Mambda

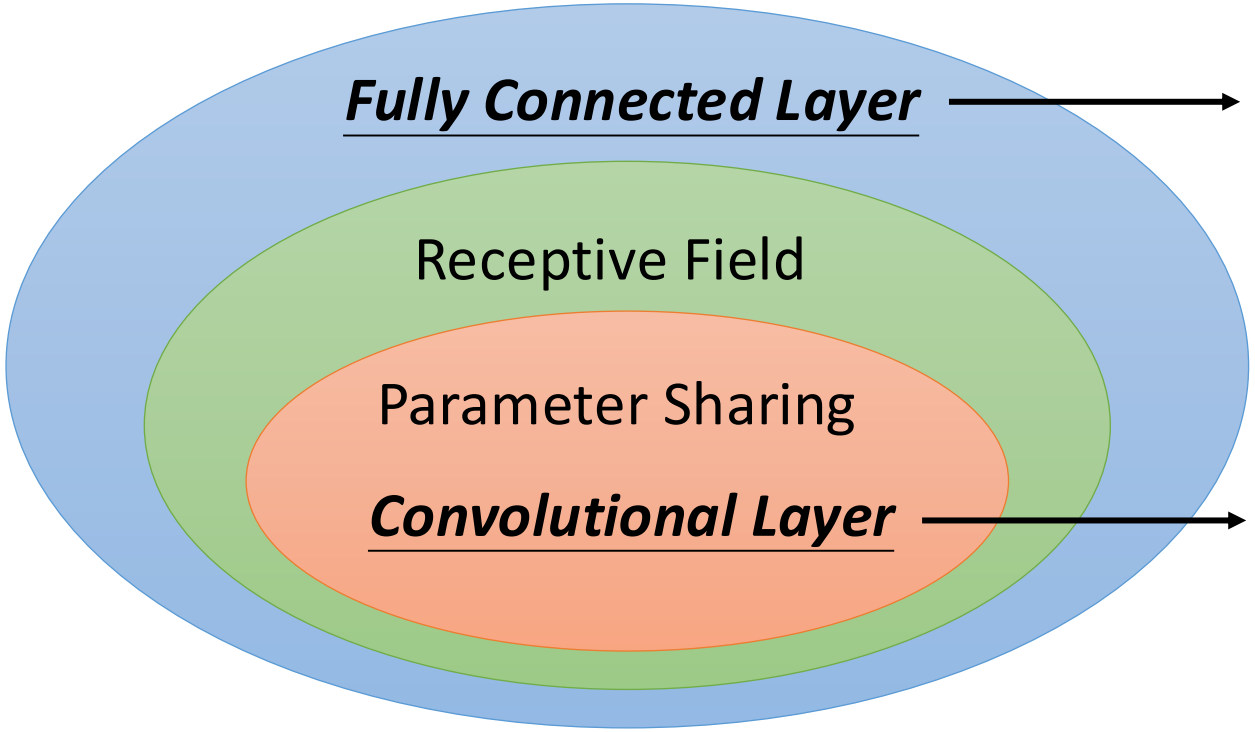
Qiang Sun

University of Toronto

Credits to Hung-yi Lee.

Every architecture has its advantages

- CNN's advantages ?



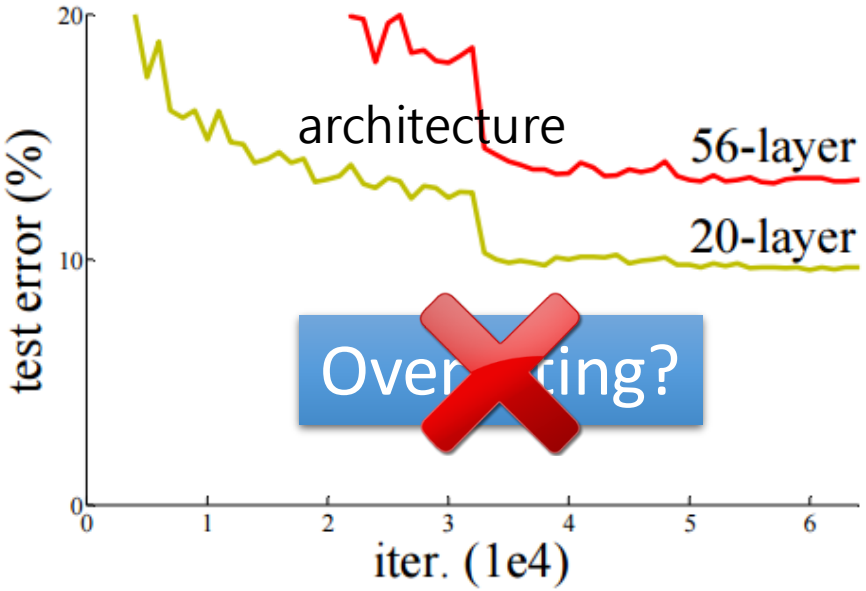
Jack of all trades, master of none

Larger model bias (for image)

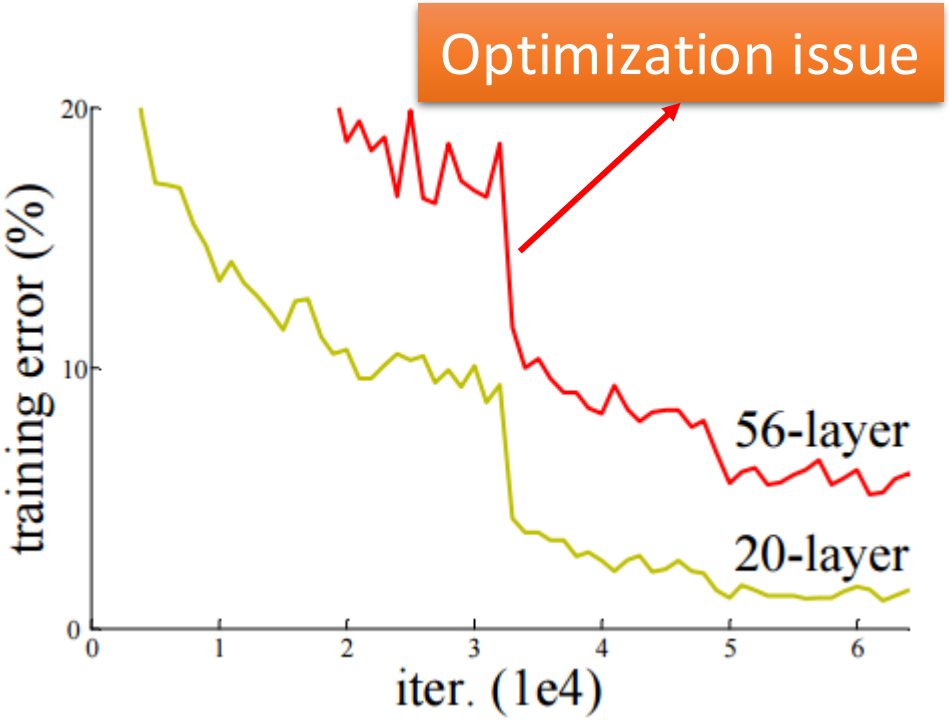
Using image priors, reduce unnecessary parameters, **prevent overfitting**

Every architecture has its advantages

- Residual Connection's advantage?



Testing Data

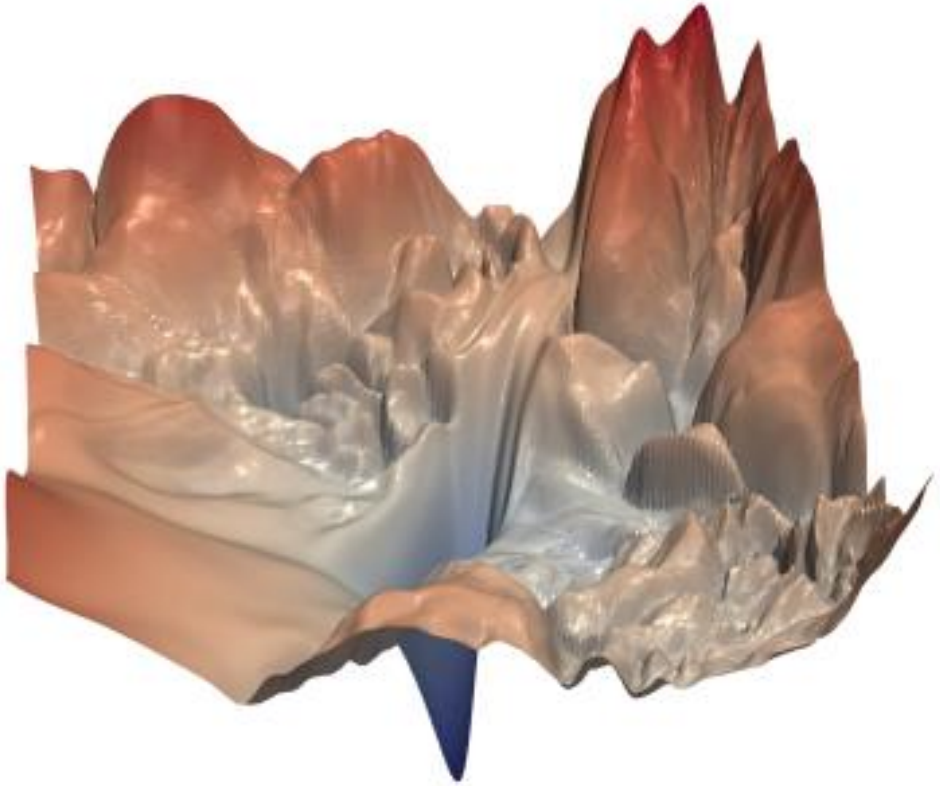


Training Data

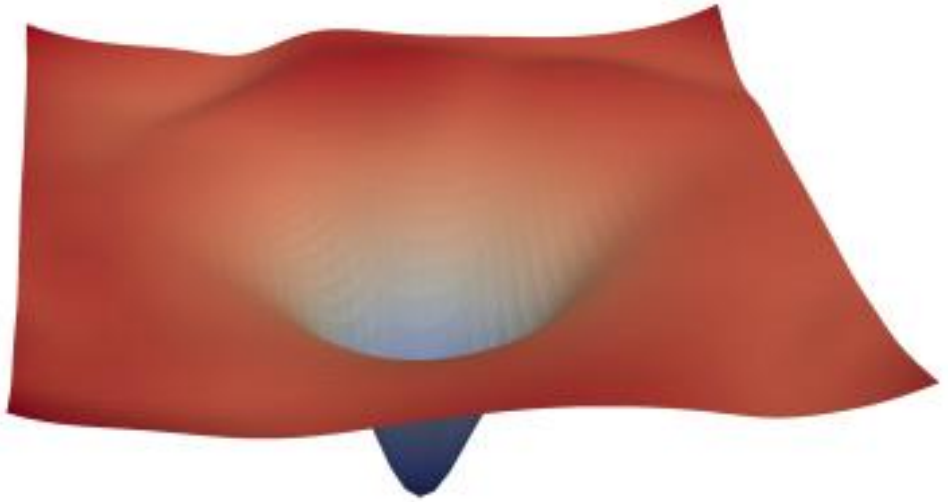
Source of image: http://arxiv.org/abs/1512.03385

Every architecture has its advantages

- Residual Connection? Optimize easier and better



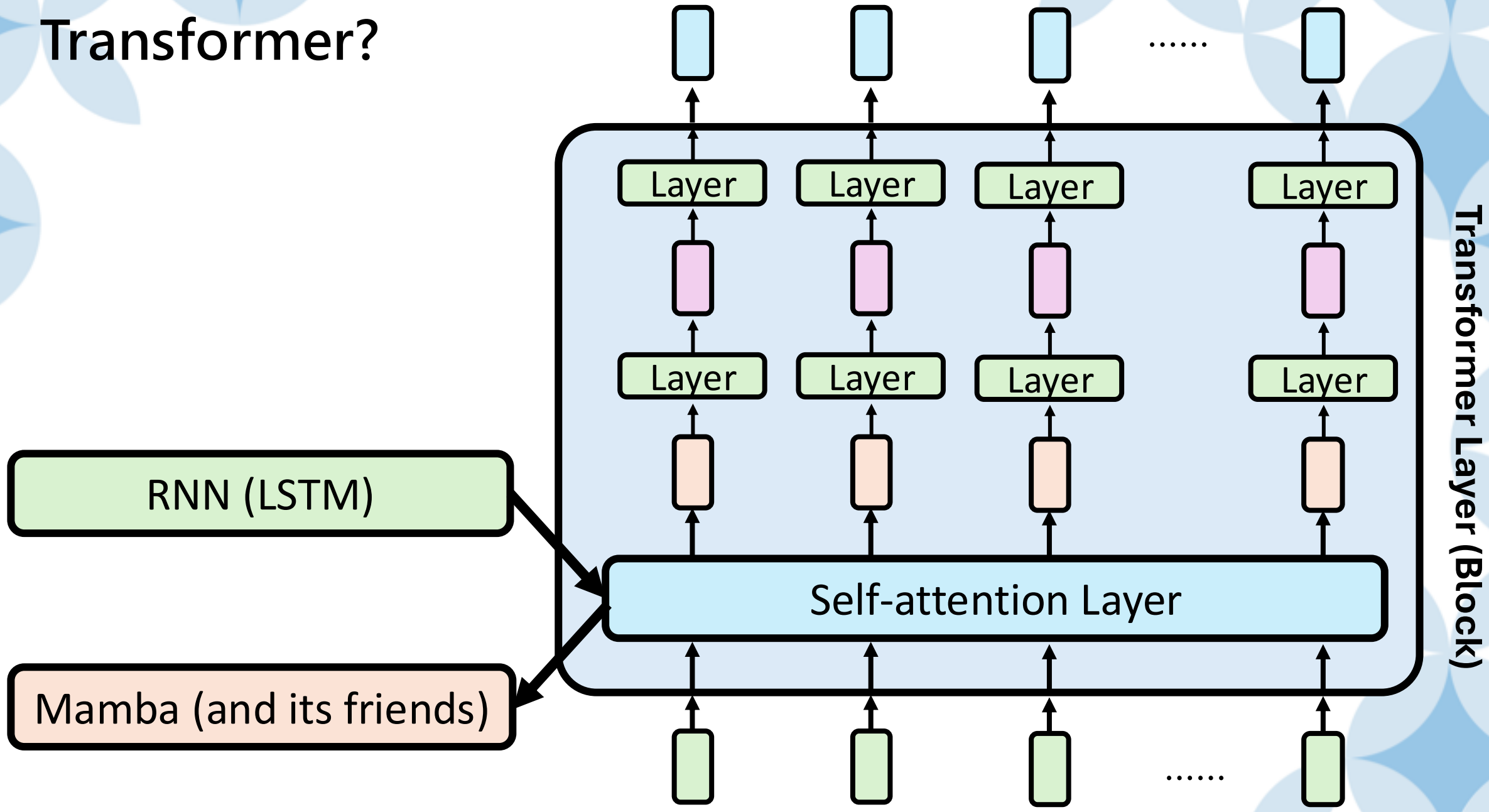
(a) without skip connections



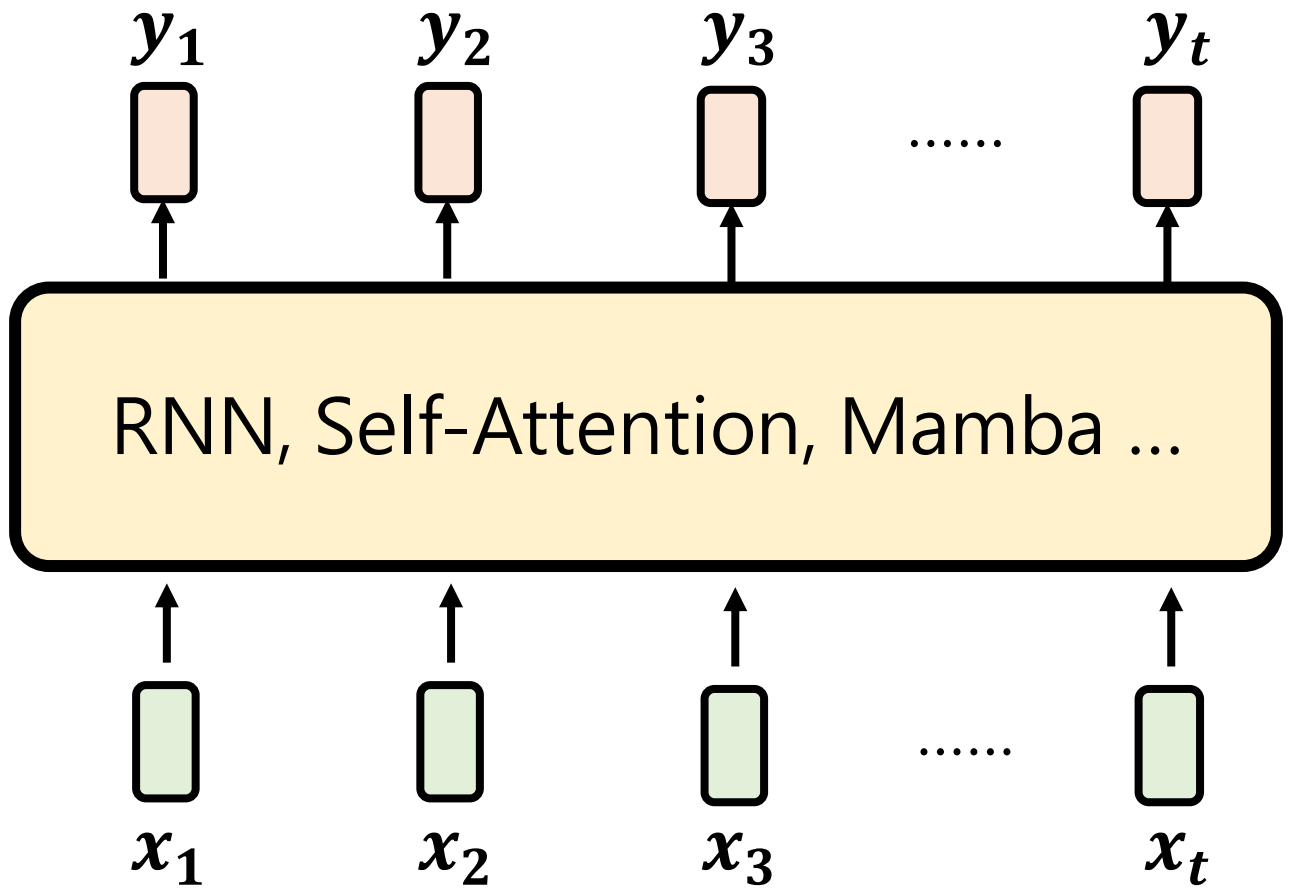
(b) with skip connections

Source of image: https://arxiv.org/abs/1712.09913

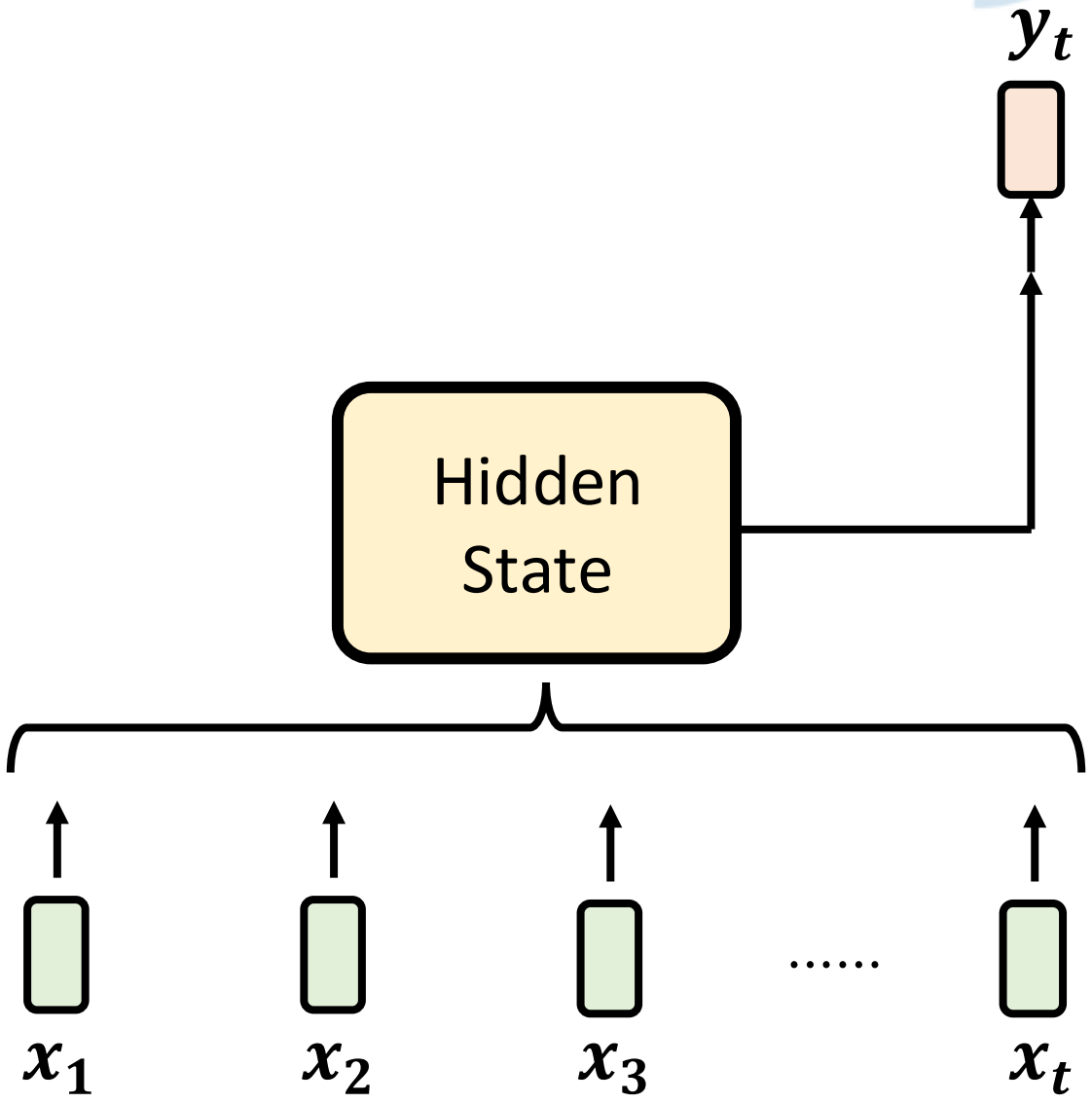
Transformer?



