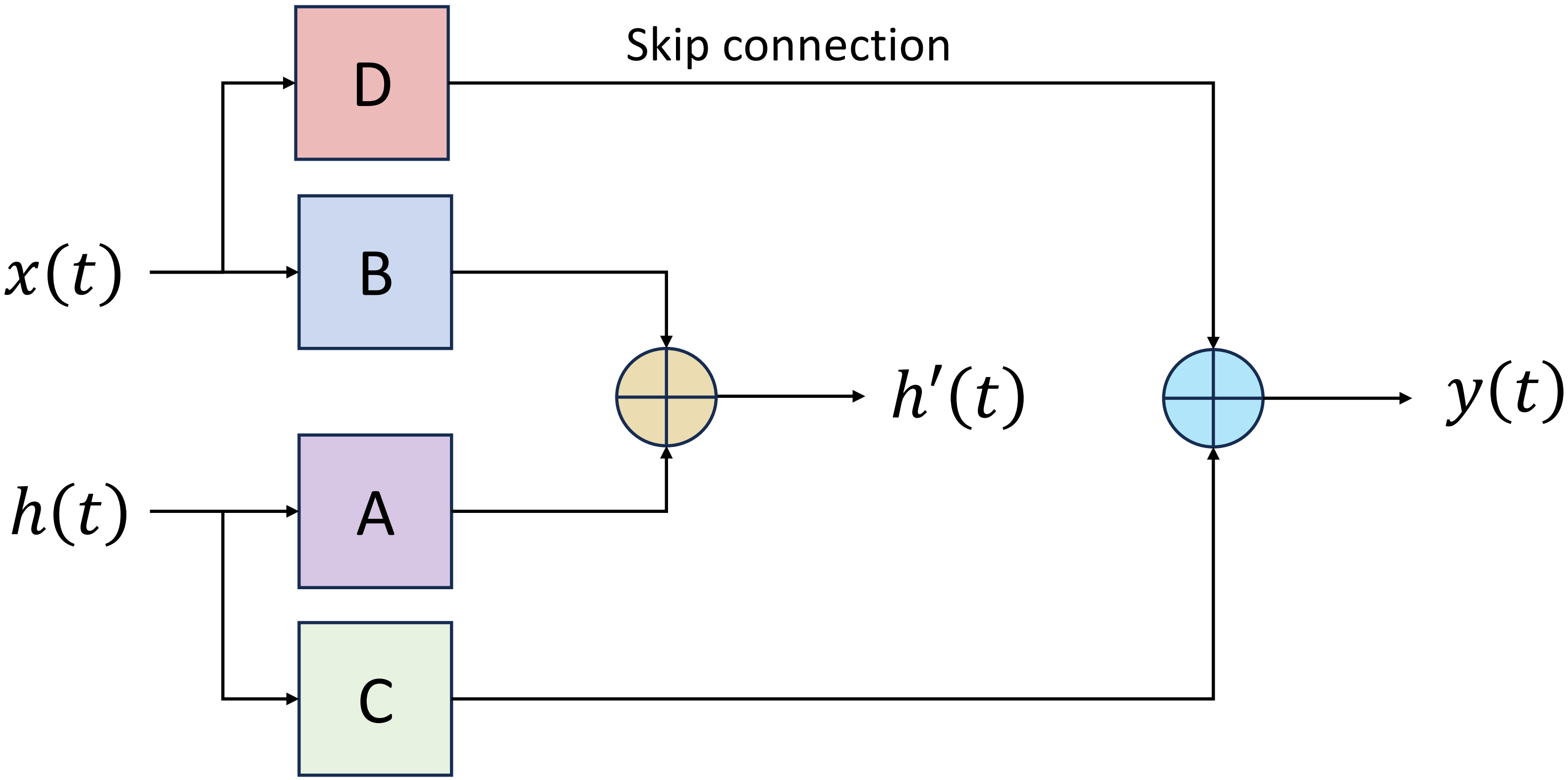
State Update

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

How the current state affects the next state

How the current input affects the next state

Architecture



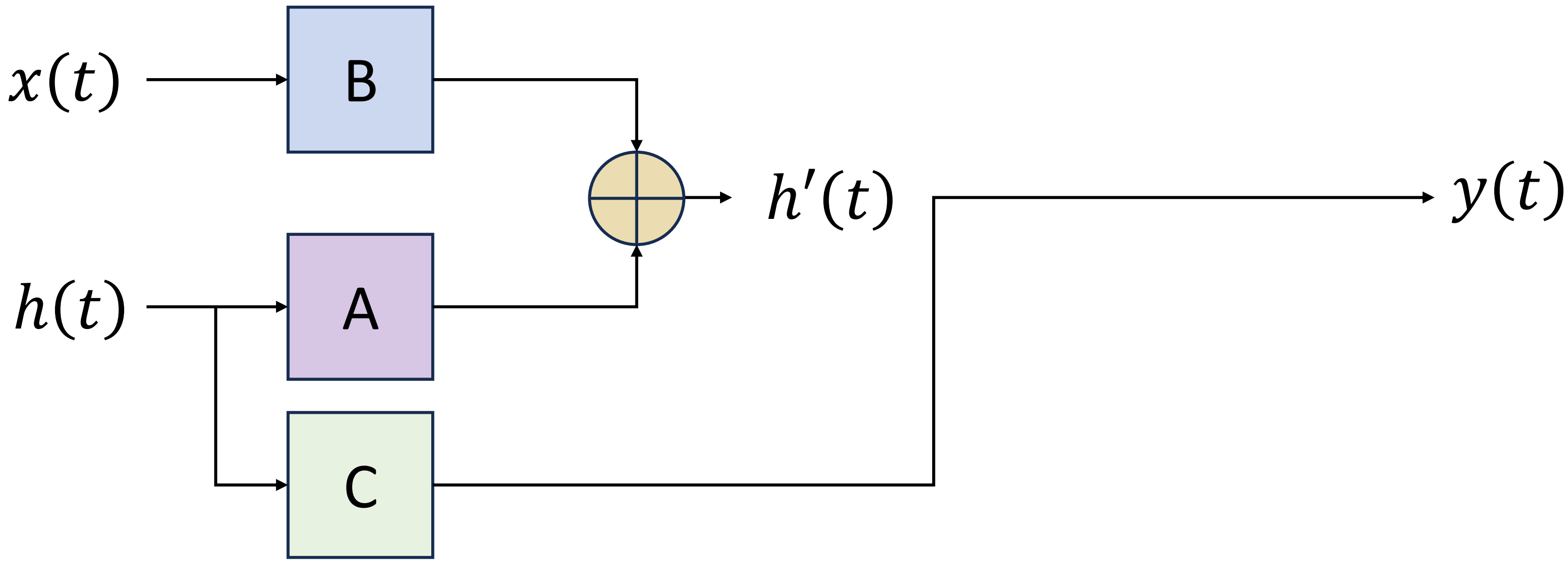
Output

$$\mathrm{y}(t) = Ch(t) + Dx(t)$$

How the current state affects the output

How the current input affects the output

Simplified Architecture w/o Skip Connection



SSM in Discrete Time

Sequence to Sequence - Translation