The key question to address



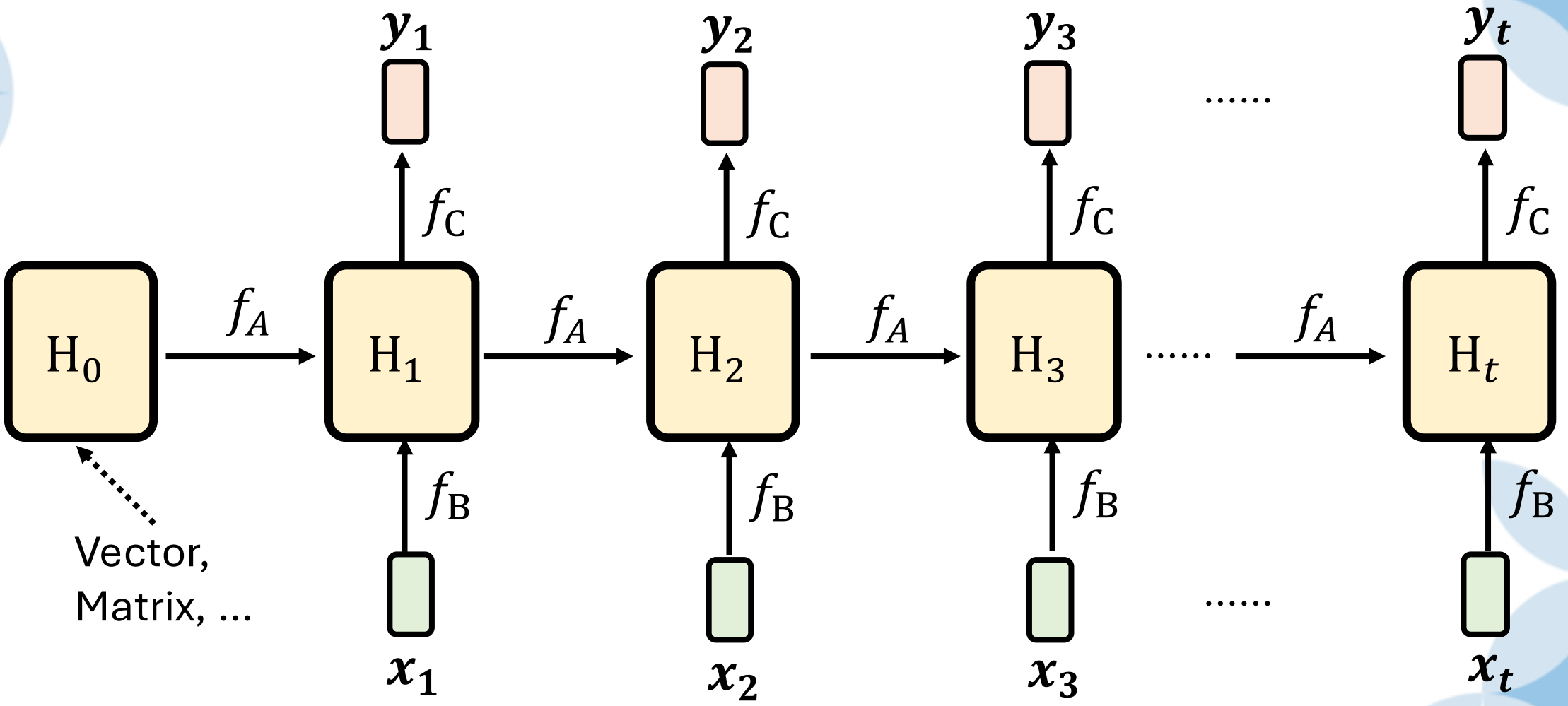
RNN-Style



RNN-Style

$$\mathrm{H}_t = f_A(\mathrm{H}_{t-1}) + f_B(\boldsymbol{x_t})$$

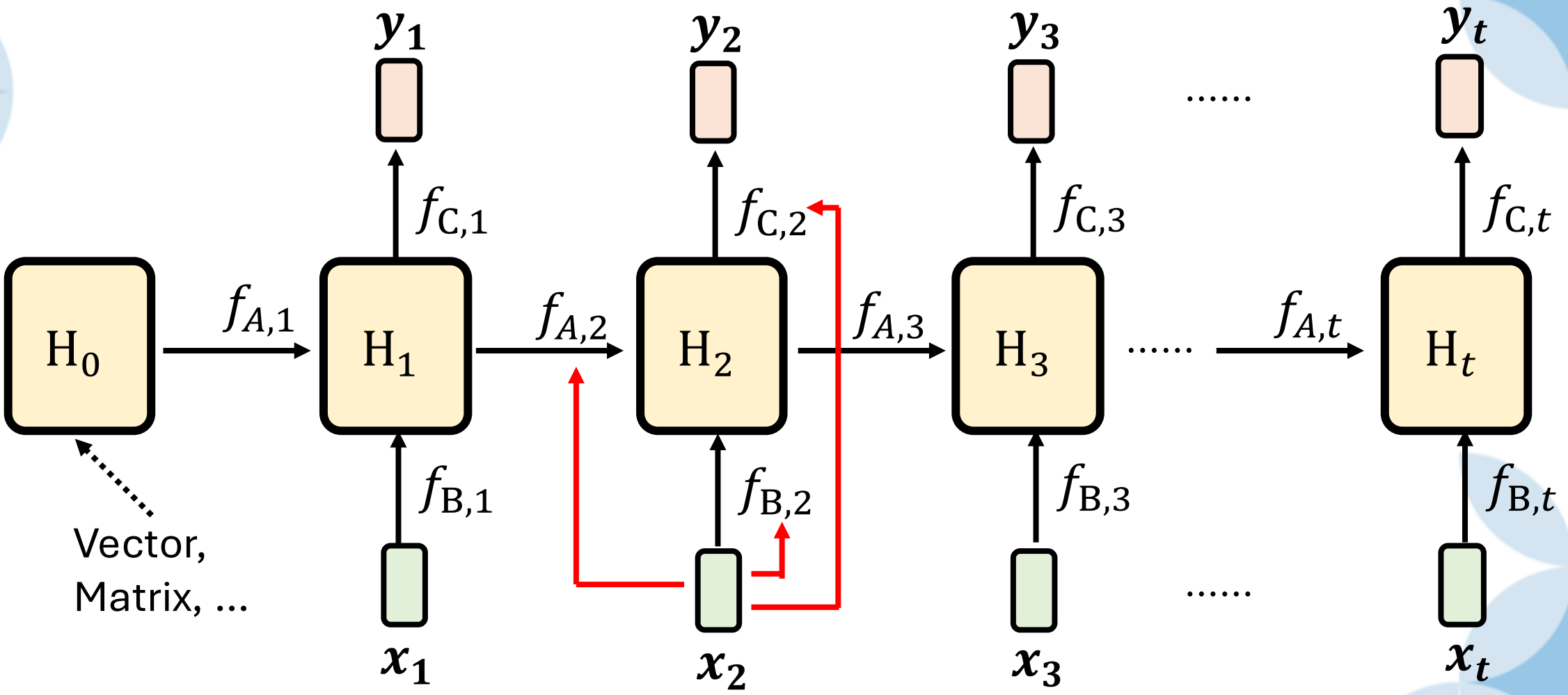
$$\boldsymbol{y_t} = f_C(\mathrm{H}_t)$$



RNN-Style

$$\mathrm{H}_t = f_{A,t}(\mathrm{H}_{t-1}) + f_{B,t}(\boldsymbol{x_t})$$

$$\boldsymbol{y_t} = f_{C,t}(\mathrm{H}_t)$$



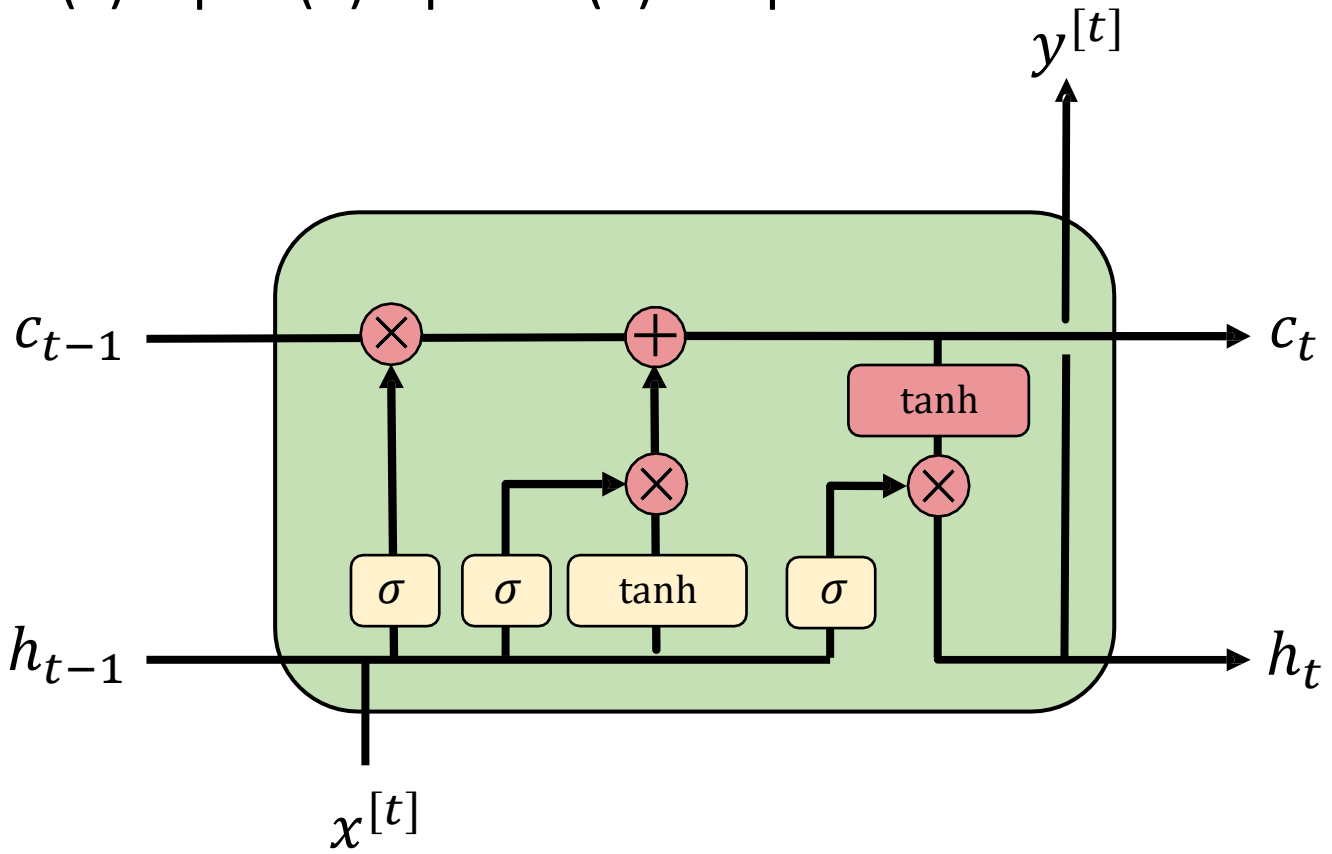
RNN-Style

$$\mathrm{H}_t = f_{A,t}(\mathrm{H}_{t-1}) + f_{B,t}(\boldsymbol{x_t})$$

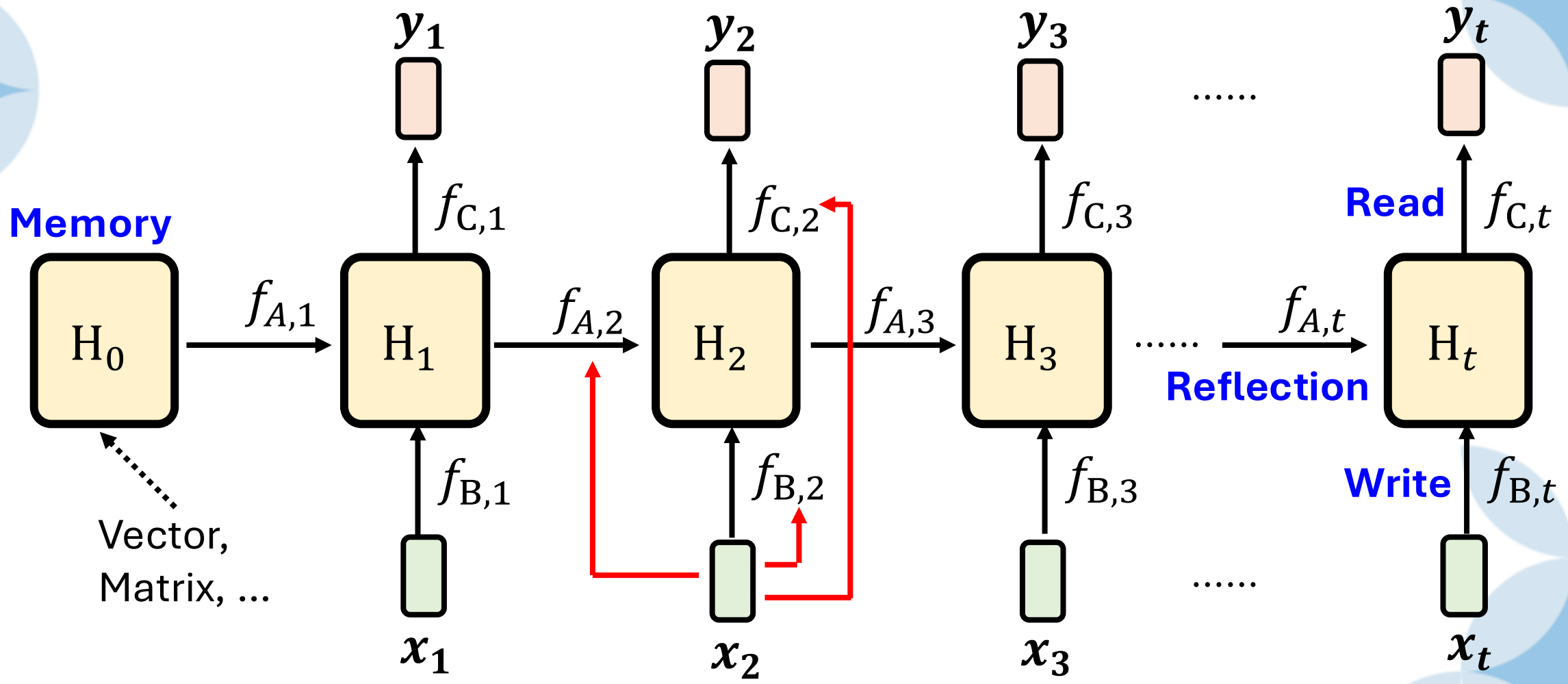
$$\boldsymbol{y_t} = f_{C,t}(\mathrm{H}_t)$$