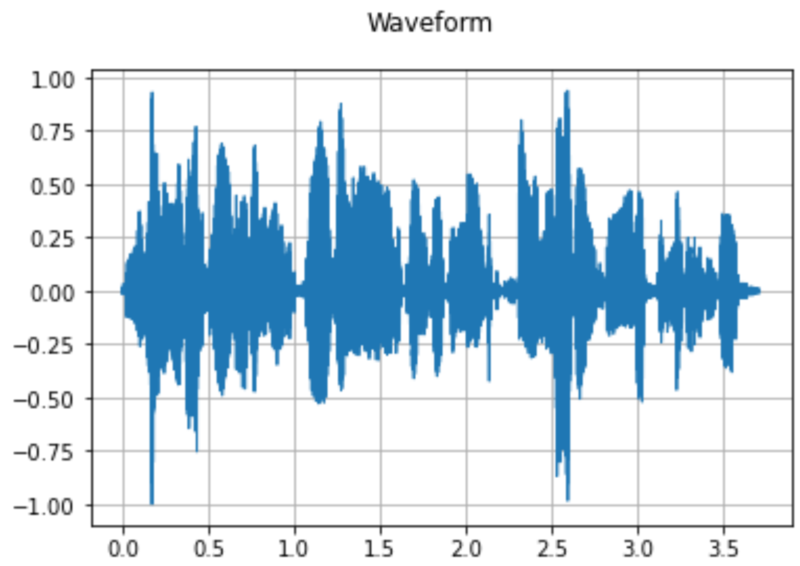
How are you? $\longrightarrow$ Kumusta ka?

Discrete Tokens *Discrete Tokens*

Sequence to Sequence - Text to Speech

the association was organized under the most promising auspices

Discrete Tokens

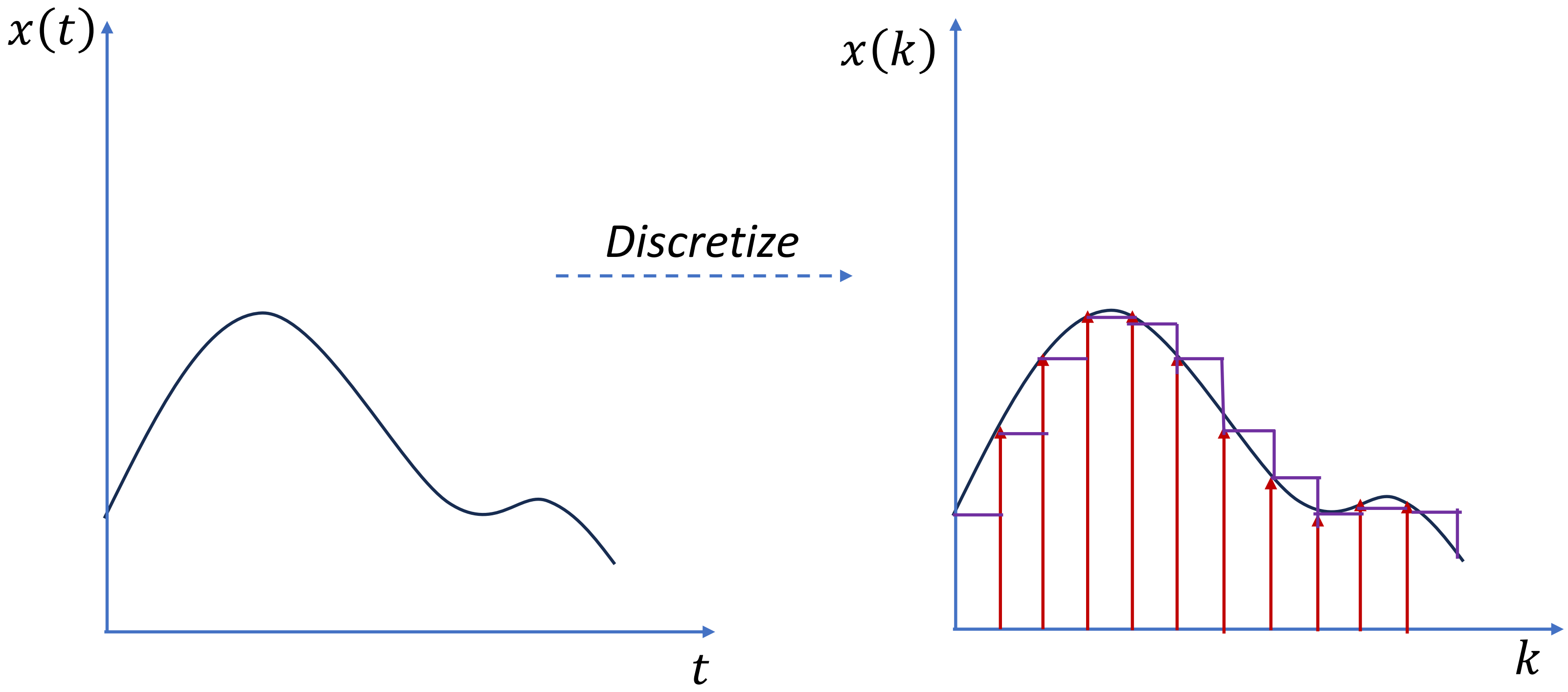


Discrete waveform signal

Continuous to Discrete State Space

- By default, SSM applies to a continuous system (analog)
- Our problems are mostly discrete (digital)
- Solution : Discretize the SS equation

Zero-Order Hold (ZOH) Technique



Discretized Matrices

$$\bar{A} = e^{\Delta A}$$

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

https://en.wikipedia.org/wiki/Discretization#discrete_function

State Update – Recurrent Operation

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

How the current state affects the next state

How the current input affects the next state

Discretized Matrices

$$\bar{C} = C$$

$$\bar{D} = D$$

Output

$$y_k = \bar{C}h_k + \bar{D}x_k$$

How the current state affects the output

How the current input affects the output

Output – No Skip Connection

$$y_k = \bar{C} h_k$$

How the current state affects the output

SSMs using Convolution Operation

Output (No skip connection), Using x_k

$$y_k = \bar{C} h_k$$

$$y_k = \bar{C}(\bar{A} h_{k-1} + \bar{B} x_k)$$

$$y_k = \bar{C}\bar{A} h_{k-1} + \overline{CB} x_k$$

Output using x_{k-1} and x_k

$$y_k = \bar{C}\bar{A}(\bar{A}h_{k-2} + \bar{B}x_{k-1}) + \overline{CB}x_k$$

$$y_k = \bar{C}\bar{A}\bar{A}h_{k-2} + \overline{CAB}x_{k-1} + \overline{CB}x_k$$

Output using x_{k-n} to x_k

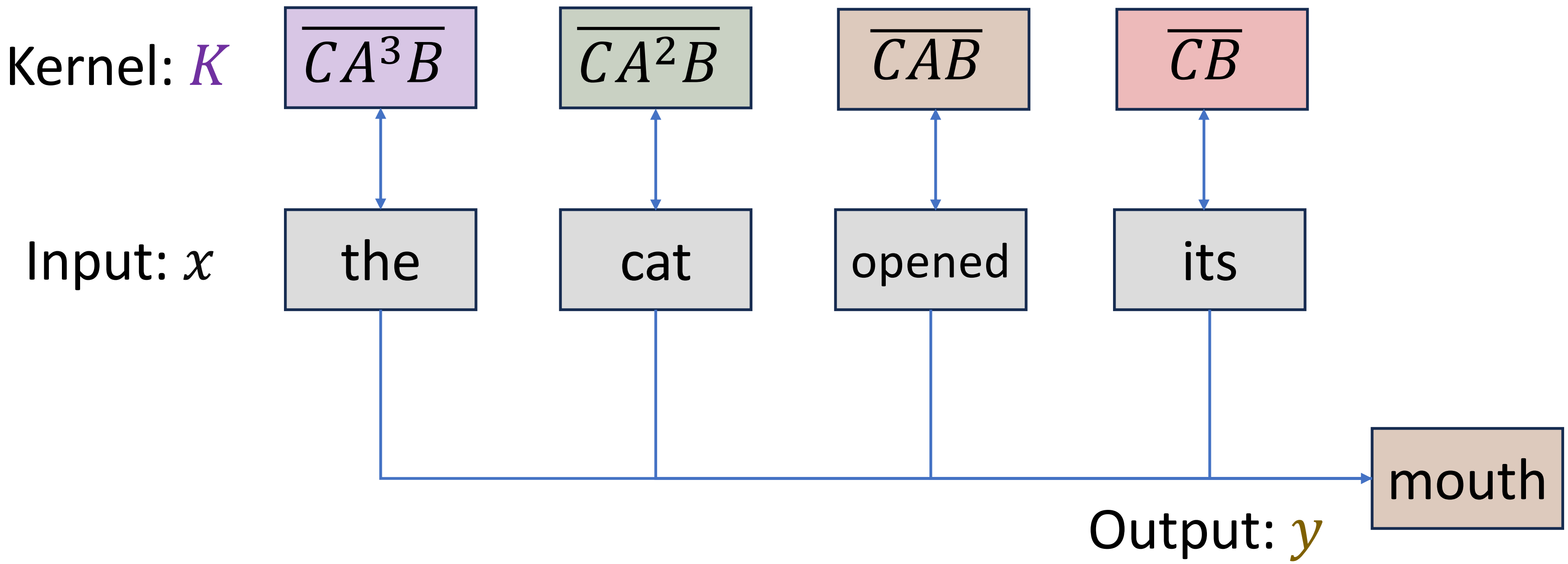
$$y_k = \bar{C}\bar{A}^{n+2}h_{k-n-1} +$$

$$\underbrace{\overline{CA^nB}x_{k-n} + \cdots + \overline{CAB}x_{k-1} + \overline{CB}x_k}_{\text{Kernel: } K}$$