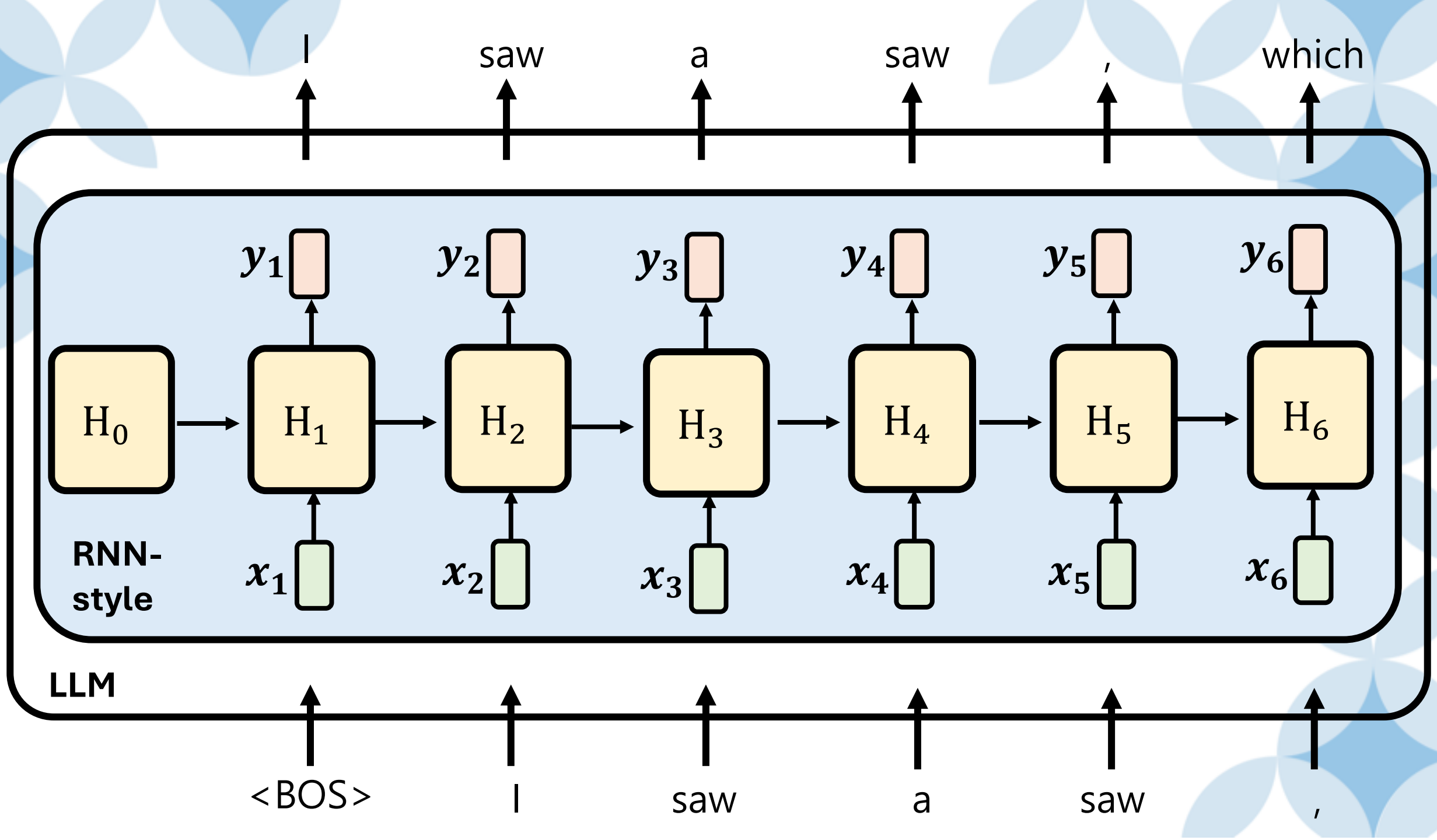
LSTM

- How do LSTM work: 4 Stages
  - (1) Forget (2) Input (3) Update (4) Output

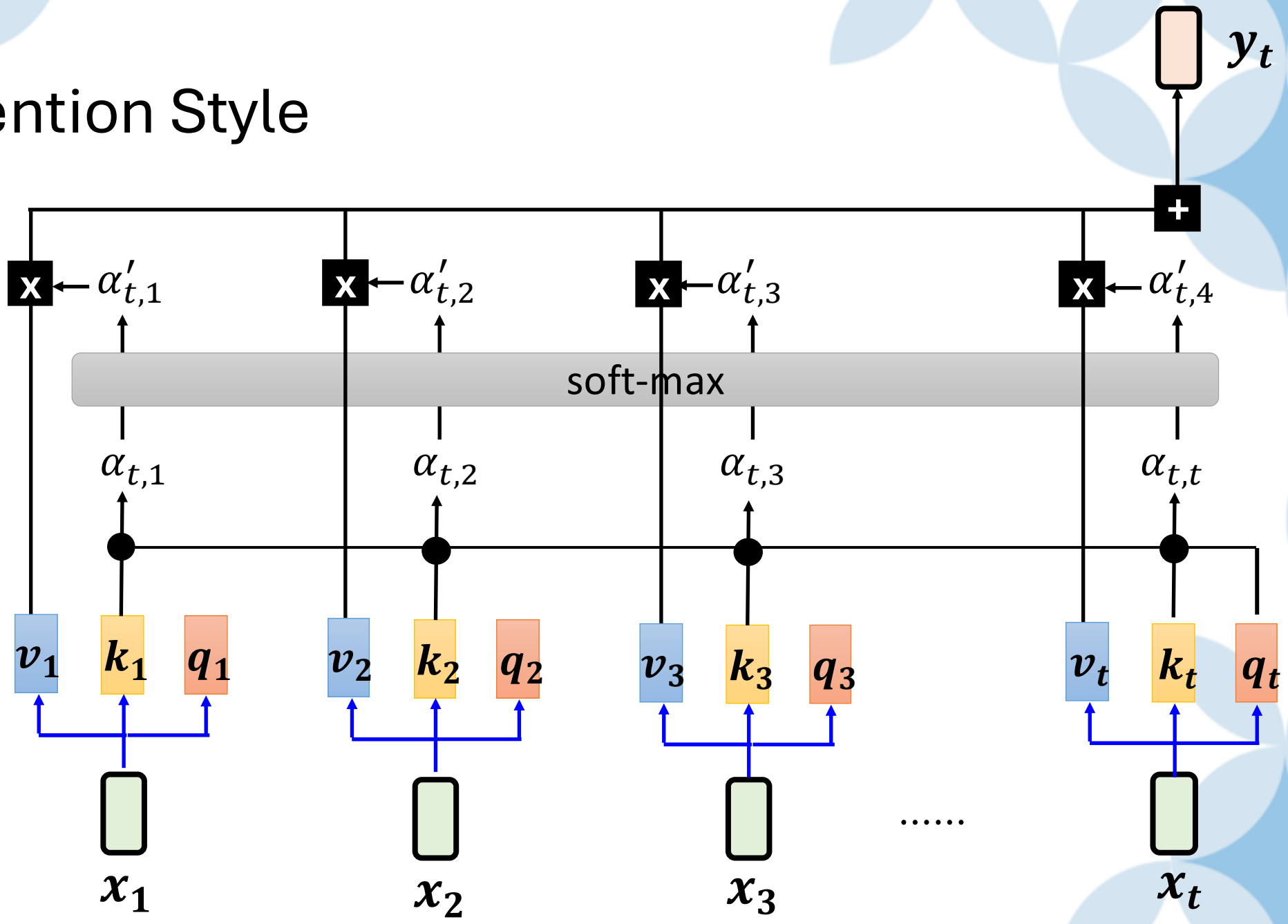


RNN-Style: An explanation

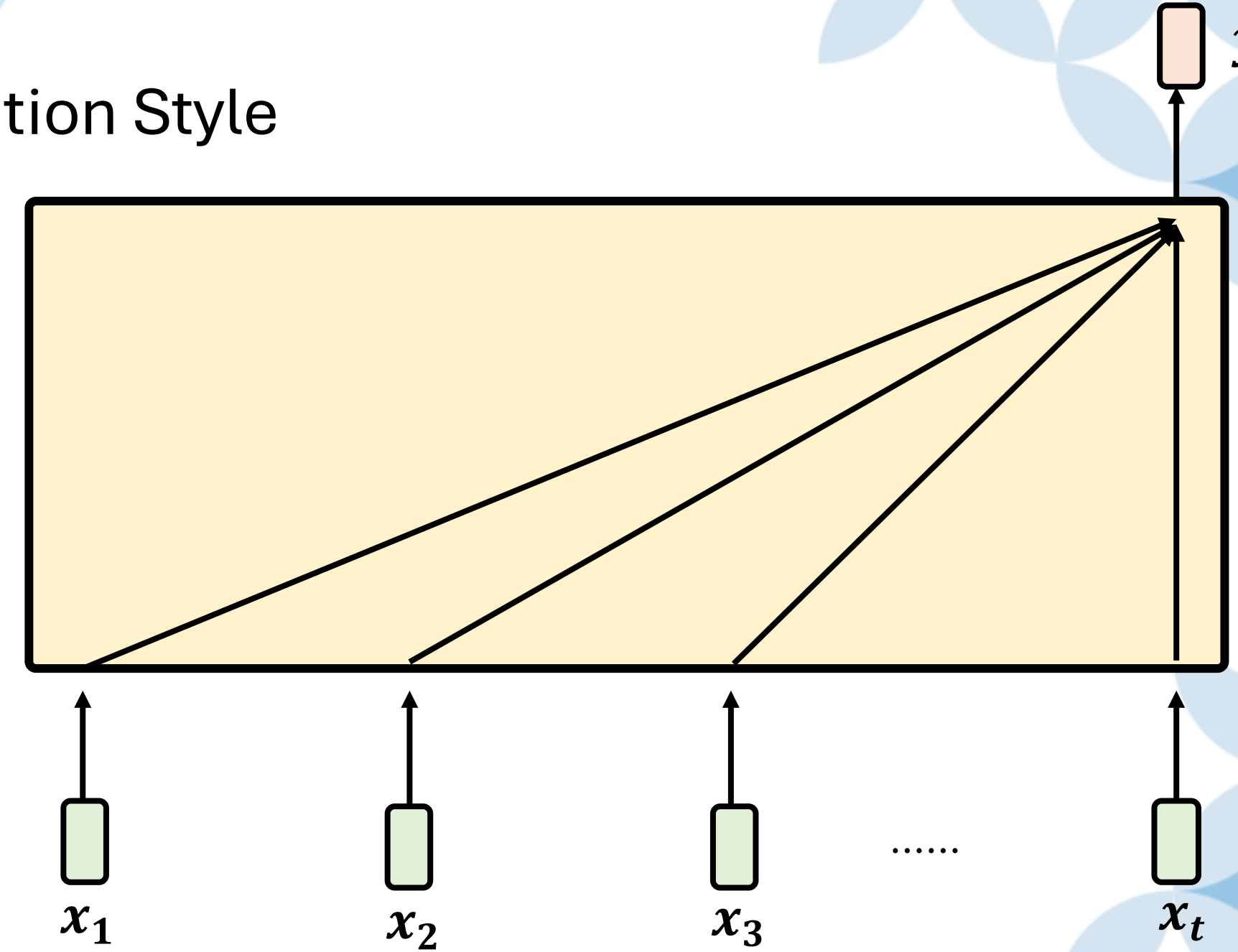



I
saw
a
saw
,
which
y_1
y_2
y_3
y_4
y_5
y_6
H_0
H_1
H_2
H_3
H_4
H_5
H_6
RNN-style
x_1
x_2
x_3
x_4
x_5
x_6
LLM
<BOS>
I
saw
a
saw
,

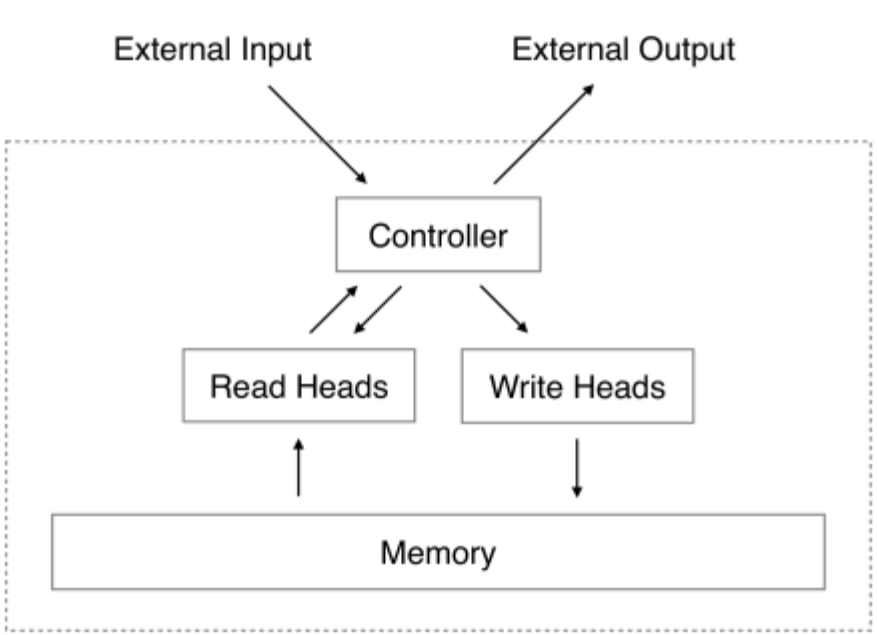
Self-Attention Style



Self-Attention Style



The concept of attention has been around for a long time



Neural Turing Machine

https://arxiv.org/abs/1410.5401

Joe went to the garden then Fred picked up the milk; Joe moved to the bathroom and Fred dropped the milk, and then Dan moved to the living_room.
Where is Dan? A: living room I believe
Where is Joe? A: the bathroom
Fred moved to the bedroom and Joe went to the kitchen then Joe took the milk there and Dan journeyed to the bedroom; Joe discarded the milk.
Where is the milk now ? A: the milk is in the kitchen
Where is Dan now? A: I think he is in the bedroom
Joe took the milk there, after that Mike travelled to the office, then Joe went to the living_room, next Dan went back to the kitchen and Joe travelled to the office.
Where is Joe now? A: I think Joe is in the office

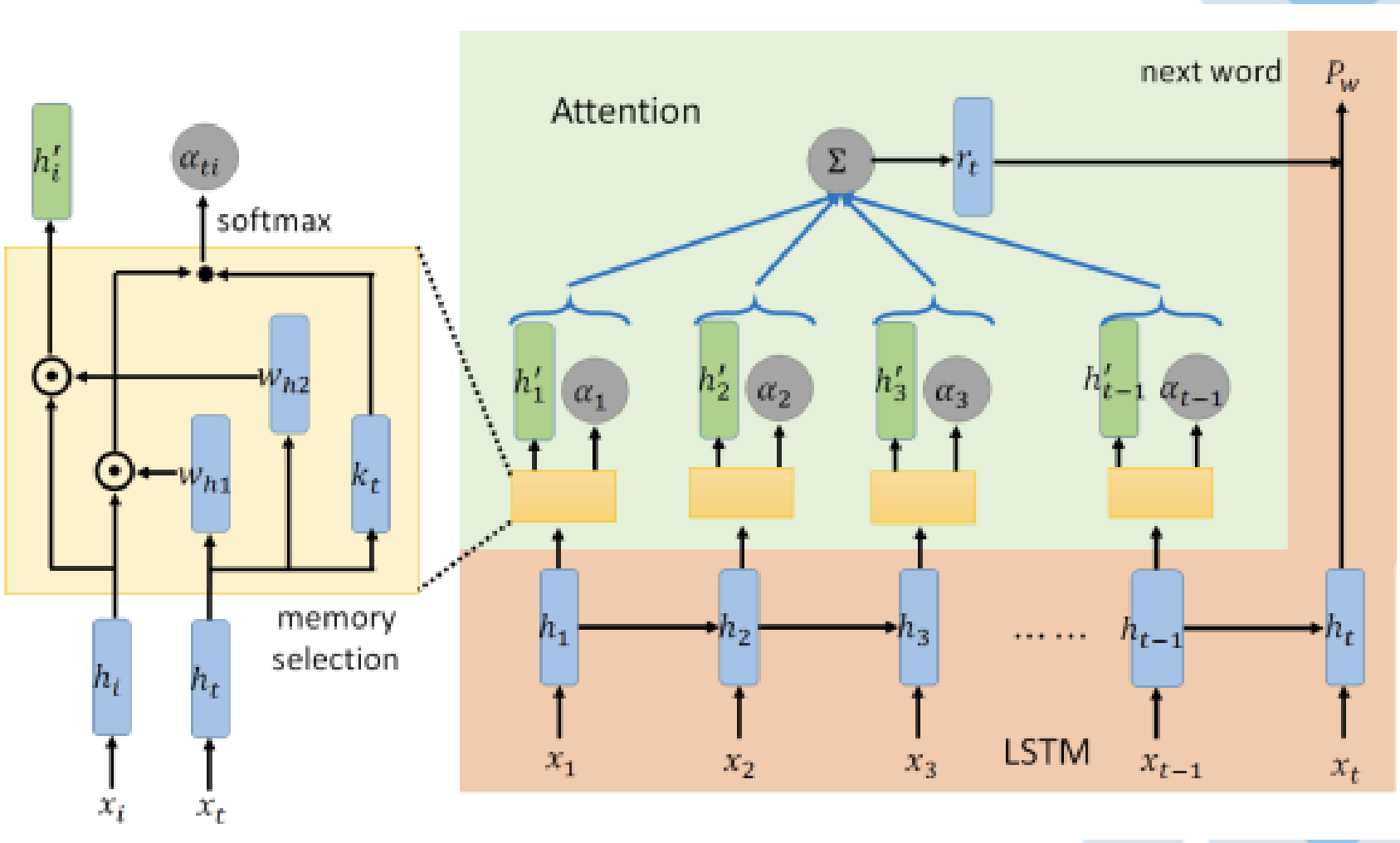
Memory Networks

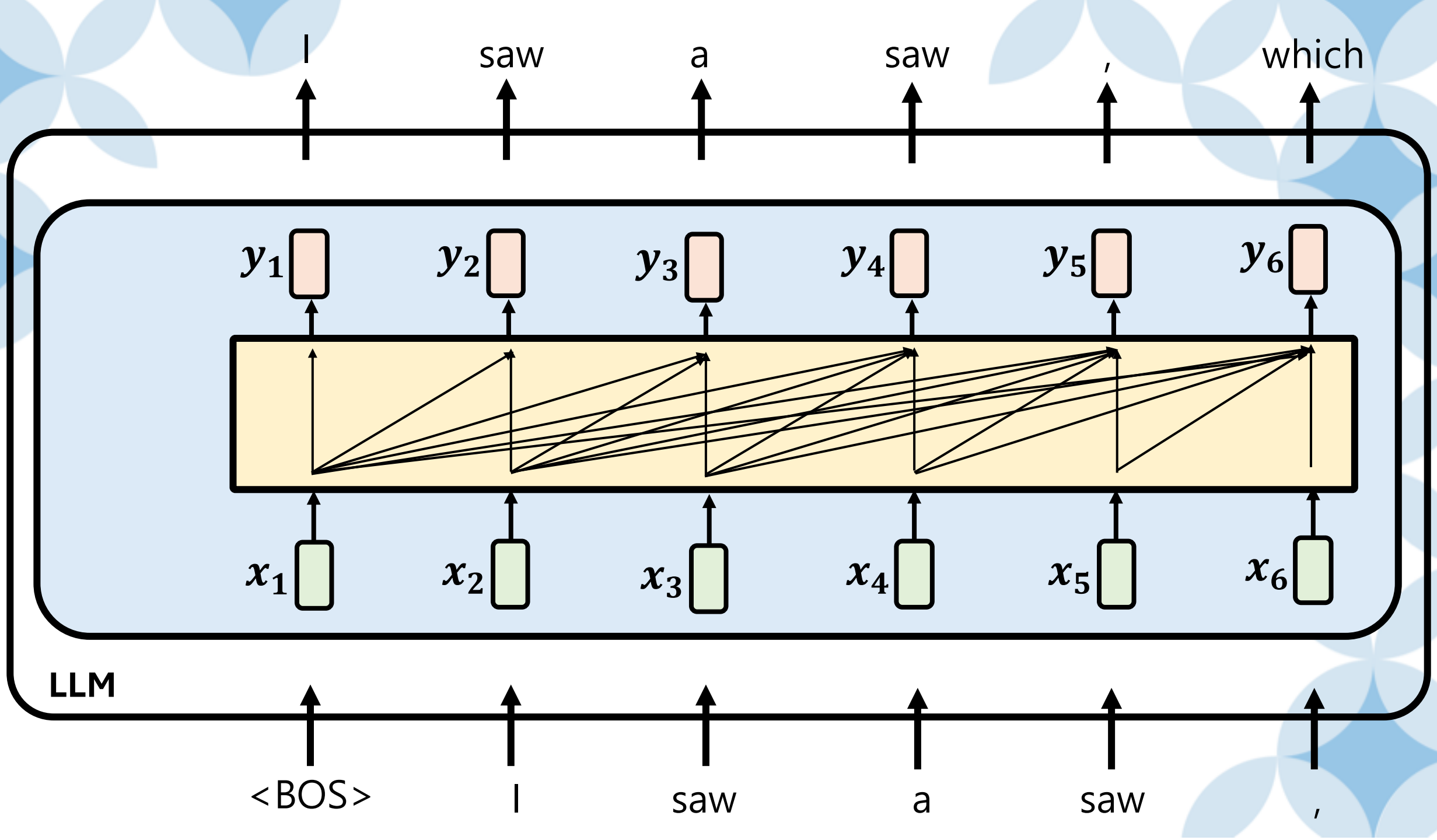
https://arxiv.org/pdf/1410.3916

The concept of attention has been around for a long time

Attention-based Memory Selection Recurrent Network for Language Modeling

https://arxiv.org/abs/1611.08656

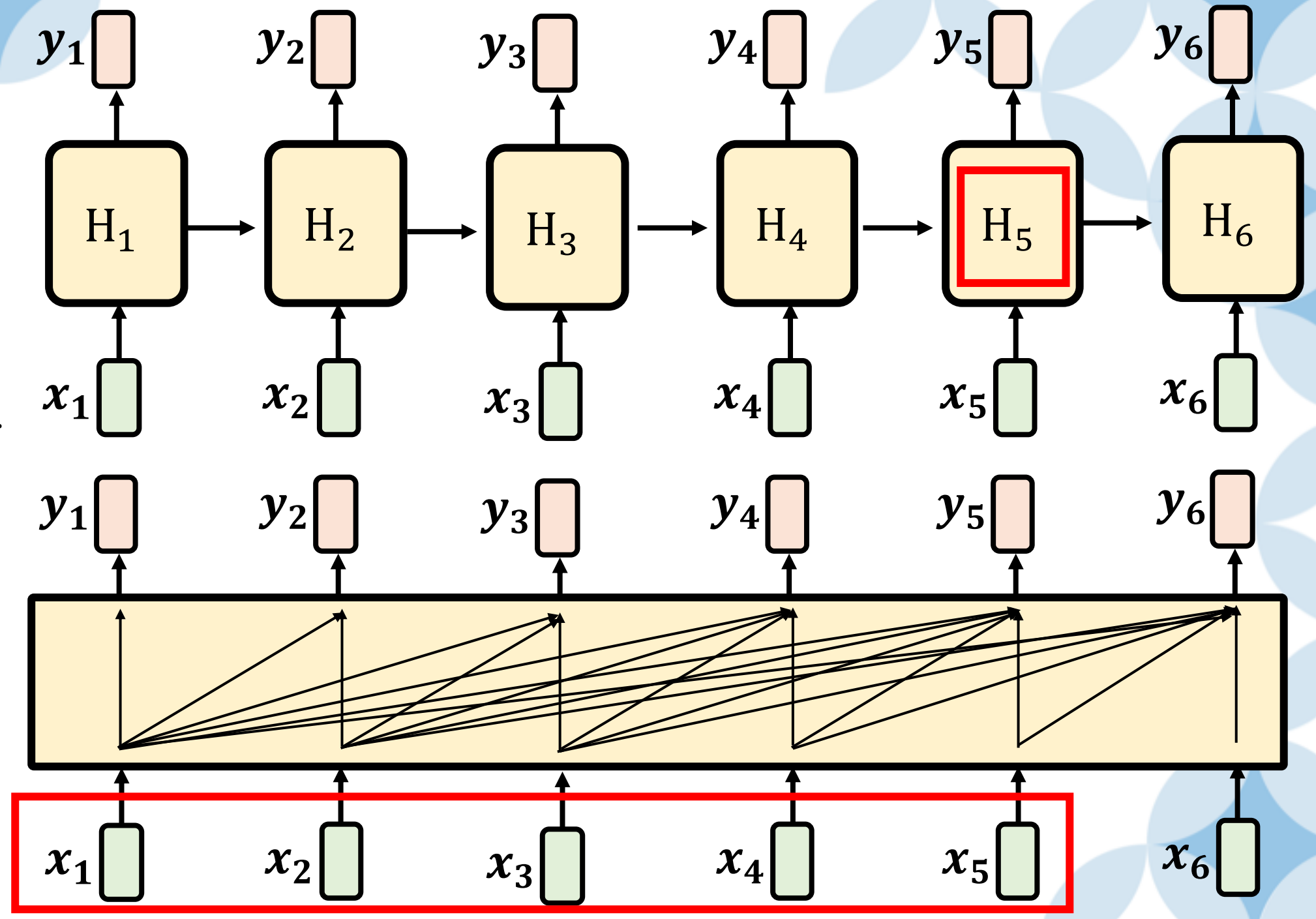



I
saw
a
saw
,
which
y_1
y_2
y_3
y_4
y_5
y_6
x_1
x_2
x_3
x_4
x_5
x_6
LLM
<BOS>
I
saw
a
saw
,

Same compute in each step

RNN can not rem. long history?

Longer input, more compute.