Output as product of convolution

$$y_k \approx x * K$$

Kernel Convolution



SSMs

	Training	Inference
SSMs	Convolution	Recurrent

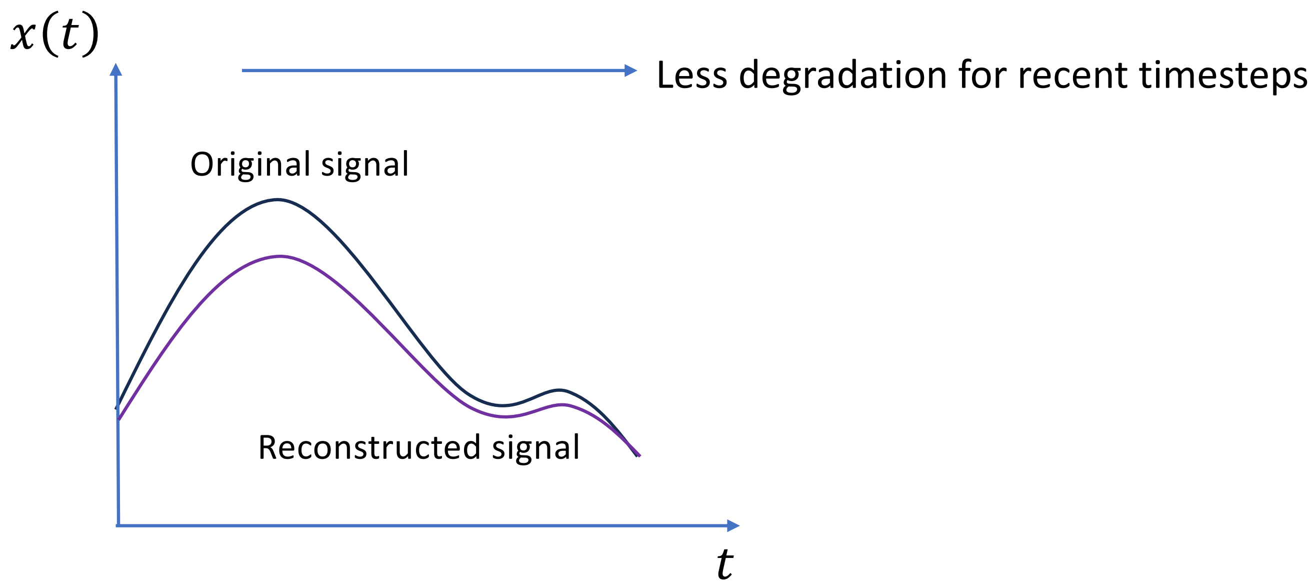
Matrix $\bar{A}$

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

$\bar{A}$: Captures all previous information to build a new state.
How do we create $\bar{A}$ that can retain a large context (memory)?

Hungry Hungry HiPPO!

- HiPPO - High-order Polynomial Projection Operator



HiPPO

$$\overline{A_{nk}} = \begin{cases} (2n+1)^{\frac{1}{2}}(2k+1)^{\frac{1}{2}} & below\ diagonal \\ n+1 & diagonal \\ 0 & above\ diagonal \end{cases}$$

HiPPO - Example

$$\bar{A} = \begin{array}{cc} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 3 & 3 & 0 \\ 1 & 3 & 5 & 4 \end{bmatrix} & k \\ n & \end{array}$$

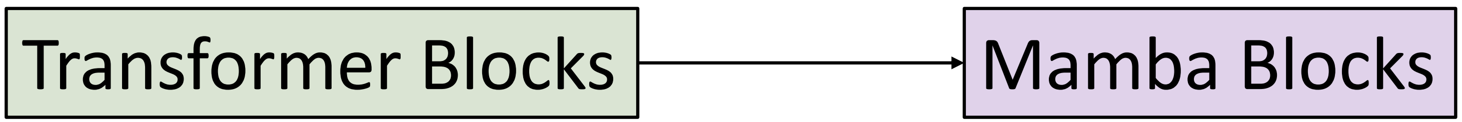
Structured State Space for Sequences (S4)

- State Space Models
- HiPPO for handling long-range dependencies
- Discretization for creating recurrent and convolution representations

S4 is unable to focus on specific inputs that are relevant to the task

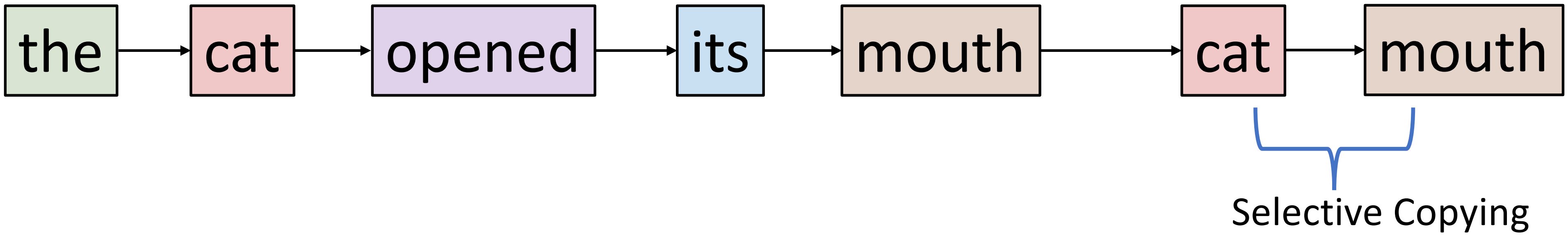
Mamba

- A selective scan algorithm, which allows the model to filter (ir)relevant information
- A hardware-aware algorithm that allows for efficient storage of (intermediate) results through parallel scan, kernel fusion, and recomputation.



Selective Copying

- SSM performs poorly in this task since it is Linear Time Invariant (LTI)
  - Same $\bar{A}, \bar{B}$ and $\bar{C}$
- Result - SSM cannot perform *content-aware reasoning*



Induction Heads

- SSM performs poorly induction heads or reproducing patterns in the input because it is LTI
  - Same $\bar{A}, \bar{B}$ and $\bar{C}$
- Result - SSM cannot perform *in-context-learning*