Attention Is All You Need

Not invent Attention, but remove modules other than Attention

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

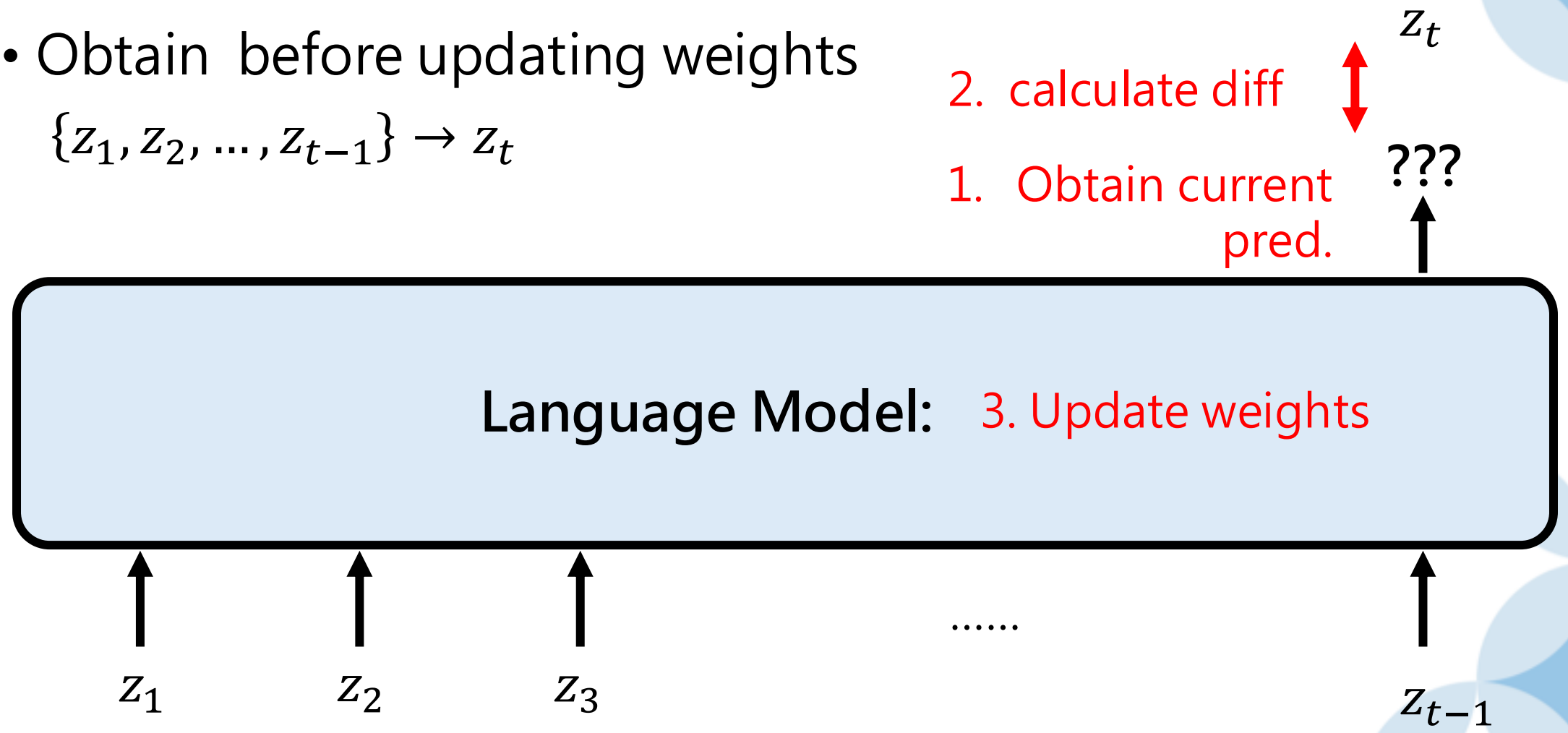
In this work we propose the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as twelve hours on eight P100 GPUs.

https://arxiv.org/abs/1706.03762

Training language models (finding parameters/weights)

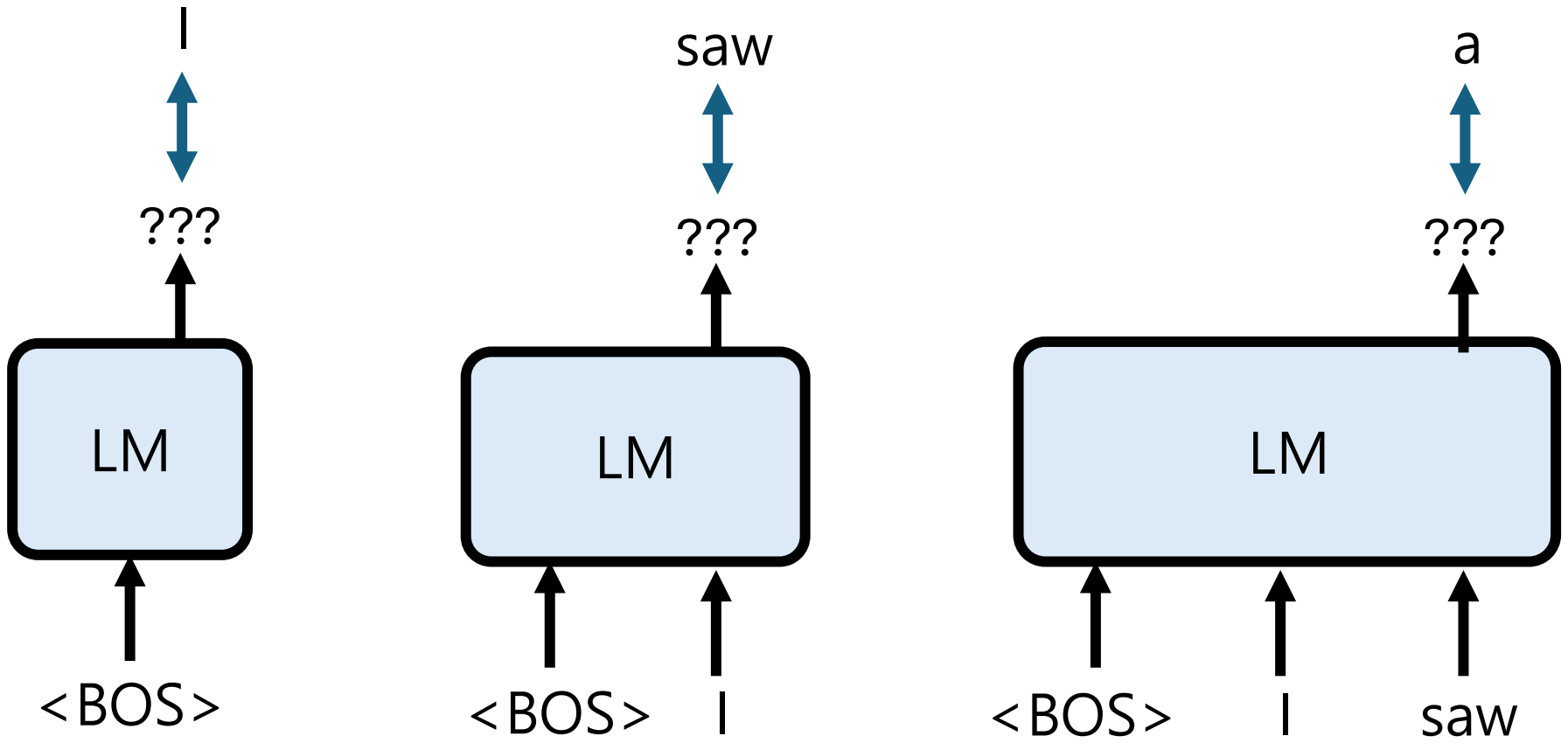
- Obtain before updating weights

$\{z_1, z_2, \dots, z_{t-1}\} \rightarrow z_t$



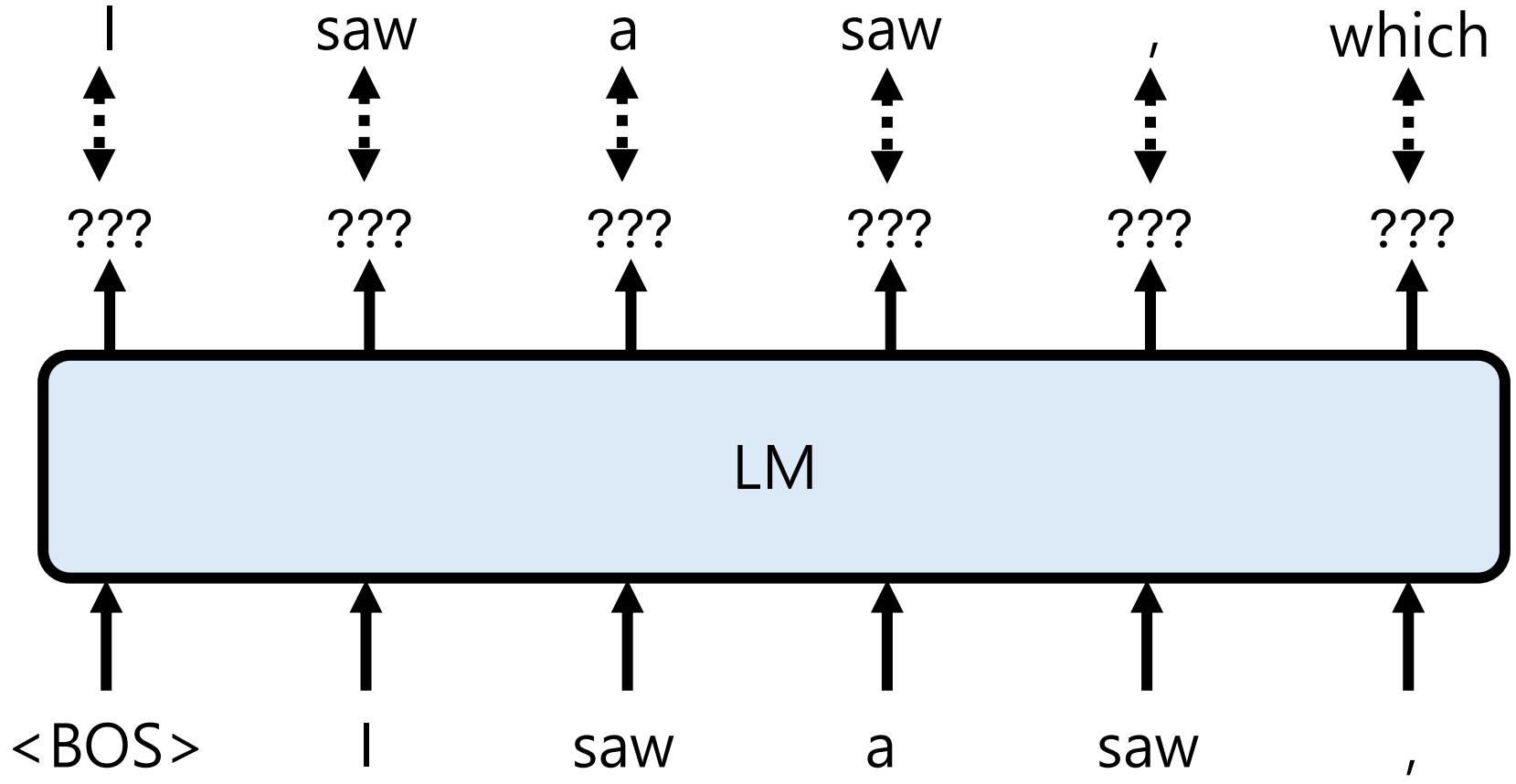
Training language models

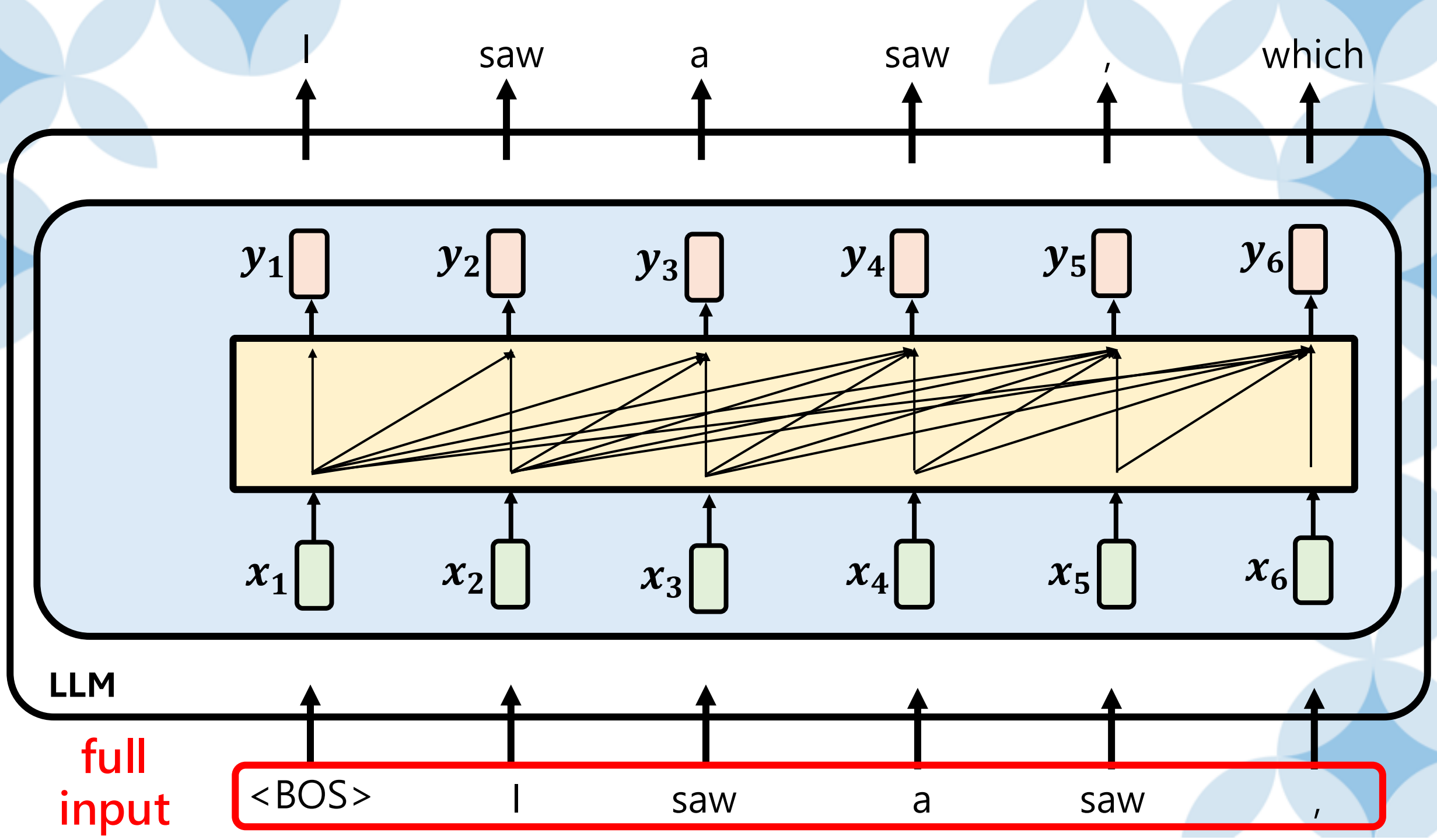
Assume we want model to say「I saw a saw, which……」

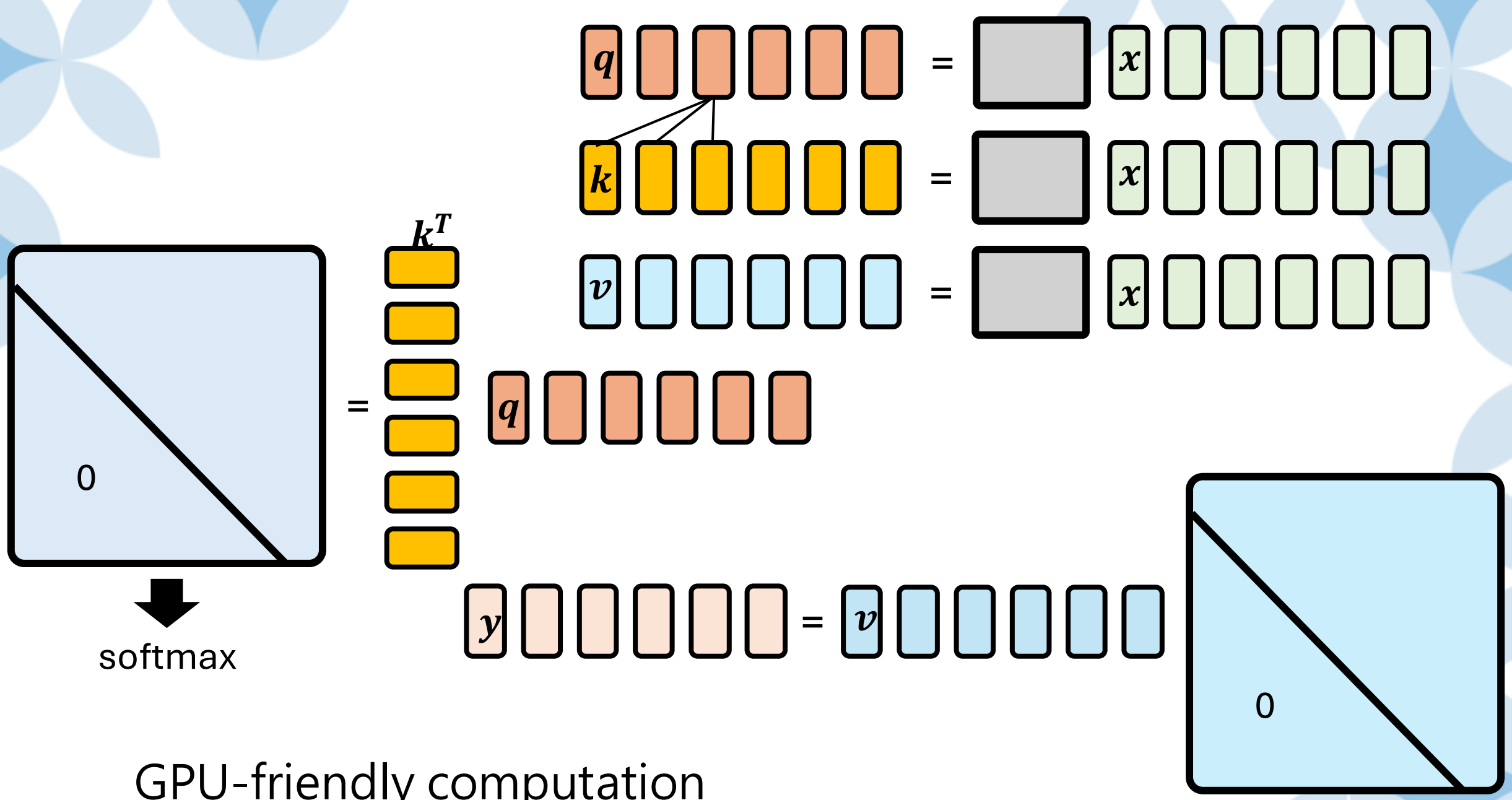


Training language models

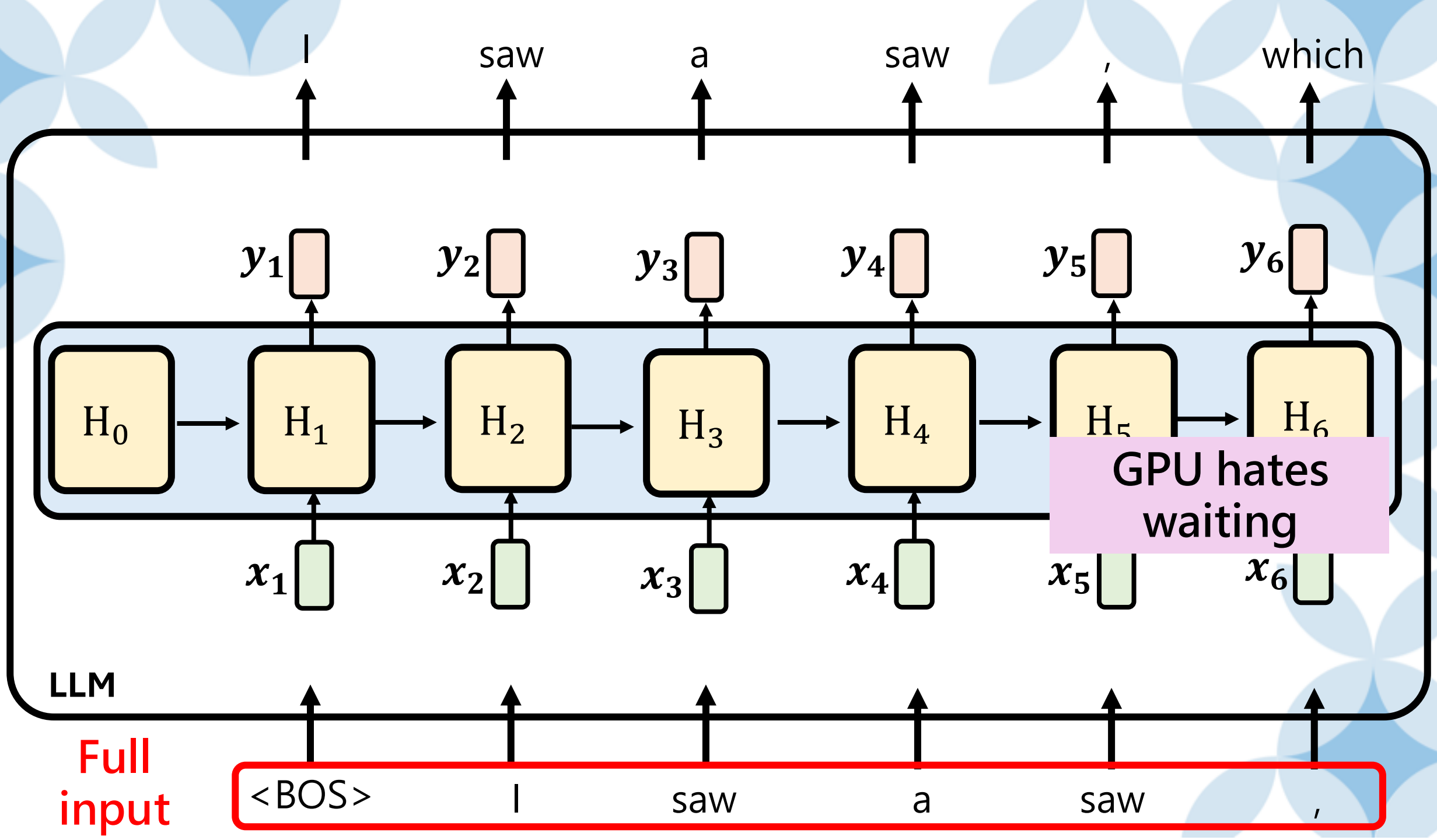
Assume we want model to say「I saw a saw, which……」




I
saw
a
saw
,
which
y_1
y_2
y_3
y_4
y_5
y_6
x_1
x_2
x_3
x_4
x_5
x_6
LLM
full input
<BOS>
I
saw
a
saw
,



GPU-friendly computation


I
saw
a
saw
,
which
y_1
y_2
y_3
y_4
y_5
y_6
H_0
H_1
H_2
H_3
H_4
H_5
H_6
GPU hates waiting
x_1
x_2
x_3
x_4
x_5
x_6
LLM
Full input
<BOS>
I
saw
a
saw
,

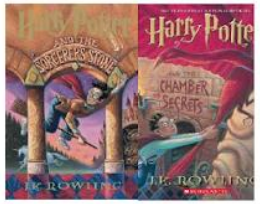
Self-attention vs. RNN-style

	Self-attention	RNN
Inference	compute, memory increases as the input increases	Fixed compute and memory requirements
Training	Easy to parallelize	Hard to parallelize (?)

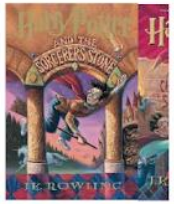
Google's Gemini 1.5 can (almost) fit the entire Harry Potter + Lord of the Ring series in its 2 million context window

Gemini 1.5 2M
(June 2024)

Claude 2.1
(July 2023)



GPT-4 Turbo
(March 2023)



GPT-3.5 Turbo
(March 2022)

RAG and AI agents both require language models to handle very long sequences.

Videos and audios are longer sequences than texts

Source of image: https://www.artfish.ai/p/long-context-llms

Is it possible to parallelize RNN training?

$$\mathrm{H}_t = f_{A,t}(\mathrm{H}_{t-1}) + f_{B,t}(\boldsymbol{x_t})$$

$$\boldsymbol{y_t} = f_{C,t}(\mathrm{H}_t)$$

$$f_{A,1}(\mathrm{H}_0) = O$$

$$\begin{cases} \mathrm{H}_1 = f_{A,1}(\mathrm{H}_0) + f_{B,1}(\boldsymbol{x_1}) = f_{B,1}(\boldsymbol{x_1}) \\ \mathrm{H}_2 = f_{A,2}(\mathrm{H}_1) + f_{B,2}(\boldsymbol{x_2}) = f_{A,2}\left(f_{B,1}(\boldsymbol{x_1})\right) + f_{B,2}(\boldsymbol{x_2}) \\ \mathrm{H}_3 = f_{A,3}(\mathrm{H}_2) + f_{B,3}(\boldsymbol{x_3}) = f_{A,3}\left(f_{A,2}\left(f_{B,1}(\boldsymbol{x_1})\right) + f_{B,2}(\boldsymbol{x_2})\right) + f_{B,3}(\boldsymbol{x_3}) \\ \vdots \\ \mathrm{H}_t = f_{A,t}(\mathrm{H}_{t-1}) + f_{B,t}(\boldsymbol{x_t}) = \underbrace{f_{A,t}(f_{A,t-1} \cdots f_{A,3}(f_{A,2}}(f_{B,1}(\boldsymbol{x_1}) \dots) \quad \dots + f_{B,t}(\boldsymbol{x_t}) \end{cases}$$

Is it possible to parallelize RNN training?

$$\mathrm{H}_t = \mathrm{H}_{t-1} + f_{B,t}(\boldsymbol{x_t})$$

$$\boldsymbol{y_t} = f_{C,t}(\mathrm{H}_t)$$

$f_{A,1}(\mathrm{H}_0) = O$

$$\begin{cases} \mathrm{H}_1 = \mathrm{H}_0 + f_{B,1}(\boldsymbol{x_1}) & = f_{B,1}(\boldsymbol{x_1}) \\ \mathrm{H}_2 = \mathrm{H}_1 + f_{B,2}(\boldsymbol{x_2}) & = f_{B,1}(\boldsymbol{x_1}) + f_{B,2}(\boldsymbol{x_2}) \\ \mathrm{H}_3 = \mathrm{H}_2 + f_{B,3}(\boldsymbol{x_3}) & = f_{B,1}(\boldsymbol{x_1}) + f_{B,2}(\boldsymbol{x_2}) + f_{B,3}(\boldsymbol{x_3}) \\ \vdots \\ \mathrm{H}_t = \mathrm{H}_{t-1} + f_{B,t}(\boldsymbol{x_t}) & = f_{B,1}(\boldsymbol{x_1}) + f_{B,2}(\boldsymbol{x_2}) + f_{B,3}(\boldsymbol{x_3}) \quad \dots\dots + f_{B,t}(\boldsymbol{x_t}) \end{cases}$$

H_t is a $d \times d$ matric

$$f_{B,t}(\boldsymbol{x_t}) = D_t$$

Is it possible to parallelize RNN training?

$$\mathrm{H}_t = \mathrm{H}_{t-1} + f_{B,t}(\boldsymbol{x_t})$$

$$\boldsymbol{y_t} = f_{C,t}(\mathrm{H}_t)$$

$f_{A,1}(\mathrm{H}_0) = O$

$$\begin{cases} \mathrm{H}_1 = D_1 & \boldsymbol{y_1} = D_1\boldsymbol{q_1} \\ \mathrm{H}_2 = D_1 + D_2 & \boldsymbol{y_2} = D_1\boldsymbol{q_2} + D_2\boldsymbol{q_2} \\ \mathrm{H}_3 = D_1 + D_2 + D_3 & \boldsymbol{y_3} = D_1\boldsymbol{q_3} + D_2\boldsymbol{q_3} + D_3\boldsymbol{q_3} \\ \vdots & \\ \mathrm{H}_t = D_1 + D_2 + \cdots + D_t & \boldsymbol{y_t} = D_1\boldsymbol{q_t} + D_2\boldsymbol{q_t} + \cdots + D_t\boldsymbol{q_t} \end{cases}$$

H_t is a $d \times d$ matric

$$f_{B,t}(\boldsymbol{x_t}) = D_t$$

$$f_{C,t}(\mathrm{H}_t) = \mathrm{H}_t\boldsymbol{q_t}$$

$$\boldsymbol{q_t} = W_Q\boldsymbol{x_t}$$

Is it possible to parallelize RNN training?

$$f_{A,1}(\mathrm{H}_0) = O$$

$$\begin{cases} \boldsymbol{y_1} = D_1\boldsymbol{q_1} \\ \boldsymbol{y_2} = D_1\boldsymbol{q_2} + D_2\boldsymbol{q_2} \\ \boldsymbol{y_3} = D_1\boldsymbol{q_3} + D_2\boldsymbol{q_3} + D_3\boldsymbol{q_3} \\ \vdots \\ \boldsymbol{y_t} = D_1\boldsymbol{q_t} + D_2\boldsymbol{q_t} + \cdots + D_t\boldsymbol{q_t} \end{cases}$$

$$\mathrm{H}_t = \mathrm{H}_{t-1} + f_{B,t}(\boldsymbol{x_t})$$

$$\boldsymbol{y_t} = f_{C,t}(\mathrm{H}_t)$$

H_t is a $d \times d$ matric

$$f_{B,t}(\boldsymbol{x_t}) = D_t$$

$$D_t = \boldsymbol{v_t}\boldsymbol{k_t}^T$$

$$\boldsymbol{v_t} = W_v\boldsymbol{x_t}$$

$$\boldsymbol{k_t} = W_k\boldsymbol{x_t}$$

$$f_{C,t}(\mathrm{H}_t) = \mathrm{H}_t\boldsymbol{q_t}$$

$$\boldsymbol{q_t} = W_Q\boldsymbol{x_t}$$

Is it possible to parallelize RNN training?

$$f_{A,1}(\mathrm{H}_0) = O$$

$$\begin{cases} \boldsymbol{y_1} = \boldsymbol{v_1}\boldsymbol{k_1}^T\boldsymbol{q_1} \\ \boldsymbol{y_2} = \boldsymbol{v_1}\boldsymbol{k_1}^T\boldsymbol{q_2} + \boldsymbol{v_2}\boldsymbol{k_2}^T\boldsymbol{q_2} \\ \boldsymbol{y_3} = \boldsymbol{v_1}\boldsymbol{k_1}^T\boldsymbol{q_3} + \boldsymbol{v_2}\boldsymbol{k_2}^T\boldsymbol{q_3} + \boldsymbol{v_3}\boldsymbol{k_3}^T\boldsymbol{q_3} \\ \vdots \\ \boldsymbol{y_t} = \boldsymbol{v_1}\boldsymbol{k_1}^T\boldsymbol{q_t} + \boldsymbol{v_2}\boldsymbol{k_2}^T\boldsymbol{q_t} + \cdots + \boldsymbol{v_t}\boldsymbol{k_t}^T\boldsymbol{q_t} \end{cases}$$

$$\mathrm{H}_t = \mathrm{H}_{t-1} + f_{B,t}(\boldsymbol{x_t})$$

$$\boldsymbol{y_t} = f_{C,t}(\mathrm{H}_t)$$

H_t is a $d \times d$ matric

$$f_{B,t}(\boldsymbol{x_t}) = D_t$$

$$D_t = \boldsymbol{v_t}\boldsymbol{k_t}^T$$

$$\boldsymbol{v_t} = W_v\boldsymbol{x_t}$$

$$\boldsymbol{k_t} = W_k\boldsymbol{x_t}$$

$$f_{C,t}(\mathrm{H}_t) = \mathrm{H}_t\boldsymbol{q_t}$$

$$\boldsymbol{q_t} = W_Q\boldsymbol{x_t}$$

Is it possible to parallelize RNN training?

$$f_{A,1}(\mathrm{H}_0) = O$$

$$\boldsymbol{y_t} = \boldsymbol{v_1}\boldsymbol{k_1}^T\boldsymbol{q_t} + \boldsymbol{v_2}\boldsymbol{k_2}^T\boldsymbol{q_t} + \cdots + \boldsymbol{v_t}\boldsymbol{k_t}^T\boldsymbol{q_t}$$

$$= \boldsymbol{v_1}a_{t,1} + \boldsymbol{v_2}a_{t,2} + \cdots + \boldsymbol{v_t}a_{t,t}$$

$$= a_{t,1}\boldsymbol{v_1} + a_{t,2}\boldsymbol{v_2} + \cdots + a_{t,t}\boldsymbol{v_t}$$

This is self-attention! (w/o softmax)

↦ Linear Attention

$$\mathrm{H}_t = \mathrm{H}_{t-1} + f_{B,t}(\boldsymbol{x_t})$$

$$\boldsymbol{y_t} = f_{C,t}(\mathrm{H}_t)$$

H_t is a $d \times d$ matric

$$f_{B,t}(\boldsymbol{x_t}) = D_t$$

$$D_t = \boldsymbol{v_t}\boldsymbol{k_t}^T$$

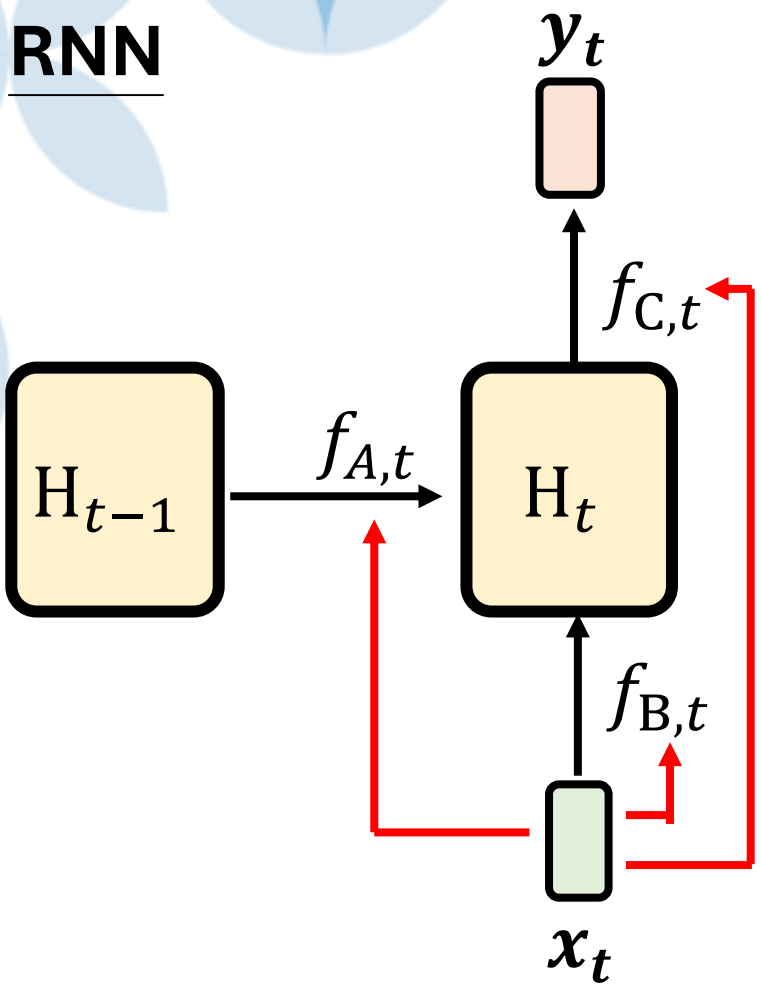
$$\boldsymbol{v_t} = W_v\boldsymbol{x_t}$$

$$\boldsymbol{k_t} = W_k\boldsymbol{x_t}$$

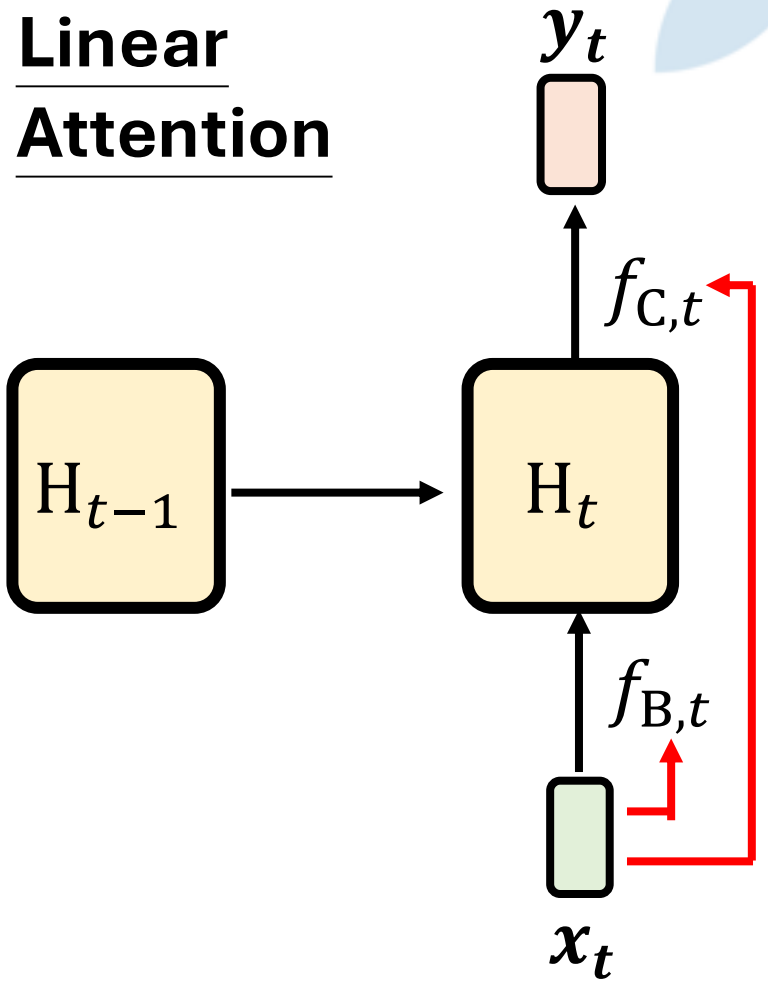
$$f_{C,t}(\mathrm{H}_t) = \mathrm{H}_t\boldsymbol{q_t}$$

$$\boldsymbol{q_t} = W_Q\boldsymbol{x_t}$$

RNN



Linear Attention



$$f_{C,t}(\mathrm{H}_t) = \mathrm{H}_t \boldsymbol{q_t}$$

$$\boldsymbol{q_t} = W_Q \boldsymbol{x_t}$$

$$f_{B,t}(\boldsymbol{x_t}) = \boldsymbol{v_t} \boldsymbol{k_t}^T$$

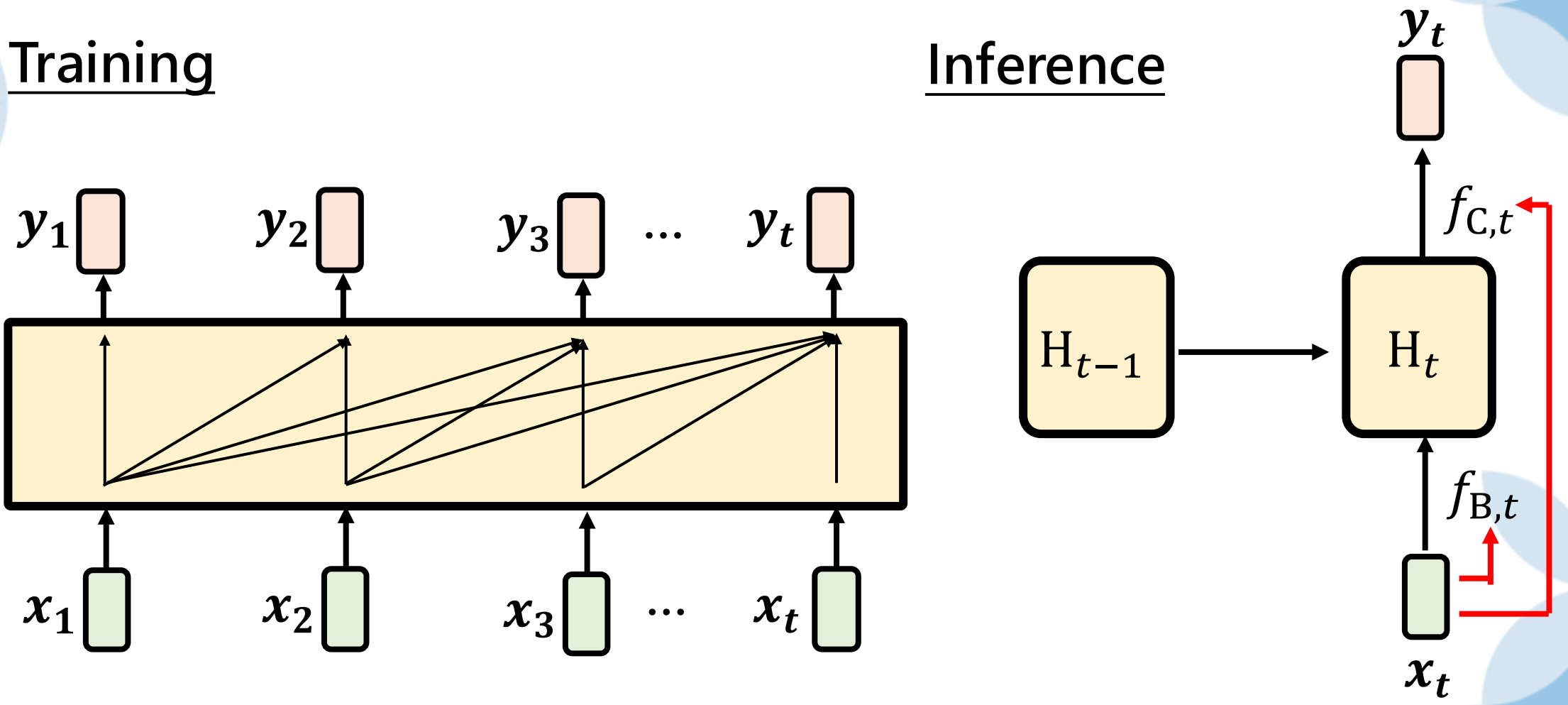
$$\boldsymbol{v_t} = W_v \boldsymbol{x_t}$$

$$\boldsymbol{k_t} = W_k \boldsymbol{x_t}$$

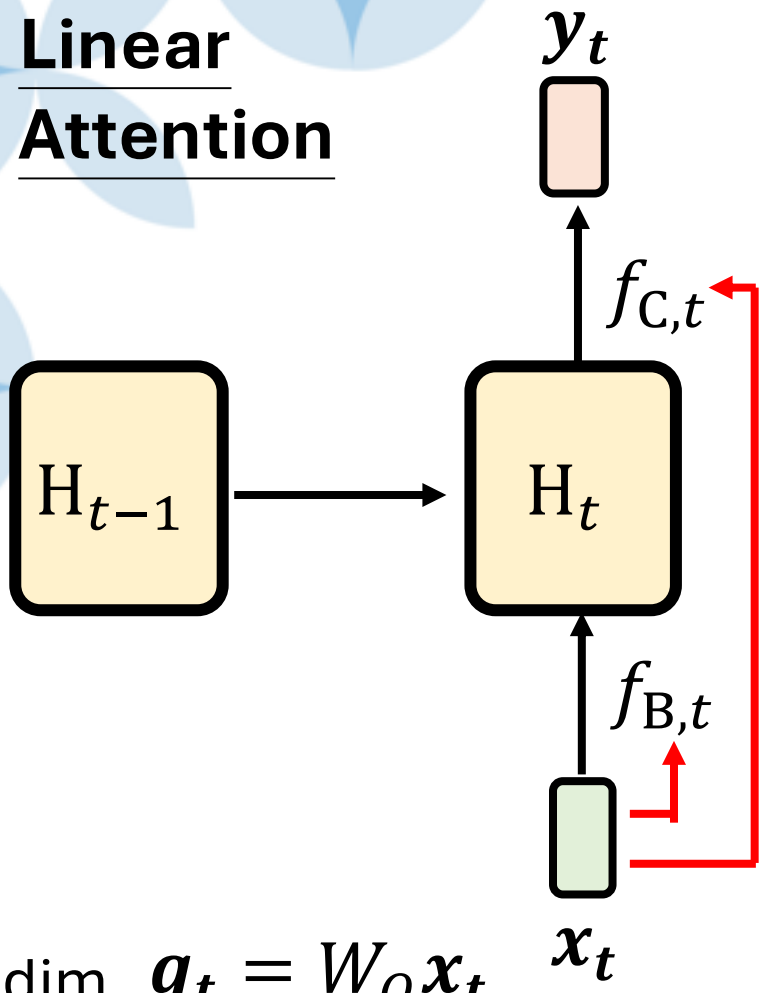
- Linear Attention is generalized RNN w/o "Reflection" $f_{A,t}$
- Linear Attention is Self-attention w/o Softmax