Q: to + day = ? → A: today

Q: night + time = ? $\xrightarrow{\text{Induction heads}}$ A: nighttime

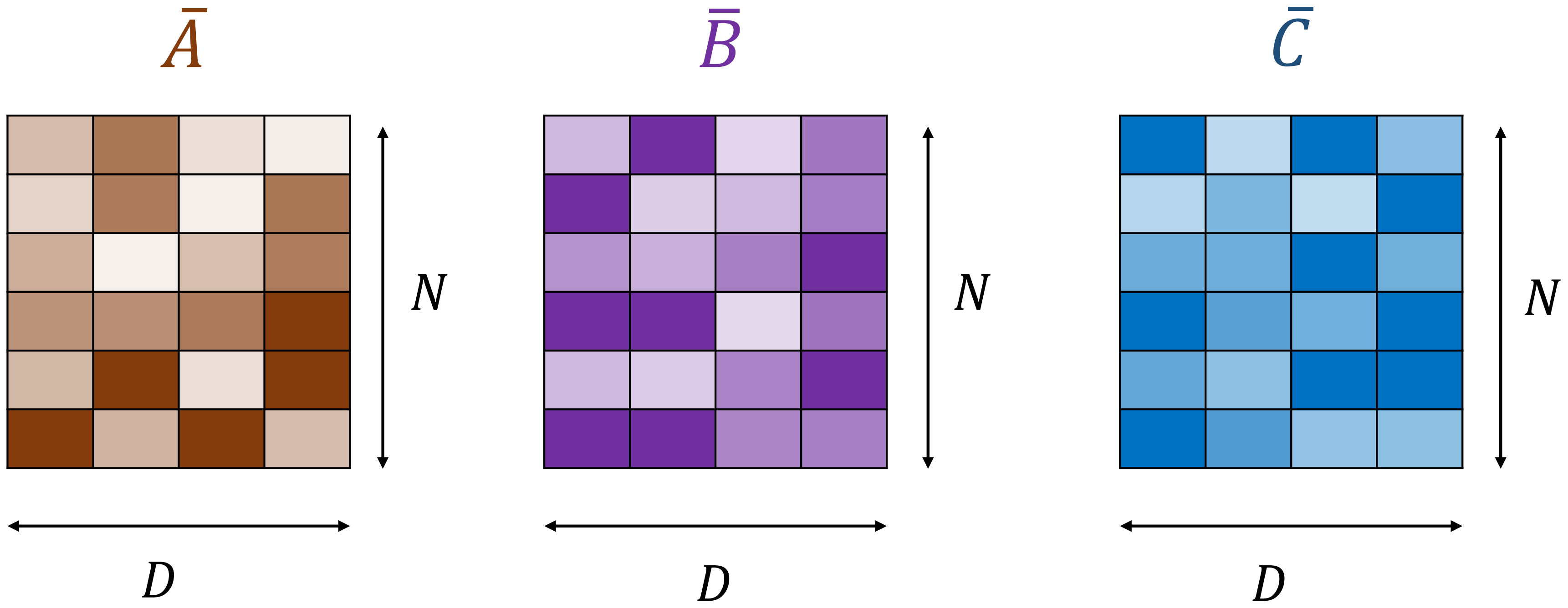
State Update – Recurrent Operation

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

$$y_k = \bar{C}h_k$$

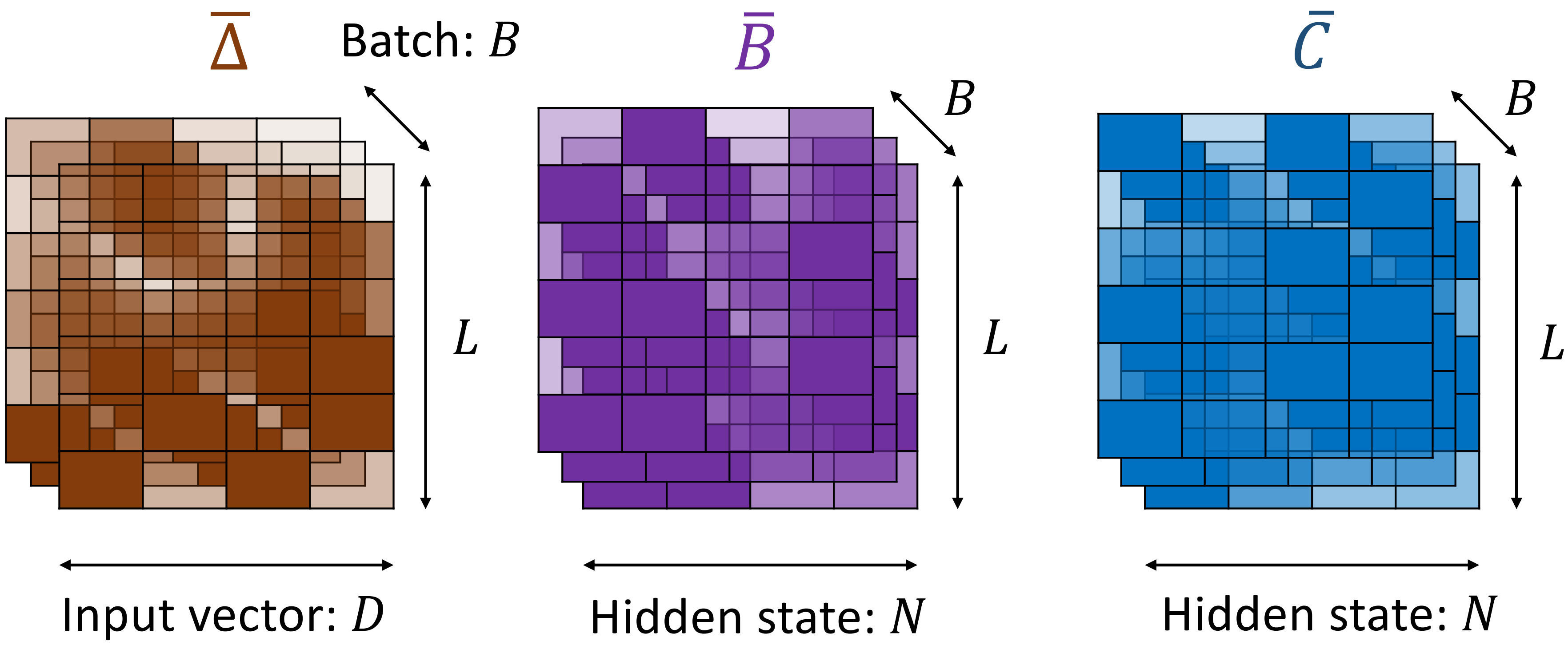
Constant regardless of input

S4 $\bar{A}$, $\bar{B}$ and $\bar{C}$ Matrices are Input Independent



Static: D & N

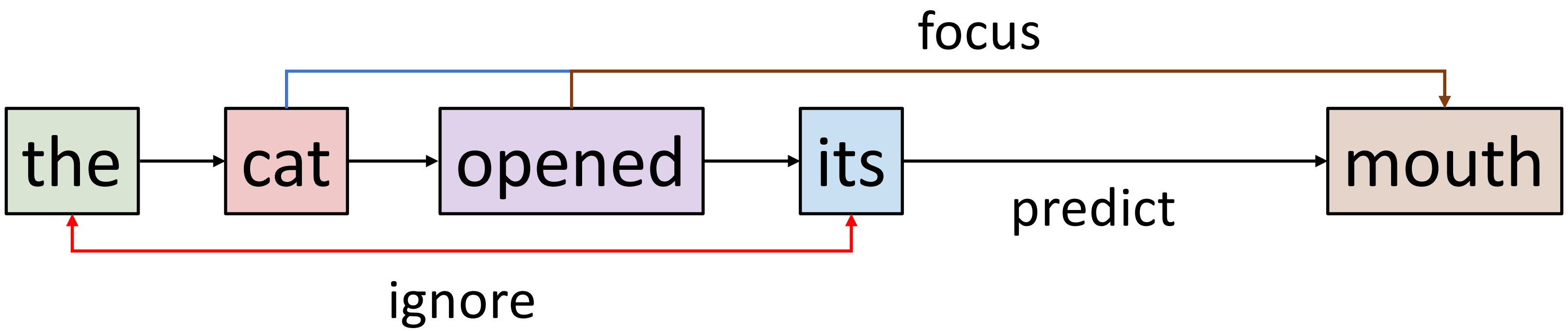
Mamba $\bar{\Delta}$, $\bar{B}$ and $\bar{C}$ Matrices are Input Dependent