Linear Attention

at training, it works like self-attention
at inference, it works like RNN

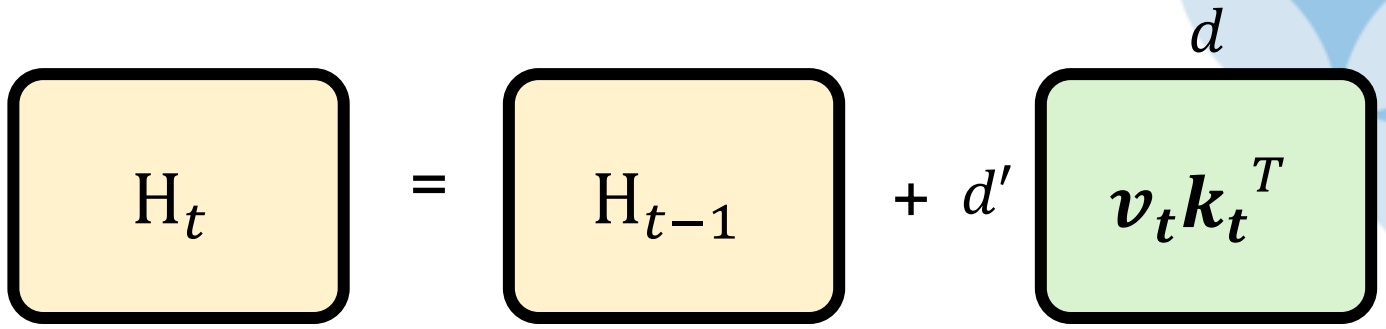


Linear Attention



$$H_t = H_{t-1} + f_{B,t}(x_t) \qquad f_{B,t}(x_t) = v_t k_t^T$$

$$y_t = f_{C,t}(H_t) \qquad f_{C,t}(H_t) = H_t q_t$$

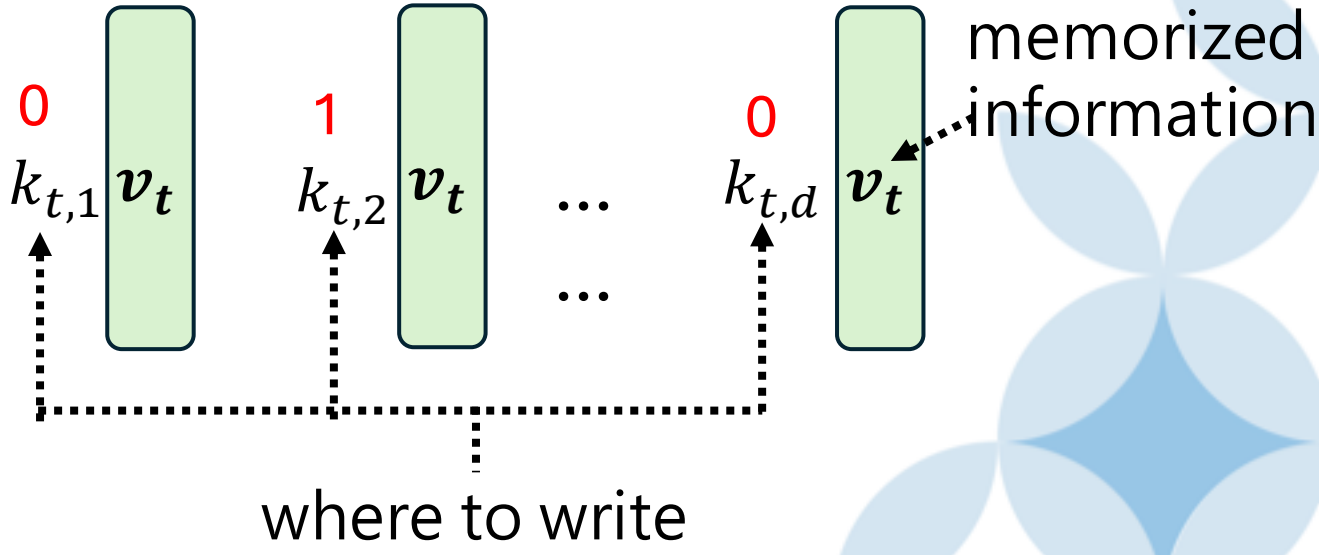


d dim $q_t = W_Q x_t$

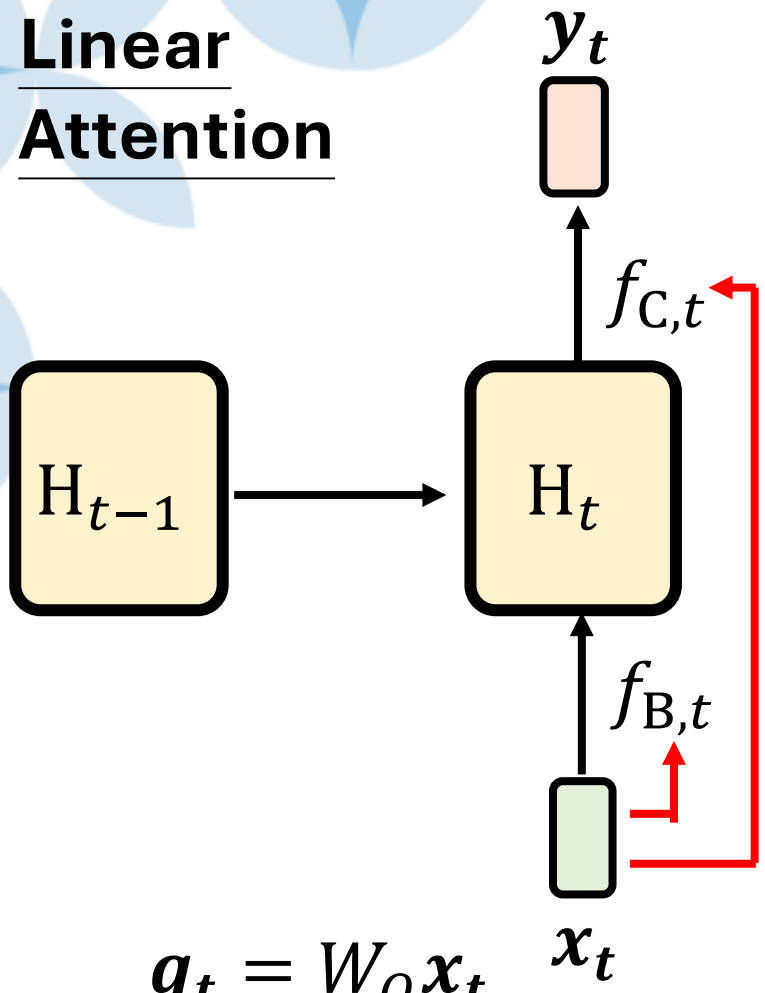
d dim $k_t = W_k x_t$

d' dim $v_t = W_v x_t$

Write v_t into 2nd column of H



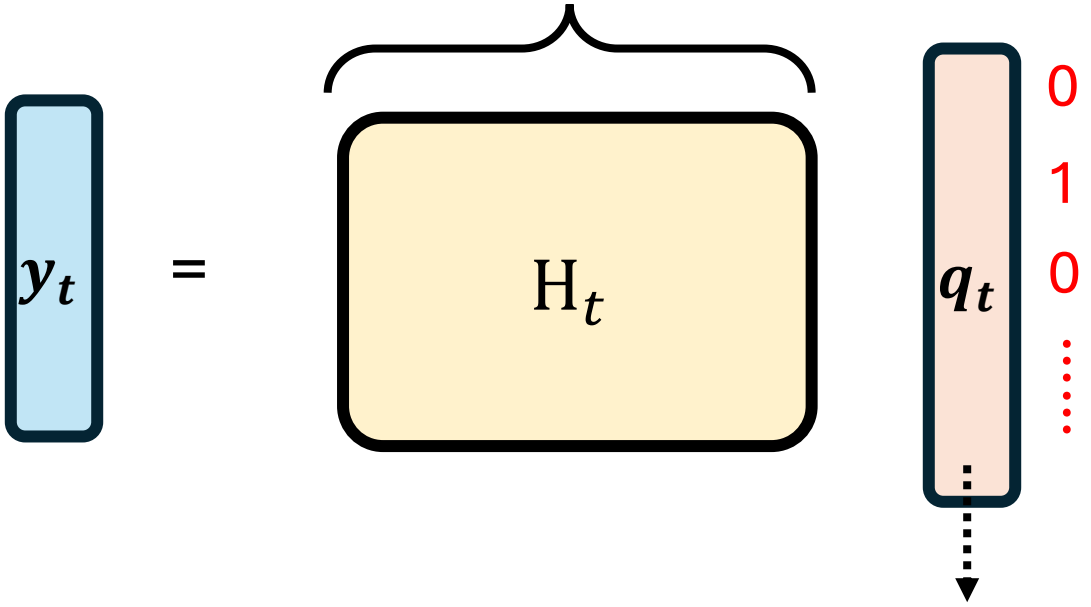
Linear Attention



$$H_t = H_{t-1} + f_{B,t}(x_t) \qquad f_{B,t}(x_t) = v_t k_t^T$$

$$y_t = f_{C,t}(H_t) \qquad f_{C,t}(H_t) = H_t q_t$$

Different information into different Columns



How much information to take from each column

d dim $\quad q_t = W_Q x_t$

d dim $\quad k_t = W_k x_t$

d' dim $\quad v_t = W_v x_t$

This is not a new idea

Transformers are RNNs:
Fast Autoregressive Transformers with Linear Attention

Angelos Katharopoulos [1 2] **Apoorv Vyas** [1 2] **Nikolaos Pappas** [3] **François Fleuret** [2 4 *]

Abstract

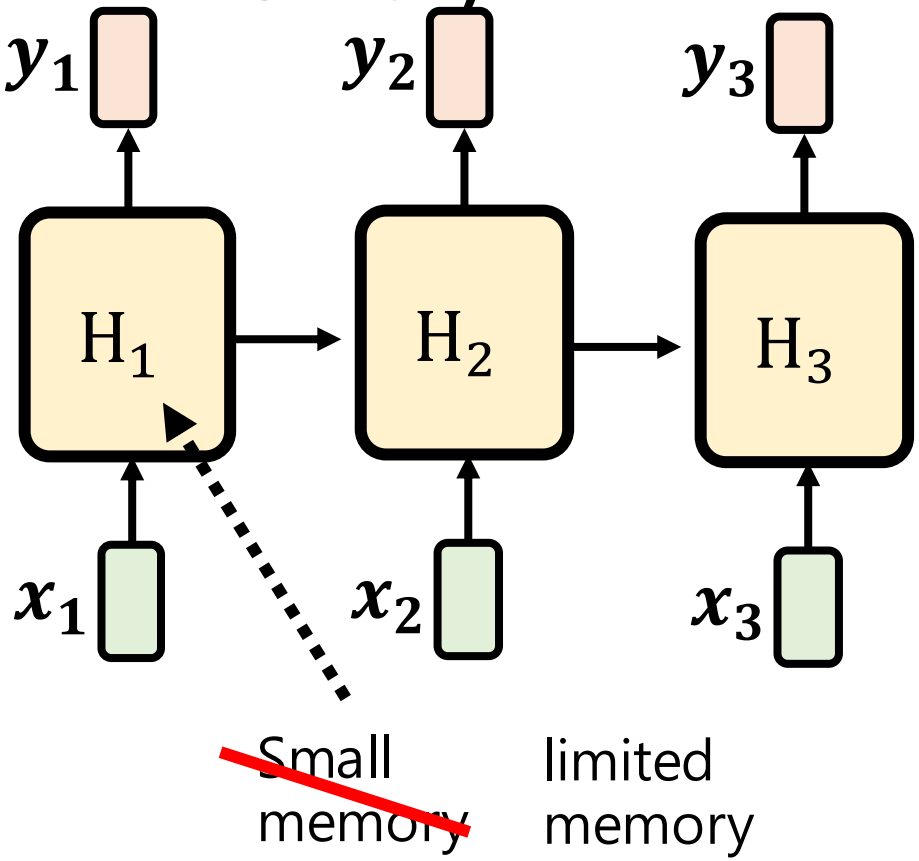
Transformers achieve remarkable performance in several tasks but due to their quadratic complexity, with respect to the input's length, they are prohibitively slow for very long sequences. To address this limitation, we express the self-attention as a linear dot-product of kernel feature maps and make use of the associativity property of matrix

by the global receptive field of self-attention, which processes contexts of N inputs with a quadratic memory and time complexity $\mathcal{O}\left(N^2\right)$. As a result, in practice transformers are slow to train and their context is *limited*. This disrupts temporal coherence and hinders the capturing of long-term dependencies. Dai et al. (2019) addressed the latter by attending to memories from previous contexts albeit at the expense of computational efficiency.

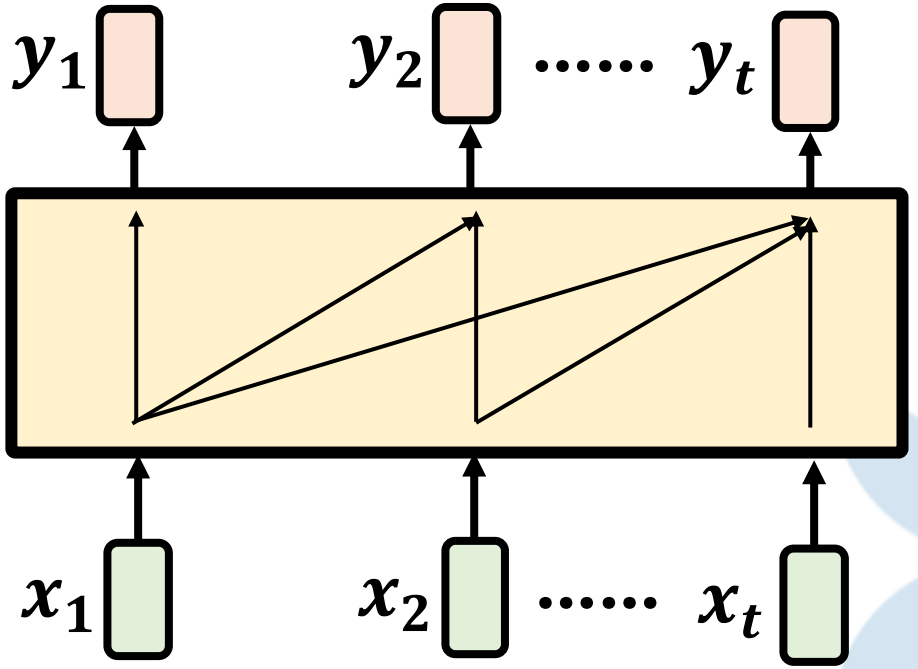
https://arxiv.org/abs/2006.16236

RNN (Linear Attention) < Transformer (Self-attention with Softmax) ?

RNN (Linear Attention)

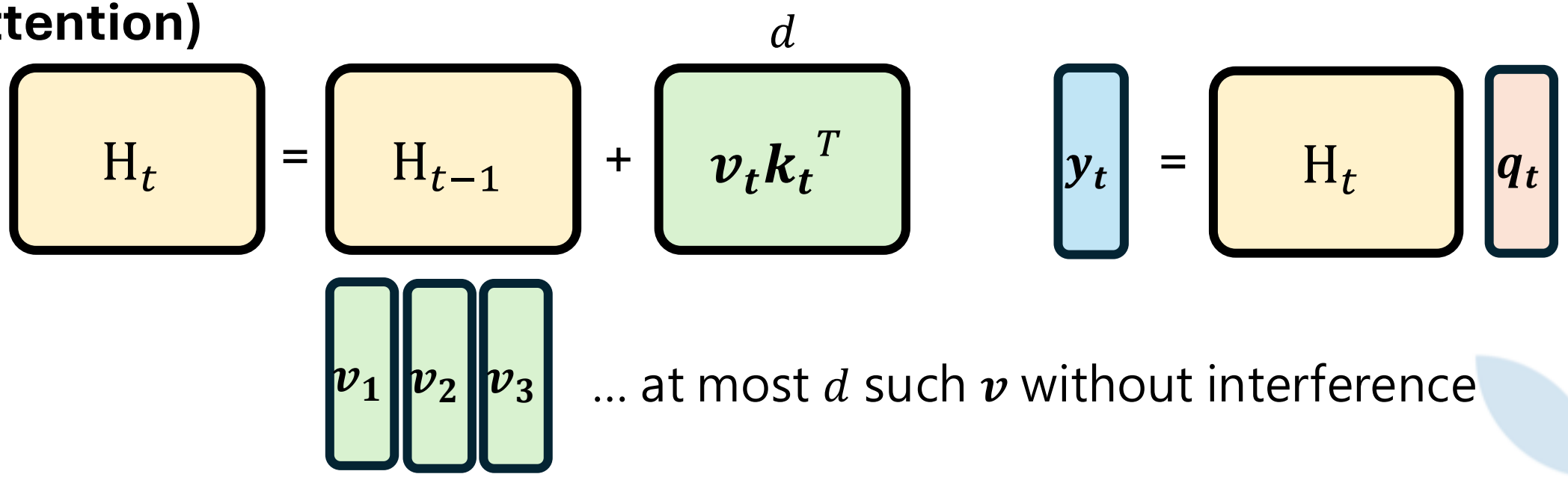


Transformer (Self-attention with softmax)



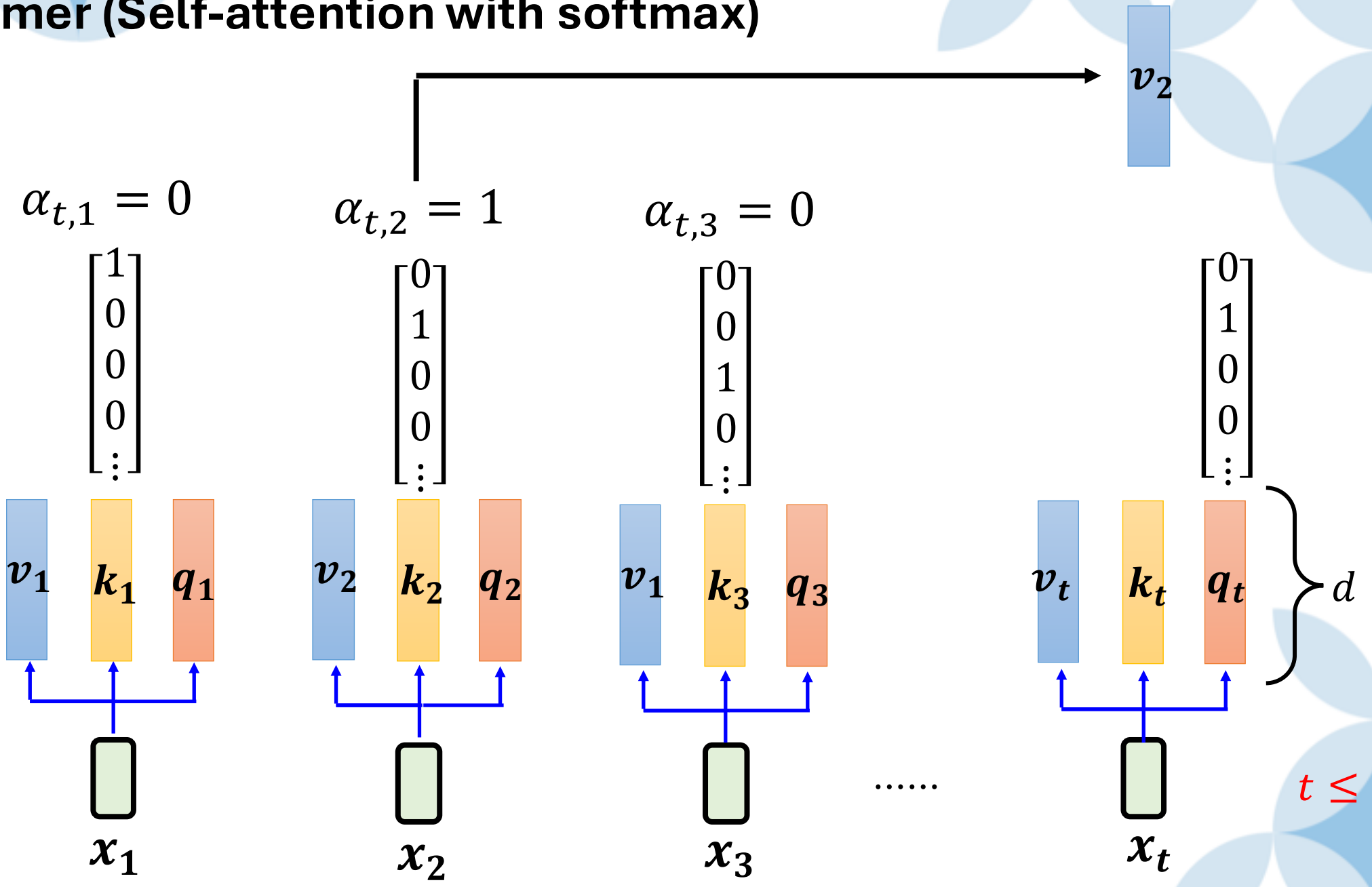
RNN (Linear Attention) < Transformer (Self-attention with Softmax) ?