Static: $\bar{A}$

Size of $\bar{\Delta}$

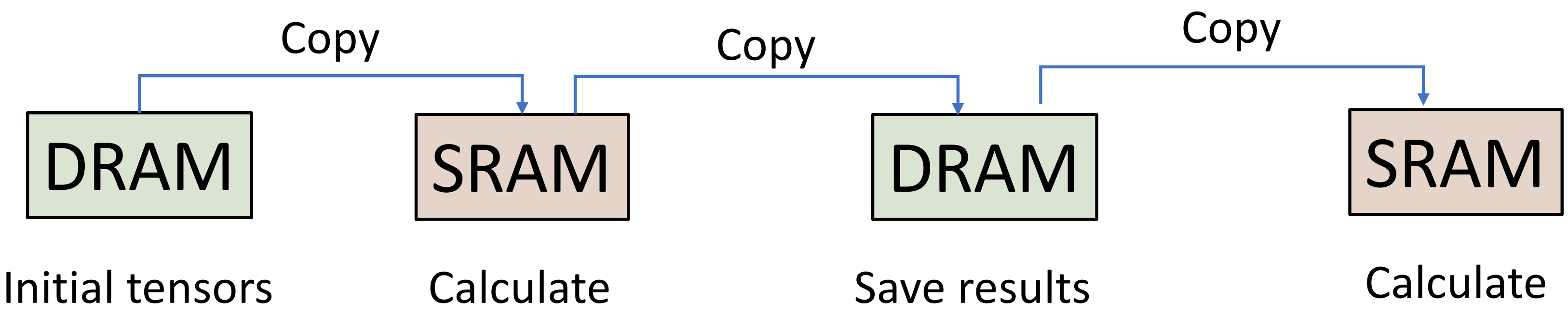
- Small $\bar{\Delta}$ - focus on context
- Large $\bar{\Delta}$ - focus on recent tokens



Hardware Optimization

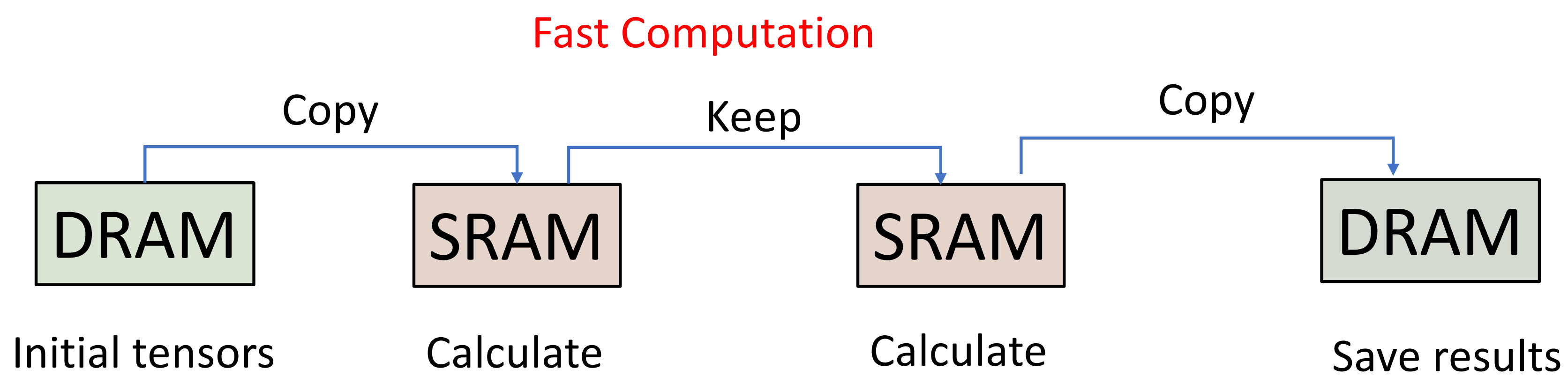
- GPU SRAM – Small and Fast
- GPU DRAM – Large and Slow

Slow Computation



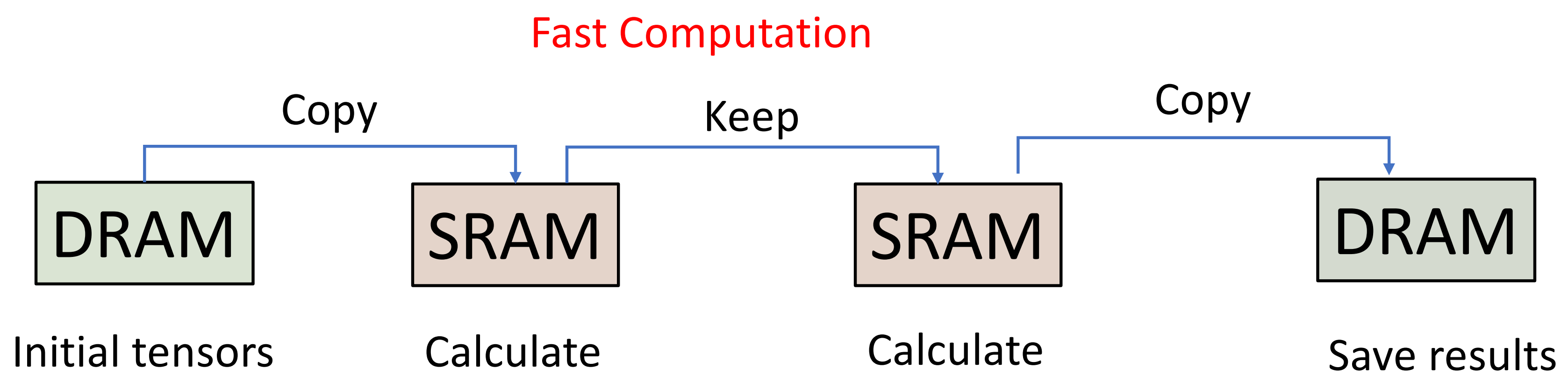
Hardware Optimization

- Keep all computations within GPU SRAM and copy only when done.
- Re-ordering of operation using a new kernel