RNN (Linear Attention)

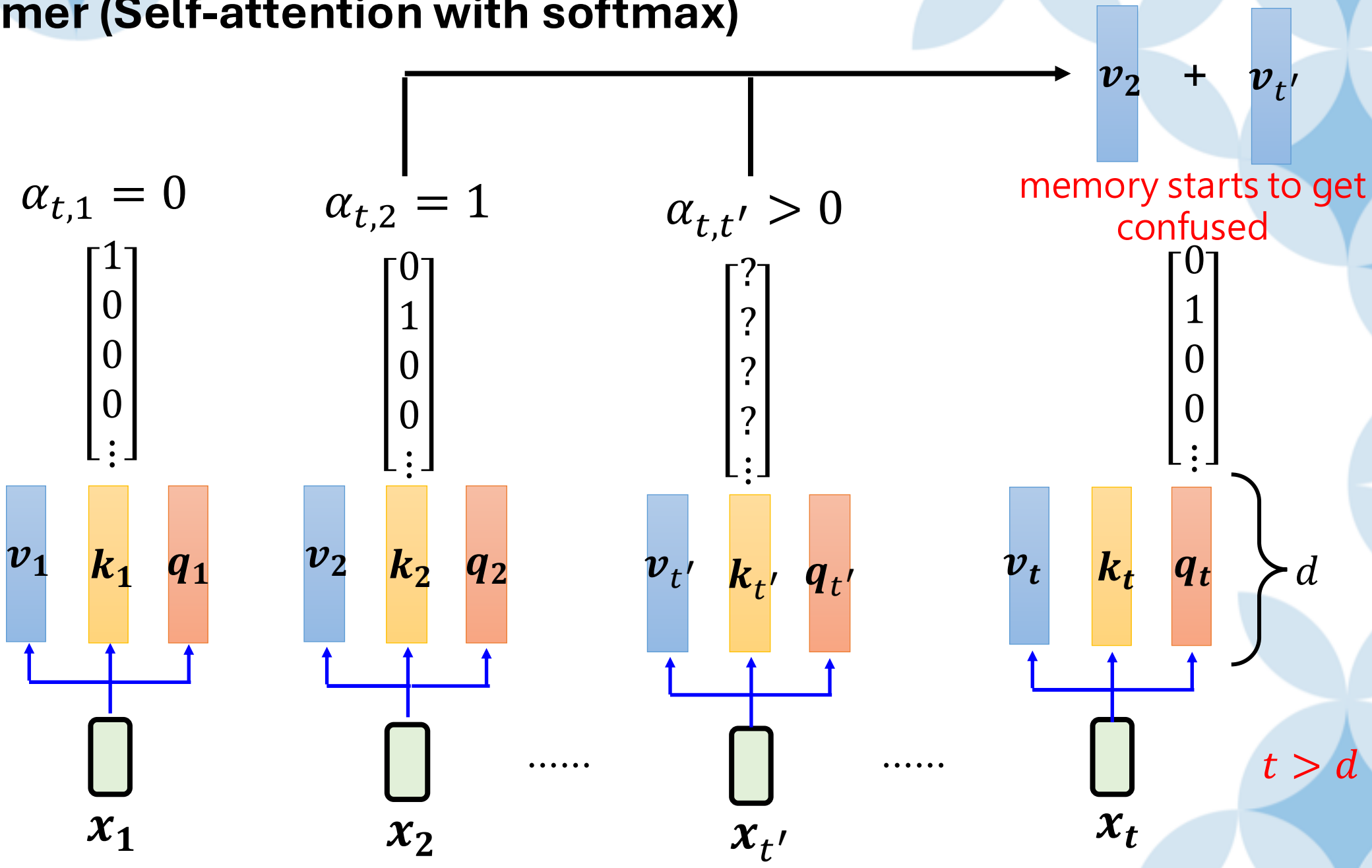


$k_1^T = [1 \quad 0 \quad ...]$ $k_2^T = [0 \quad 1 \quad ...]$ $k_3^T = [0 \quad 0 \quad 1 ...]$

Transformer (Self-attention with softmax)



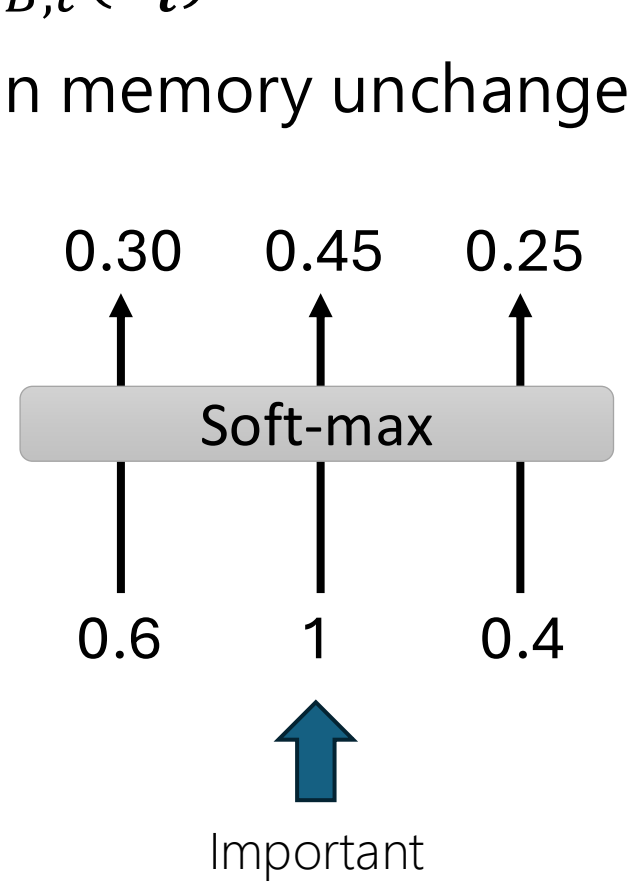
Transformer (Self-attention with softmax)

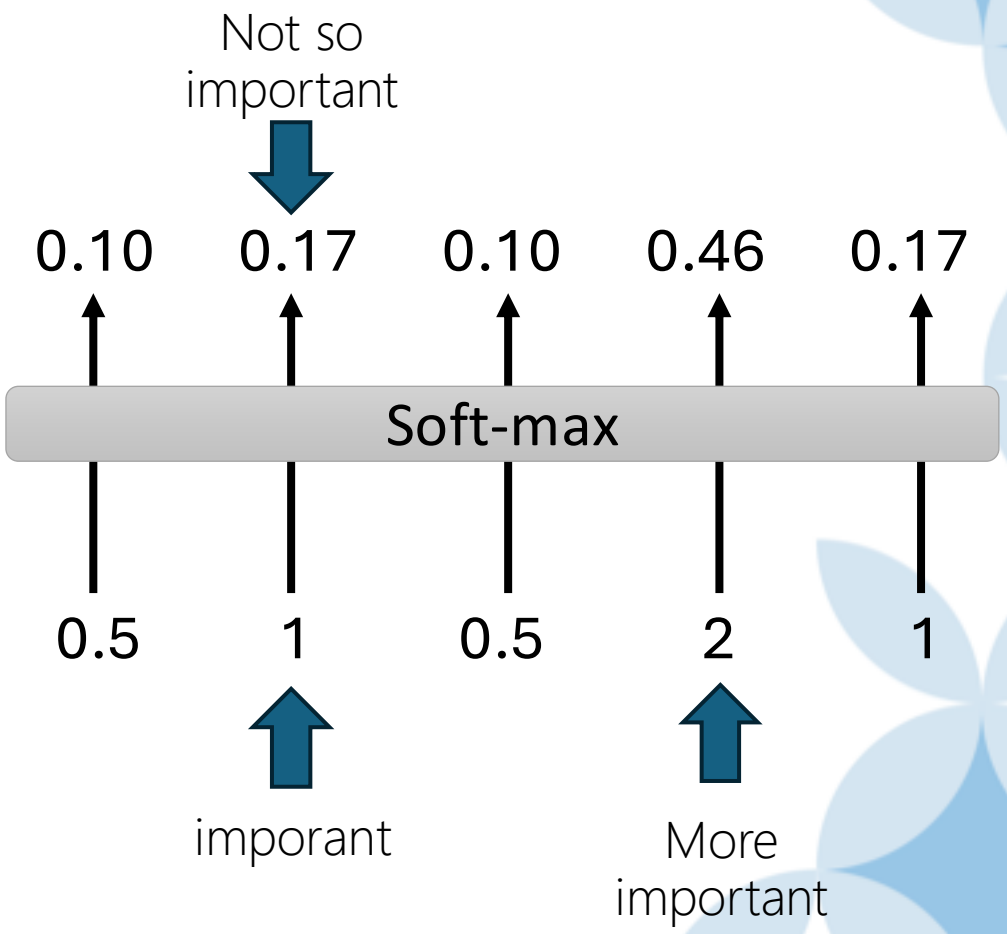


RNN (Linear Attention) < Transformer (Self-attention with Softmax) ?

$$\mathrm{H}_t = \mathrm{H}_{t-1} + f_{B,t}(\boldsymbol{x_t})$$

Linear Attention memory unchangedd





+ Reflection: gradually forget

Linear Attention

$$\mathrm{H}_t = \mathrm{H}_{t-1} + \boldsymbol{v_t}\boldsymbol{k_t}^T$$

$$\boldsymbol{y_t} = \mathrm{H}_t\boldsymbol{q_t}$$

$$\boldsymbol{v_t} = W_v\boldsymbol{x_t}$$

$$\boldsymbol{k_t} = W_k\boldsymbol{x_t}$$

$$\boldsymbol{q_t} = W_Q\boldsymbol{x_t}$$

Retention Network (RetNet)

$$\mathrm{H}_t = \gamma\mathrm{H}_{t-1} + \boldsymbol{v_t}\boldsymbol{k_t}^T$$

$$\boldsymbol{y_t} = \mathrm{H}_t\boldsymbol{q_t}$$

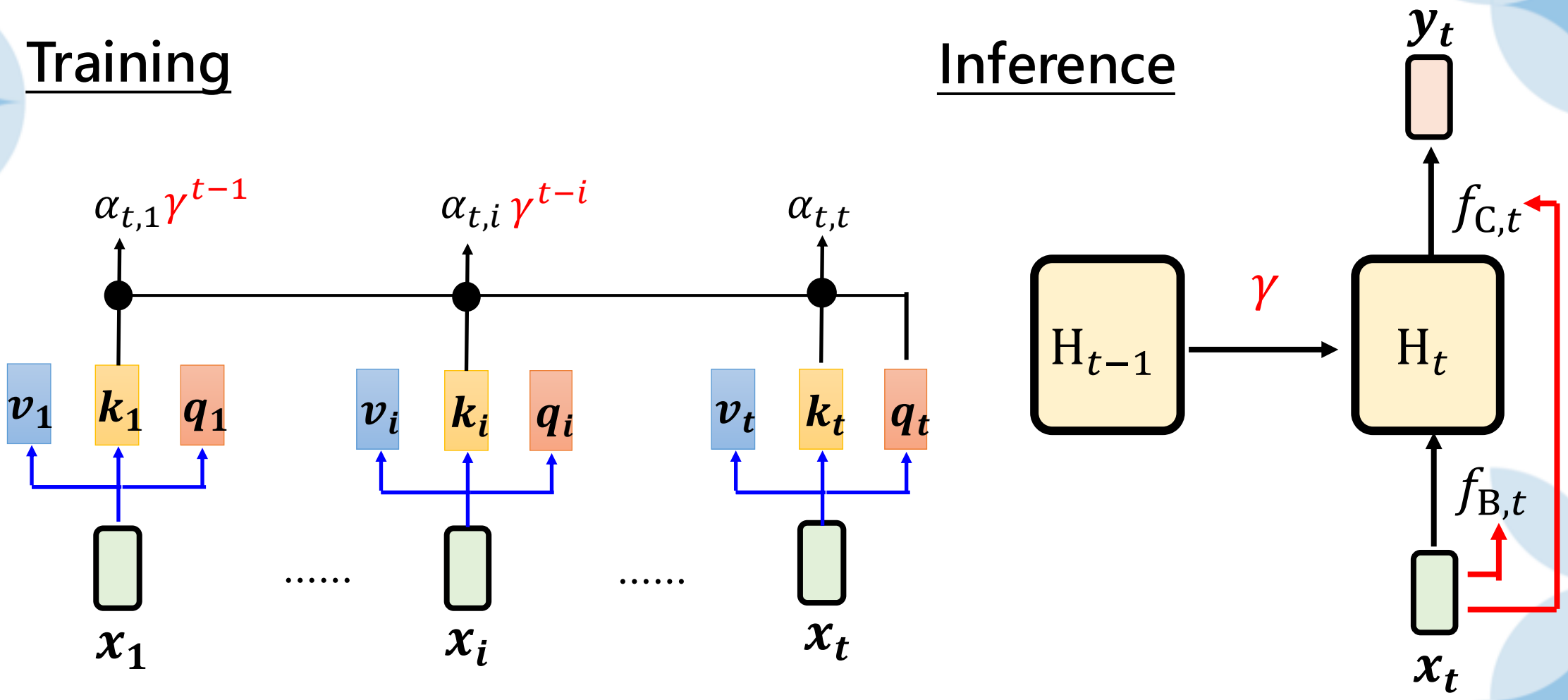
$$\boldsymbol{v_t} = W_v\boldsymbol{x_t}$$

$$\boldsymbol{k_t} = W_k\boldsymbol{x_t}$$

$$\boldsymbol{q_t} = W_Q\boldsymbol{x_t}$$

https://arxiv.org/abs/2307.08621

+ Reflection: gradually forget



+ Reflection: forget as needed

https://arxiv.org/abs/2405.05254

Retention Network (RetNet)

$$\mathrm{H}_t = \gamma \mathrm{H}_{t-1} + \boldsymbol{v_t}\boldsymbol{k_t}^T$$

$$\boldsymbol{y_t} = \mathrm{H}_t \boldsymbol{q_t}$$

$$\boldsymbol{v_t} = W_v \boldsymbol{x_t}$$

$$\boldsymbol{k_t} = W_k \boldsymbol{x_t}$$

$$\boldsymbol{q_t} = W_Q \boldsymbol{x_t}$$

Gated Retention

$$\mathrm{H}_t = \gamma_t \mathrm{H}_{t-1} + \boldsymbol{v_t}\boldsymbol{k_t}^T$$

$$\boldsymbol{y_t} = \mathrm{H}_t \boldsymbol{q_t}$$

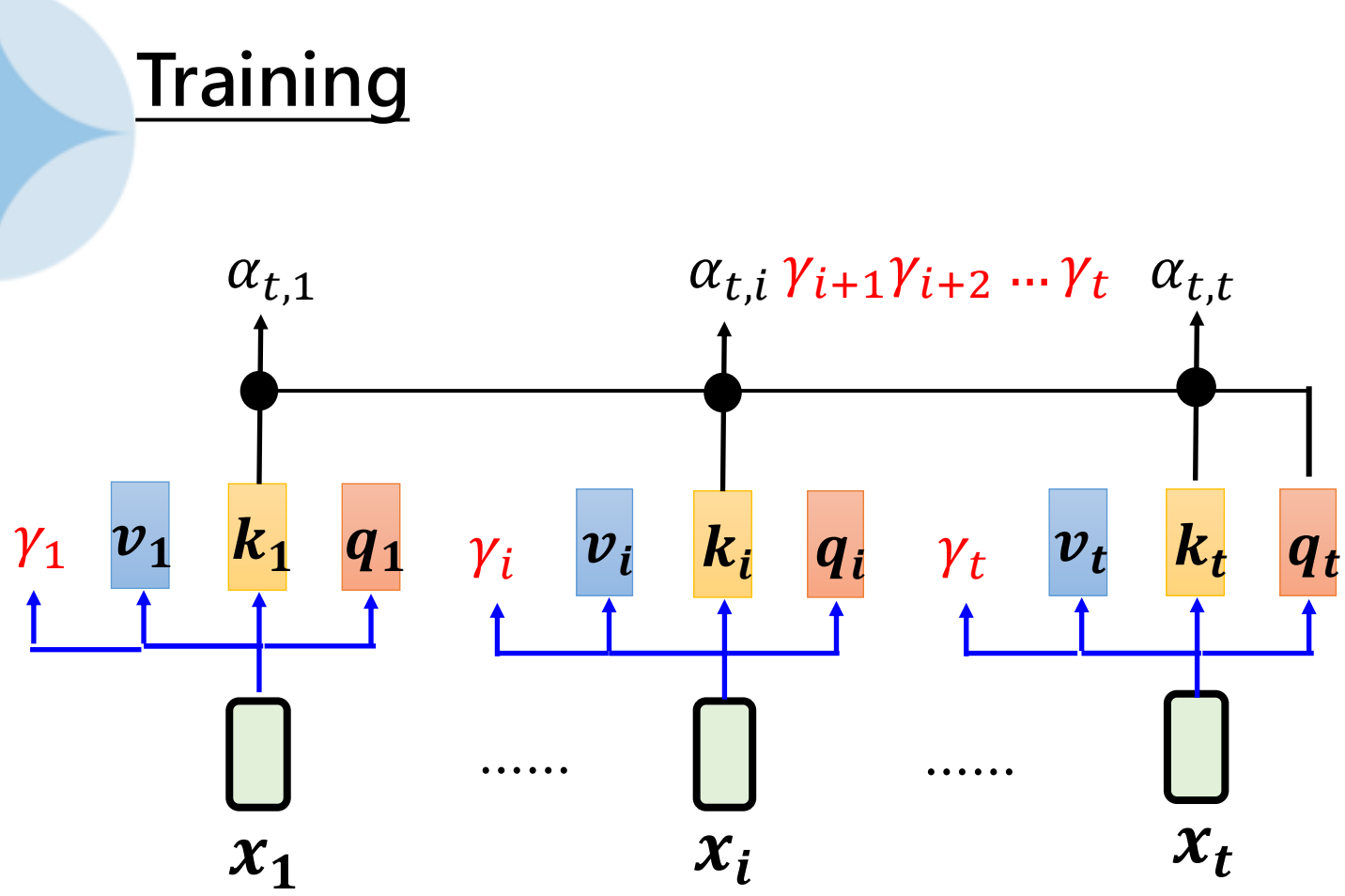
$$\boldsymbol{v_t} = W_v \boldsymbol{x_t}$$

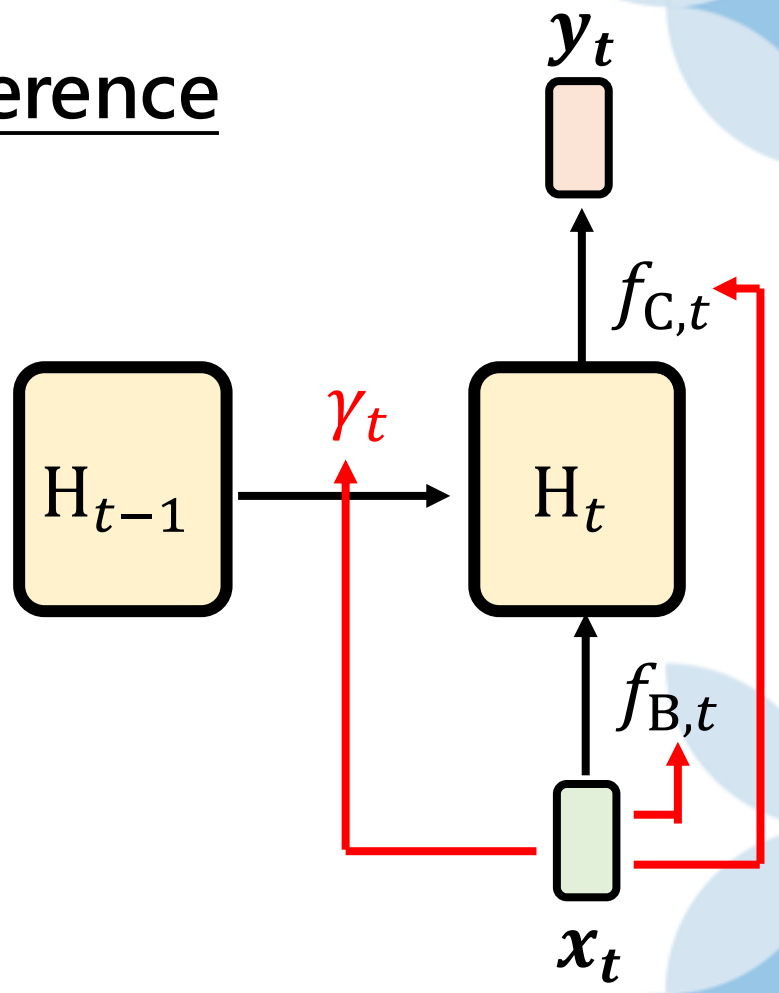
$$\boldsymbol{k_t} = W_k \boldsymbol{x_t}$$

$$\boldsymbol{q_t} = W_Q \boldsymbol{x_t}$$

$$\gamma_t = sigmoid\left(W_\gamma \boldsymbol{x_t}\right)$$

+ Reflection: gradually forget





More complex Reflection

$$\mathrm{H}_t = G_t \odot \mathrm{H}_{t-1} + \boldsymbol{v_t k_t}^T$$

$$G_t = \boldsymbol{e_t s_t}^T$$

$$G_t = \mathbf{1}\boldsymbol{s_t}^T$$

$$\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$\boldsymbol{s_t}^T = [0 \quad 1 \quad 0.1 \quad \dots\dots]$$