Hardware Optimization

- Keep all computations within GPU SRAM and copy only when done.
- Re-ordering of operation using a new kernel



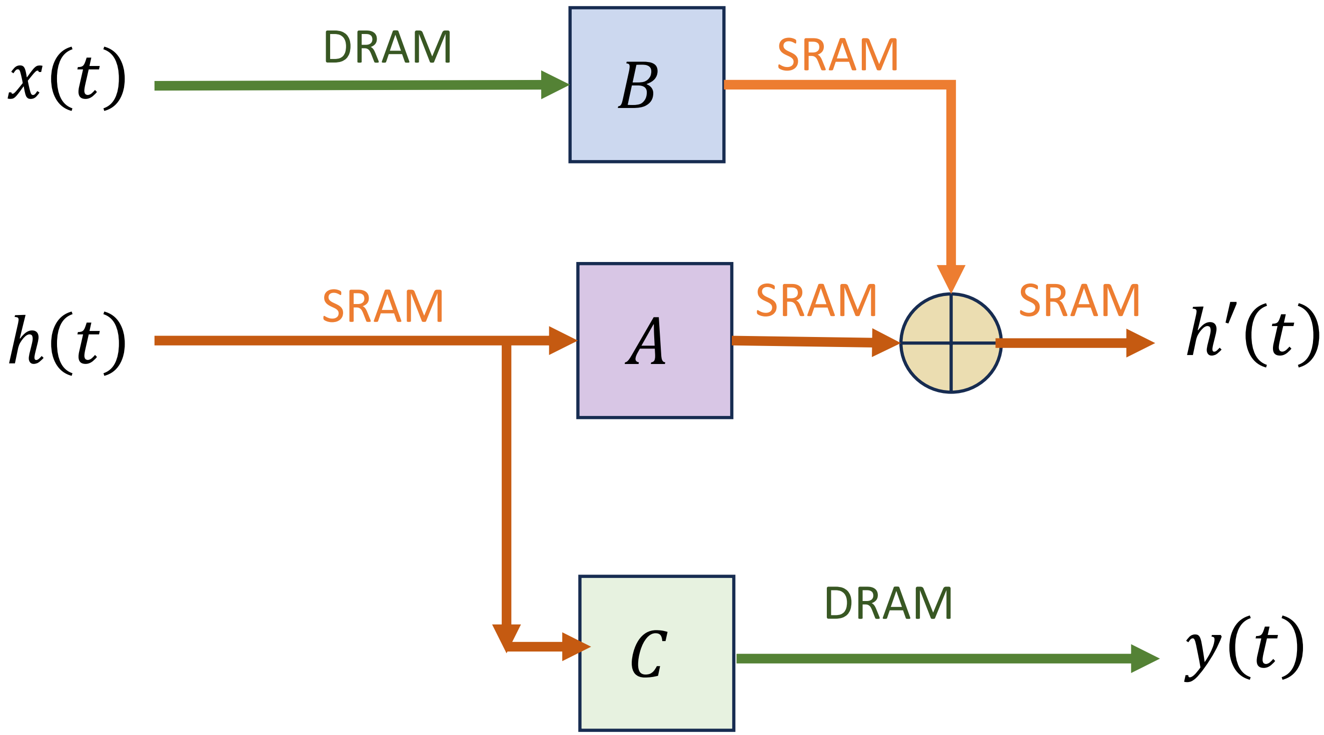
Hardware Optimization

Can be in slow DRAM

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

h_k: Keep in fast SRAM

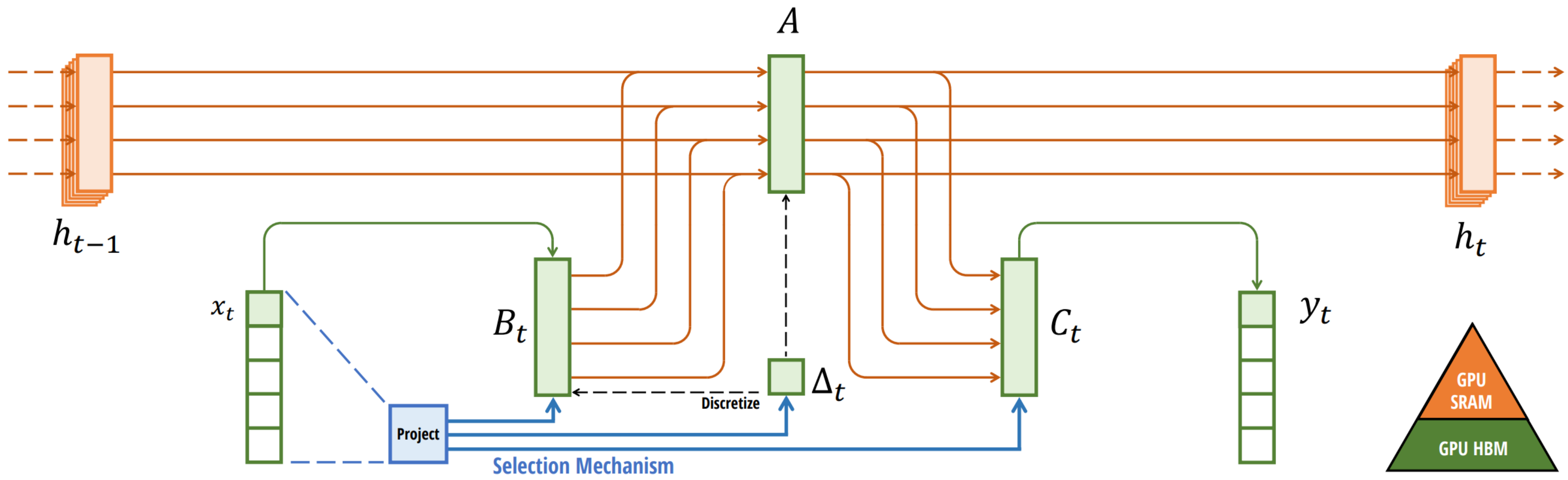
Simplified Architecture w/o Skip Connection



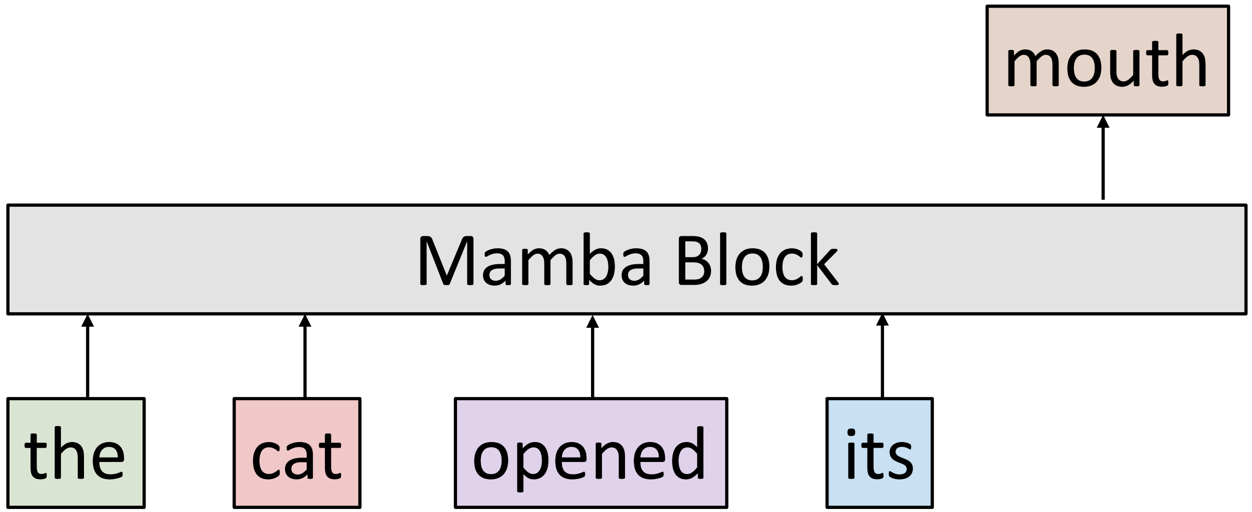
SSM Kernel

- Discretization step with step size Δ
- Selective scan algorithm
- Multiplication with C

Selective SSM (S6) or Mamba Block

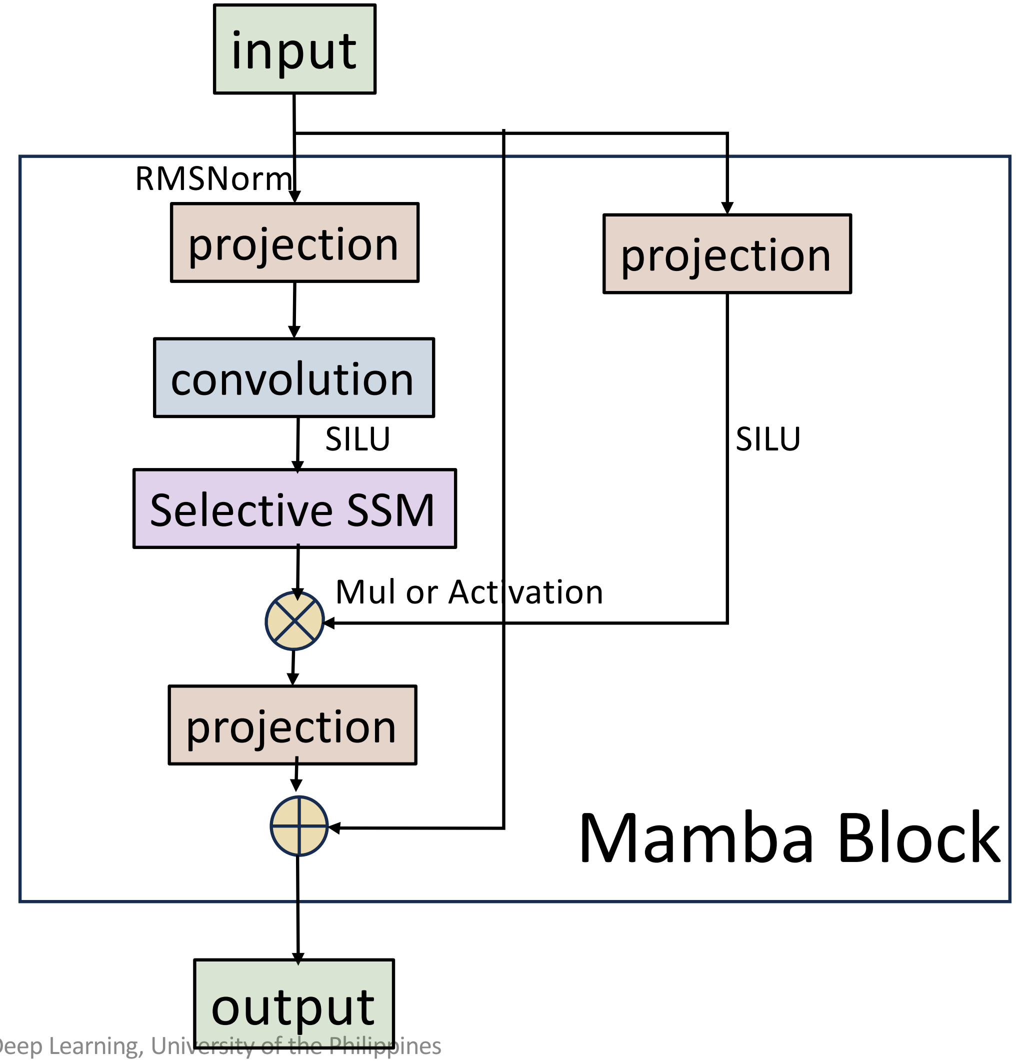


Mamba Decoder for Autoregressive Modelling



Mamba Block

N Blocks



References

- A Visual Guide to Mamba and State Space Models - https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mamba-and-state
- Mamba code and paper - https://github.com/state-spaces/mamba

End