$$\mathbf{1}\,\boldsymbol{s_t}^T \odot \mathrm{H}_{t-1}$$

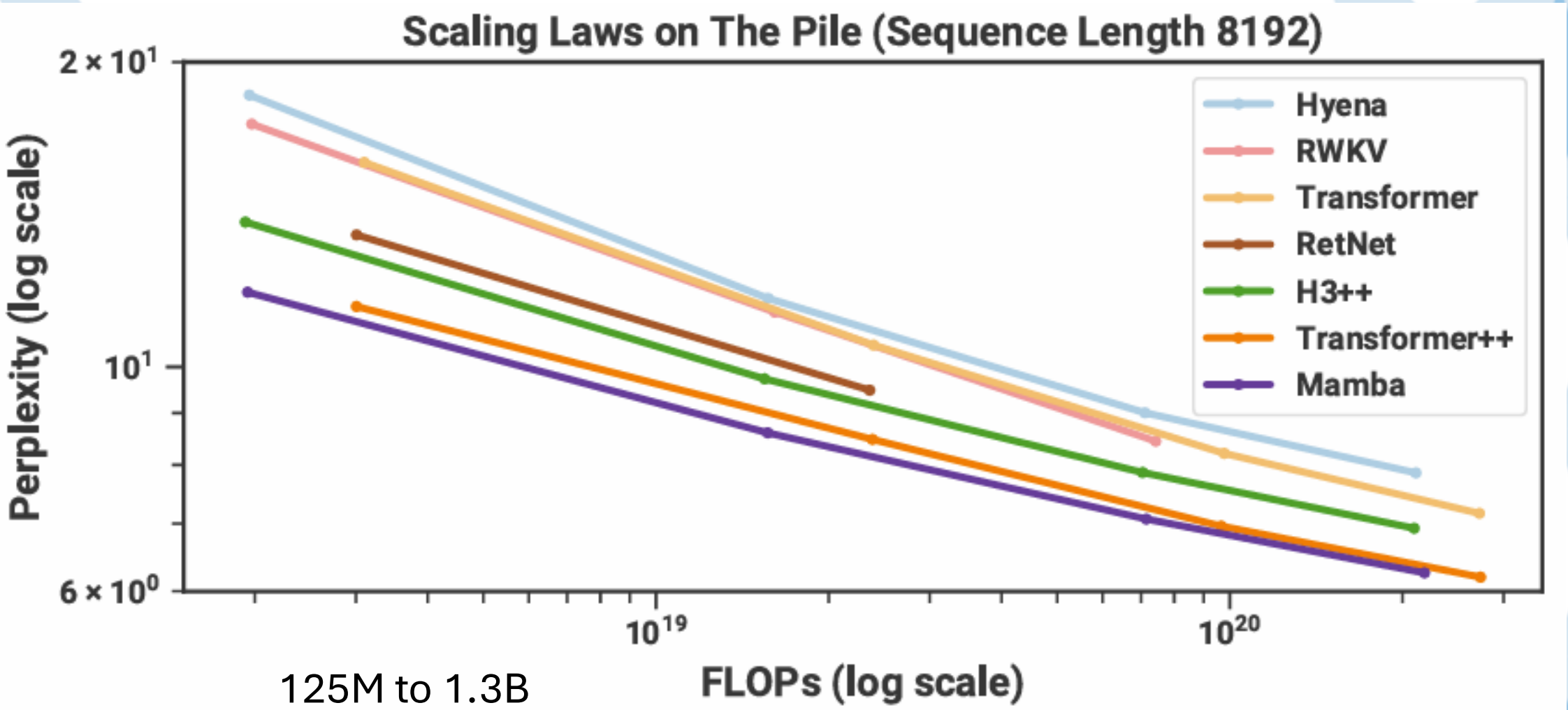
$$\begin{bmatrix} s_{t,1} & s_{t,2} & s_{t,3} & \cdots & s_{t,d} \\ s_{t,1} & s_{t,2} & s_{t,3} & \cdots & s_{t,d} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ s_{t,1} & s_{t,2} & s_{t,3} & \cdots & s_{t,d} \end{bmatrix} \odot \mathrm{H}_{t-1}$$

$$0 \qquad 1 \qquad 0.1$$

remove keep reduce

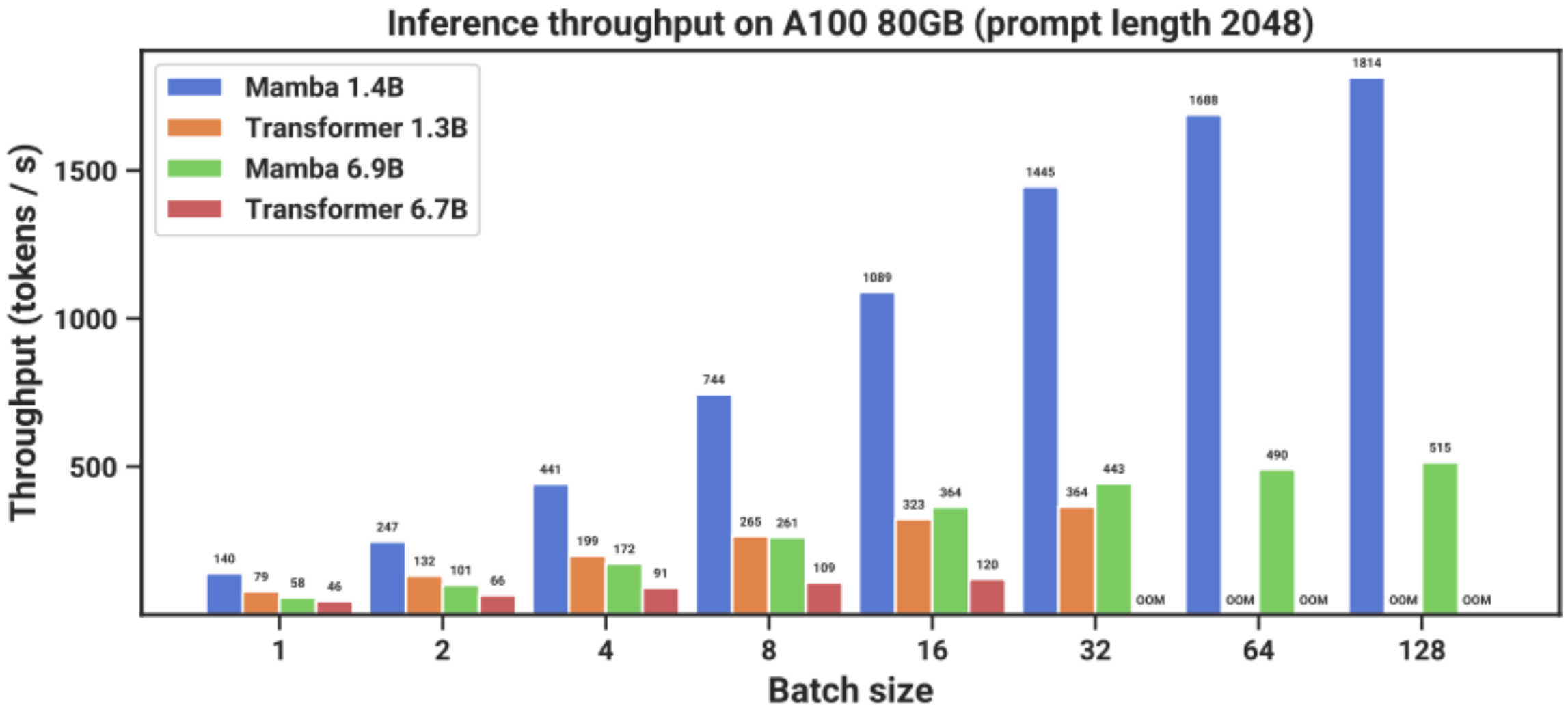
Model	Recurrence	Memory read-out
Linear Attention [48, 47]	$\mathbf{S}_t = \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$
+ Kernel	$\mathbf{S}_t = \mathbf{S}_{t-1} + \boldsymbol{v}_t \phi(\boldsymbol{k}_t)^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \phi(\boldsymbol{q}_t)$
+ Normalization	$\mathbf{S}_t = \mathbf{S}_{t-1} + \boldsymbol{v}_t \phi(\boldsymbol{k}_t)^\top, \quad \boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \phi(\boldsymbol{k}_t)$	$\boldsymbol{o}_t = \mathbf{S}_t \phi(\boldsymbol{q}_t) / (\boldsymbol{z}_t^\top \phi(\boldsymbol{q}_t))$
DeltaNet [101]	$\mathbf{S}_t = \mathbf{S}_{t-1}(\mathbf{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top) + \beta_t \boldsymbol{v}_t \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$
Gated RFA [81]	$\mathbf{S}_t = g_t \mathbf{S}_{t-1} + (1 - g_t) \boldsymbol{v}_t \boldsymbol{k}_t^\top, \quad \boldsymbol{z}_t = g_t \boldsymbol{z}_{t-1} + (1 - g_t) \boldsymbol{k}_t$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t / (\boldsymbol{z}_t^\top \boldsymbol{q}_t)$
S4 [32, 106]	$\mathbf{S}_t = \mathbf{S}_{t-1} \odot \exp(-(\boldsymbol{\alpha} \mathbf{1}^\top) \odot \exp(\boldsymbol{A})) + \boldsymbol{B} \odot (\boldsymbol{v}_t \mathbf{1}^\top)$	$\boldsymbol{o}_t = (\mathbf{S}_t \odot \boldsymbol{C}) \mathbf{1} + \boldsymbol{d} \odot \boldsymbol{v}_t$
ABC [82]	$\mathbf{S}_t^{\boldsymbol{k}} = \mathbf{S}_{t-1}^{\boldsymbol{k}} + \boldsymbol{k}_t \boldsymbol{\phi}_t^\top, \quad \mathbf{S}_t^{\boldsymbol{v}} = \mathbf{S}_{t-1}^{\boldsymbol{v}} + \boldsymbol{v}_t \boldsymbol{\phi}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t^{\boldsymbol{v}} \operatorname{softmax}\left(\mathbf{S}_t^{\boldsymbol{k}} \boldsymbol{q}_t\right)$
DFW [63]	$\mathbf{S}_t = \mathbf{S}_{t-1} \odot (\boldsymbol{\beta}_t \boldsymbol{\alpha}_t^\top) + \boldsymbol{v}_t \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$
RetNet [108]	$\mathbf{S}_t = \gamma \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$
Mamba [31]	$\mathbf{S}_t = \mathbf{S}_{t-1} \odot \exp(-(\boldsymbol{\alpha}_t \mathbf{1}^\top) \odot \exp(\boldsymbol{A})) + (\boldsymbol{\alpha}_t \odot \boldsymbol{v}_t) \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t + \boldsymbol{d} \odot \boldsymbol{v}_t$
GLA [124]	$\mathbf{S}_t = \mathbf{S}_{t-1} \odot (\mathbf{1} \boldsymbol{\alpha}_t^\top) + \boldsymbol{v}_t \boldsymbol{k}_t^\top = \mathbf{S}_{t-1} \operatorname{Diag}(\boldsymbol{\alpha}_t) + \boldsymbol{v}_t \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$
RWKV-6 [79]	$\mathbf{S}_t = \mathbf{S}_{t-1} \operatorname{Diag}(\boldsymbol{\alpha}_t) + \boldsymbol{v}_t \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = (\mathbf{S}_{t-1} + (\boldsymbol{d} \odot \boldsymbol{v}_t) \boldsymbol{k}_t^\top) \boldsymbol{q}_t$
HGRN-2 [92]	$\mathbf{S}_t = \mathbf{S}_{t-1} \operatorname{Diag}(\boldsymbol{\alpha}_t) + \boldsymbol{v}_t (\mathbf{1} - \boldsymbol{\alpha}_t)^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$
mLSTM [9]	$\mathbf{S}_t = f_t \mathbf{S}_{t-1} + i_t \boldsymbol{v}_t \boldsymbol{k}_t^\top, \quad \boldsymbol{z}_t = f_t \boldsymbol{z}_{t-1} + i_t \boldsymbol{k}_t$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t / \max\{1, \lvert \boldsymbol{z}_t^\top \boldsymbol{q}_t \rvert\}$
Mamba-2 [19]	$\mathbf{S}_t = \gamma_t \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$
GSA [131]	$\mathbf{S}_t^{\boldsymbol{k}} = \mathbf{S}_{t-1}^{\boldsymbol{k}} \operatorname{Diag}(\boldsymbol{\alpha}_t) + \boldsymbol{k}_t \boldsymbol{\phi}_t^\top, \quad \mathbf{S}_t^{\boldsymbol{v}} = \mathbf{S}_{t-1}^{\boldsymbol{v}} \operatorname{Diag}(\boldsymbol{\alpha}_t) + \boldsymbol{v}_t \boldsymbol{\phi}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t^{\boldsymbol{v}} \operatorname{softmax}\left(\mathbf{S}_t^{\boldsymbol{k}} \boldsymbol{q}_t\right)$
Gated DeltaNet [125]	$\mathbf{S}_t = \mathbf{S}_{t-1} \left(\alpha_t (\mathbf{I} - \beta_t \boldsymbol{k}_t \boldsymbol{k}_t^\top)\right) + \beta_t \boldsymbol{v}_t \boldsymbol{k}_t^\top$	$\boldsymbol{o}_t = \mathbf{S}_t \boldsymbol{q}_t$

https://arxiv.org/abs/2406.06484



125M to 1.3B

https://arxiv.org/abs/2312.00752



https://arxiv.org/abs/2312.00752

DeltaNet

https://arxiv.org/abs/2406.06484

$$\mathrm{H}_t = \mathrm{H}_{t-1}\left(I - \beta_t \boldsymbol{k_t}\boldsymbol{k_t}^T\right) + \beta_t \boldsymbol{v_t}\boldsymbol{k_t}^T$$

$$\mathrm{H}_t = \mathrm{H}_{t-1} + \boldsymbol{v_t}\boldsymbol{k_t}^T \longleftarrow$$ Gradient Descent, with different L_t

$$\mathrm{H}_t = \mathrm{H}_{t-1} - \boldsymbol{v_{t,old}}\boldsymbol{k_t}^T + \boldsymbol{v_t}\boldsymbol{k_t}^T \qquad \boldsymbol{v_{t,old}} = \mathrm{H}_{t-1}\boldsymbol{k_t}$$

$$\mathrm{H}_t = \mathrm{H}_{t-1} - \beta_t \boldsymbol{v_{t,old}}\boldsymbol{k_t}^T + \beta_t \boldsymbol{v_t}\boldsymbol{k_t}^T$$

$$\mathrm{H}_t = \mathrm{H}_{t-1} - \beta_t \mathrm{H}_{t-1}\boldsymbol{k_t}\boldsymbol{k_t}^T + \beta_t \boldsymbol{v_t}\boldsymbol{k_t}^T$$

Gradient Descent

$$\mathrm{H}_t = \mathrm{H}_{t-1} - \beta_t(\mathrm{H}_{t-1}\boldsymbol{k_t} - \boldsymbol{v_t})\boldsymbol{k_t}^T$$

Parameter after update — before update — learning rate — gradient

$$L_t(H) = \frac{1}{2}\|H\boldsymbol{k_t} - \boldsymbol{v_t}\|^2$$

$$\nabla L_t(\mathrm{H}_{t-1})$$

Update H s.t. info. retreived using $\boldsymbol{k_t}$ is close to $\boldsymbol{v_t}$

Titans: Learning to Memorize at Test Time https://arxiv.org/abs/2501.00663

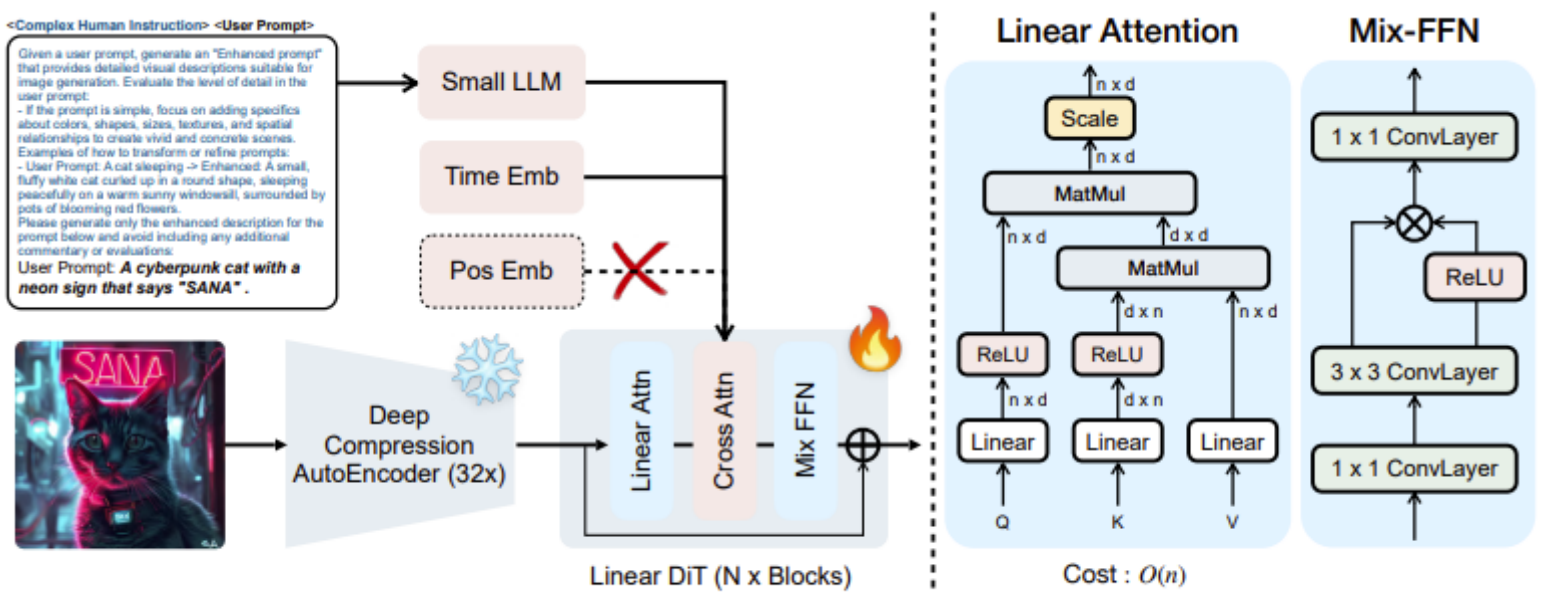
Name	Modality	Affiliations	Sizes	Access Link
Mamba 1&2	Language	Carnegie Mellon University & Princeton University	130M-2.8B	1
Falcon Mamba 7B	Language	Technology Innovation Institute	7B	2
Mistral 7B	Language	Mistral AI & NVIDIA	7B	3
Jamba	Language	AI21 Lab	12B/52B	4
Vision Mamba	Vision	Huazhong University of Science and Technology	7M-98M	5
VideoMamba	Video	OpenGVLab, Shanghai AI Laboratory	28M-392M	6
Codestral Mamba	Code	Mistral AI	7B, 22B	7

1. https://github.com/state-spaces/mamba
2. https://huggingface.co/tiiuae/falcon-mamba-7b
3. https://huggingface.co/mistralai/Mistral-7B-v0.1
4. https://huggingface.co/ai21labs/Jamba-v0.1
5. https://huggingface.co/hustvl/Vim-base-midclstok
6. https://huggingface.co/OpenGVLab/VideoMamba
7. https://mistral.ai/news/codestral-mamba/

https://arxiv.org/abs/2408.01129

Minimax-01

https://arxiv.org/abs/2501.08313



(a). Architecture overview of our Sana.

(b). Linear DiT Module.

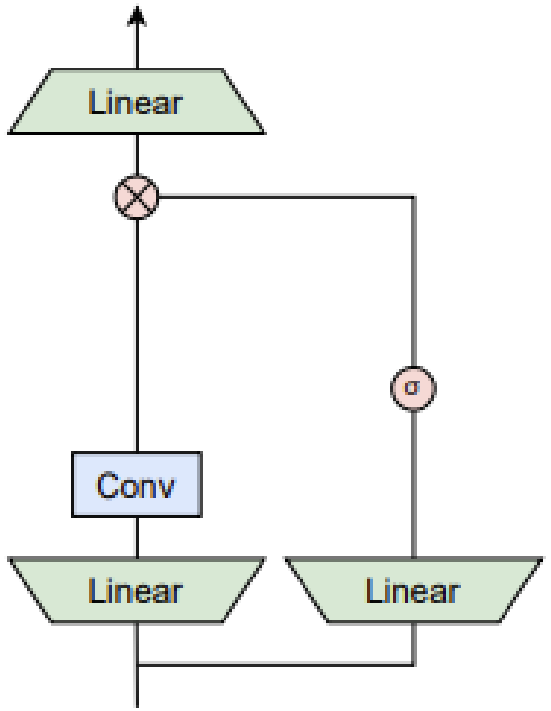
https://arxiv.org/abs/2410.10629

MambaOut: Do We Really Need Mamba for Vision?

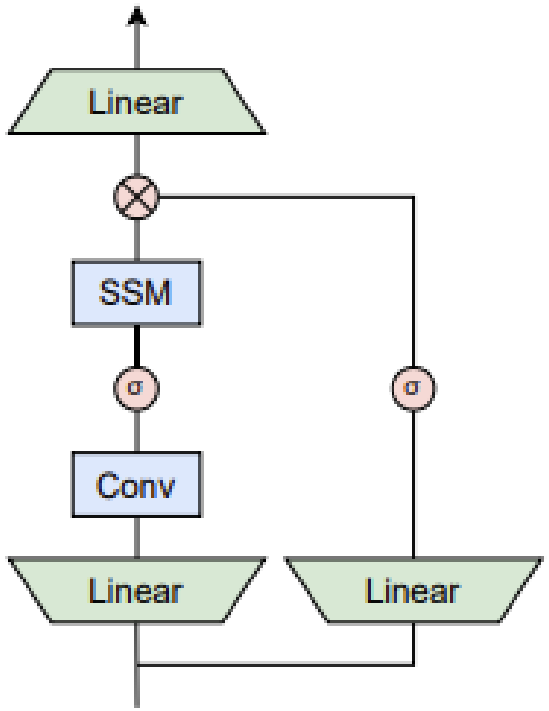
https://arxiv.org/abs/2405.07992

In memory of Kobe Bryant

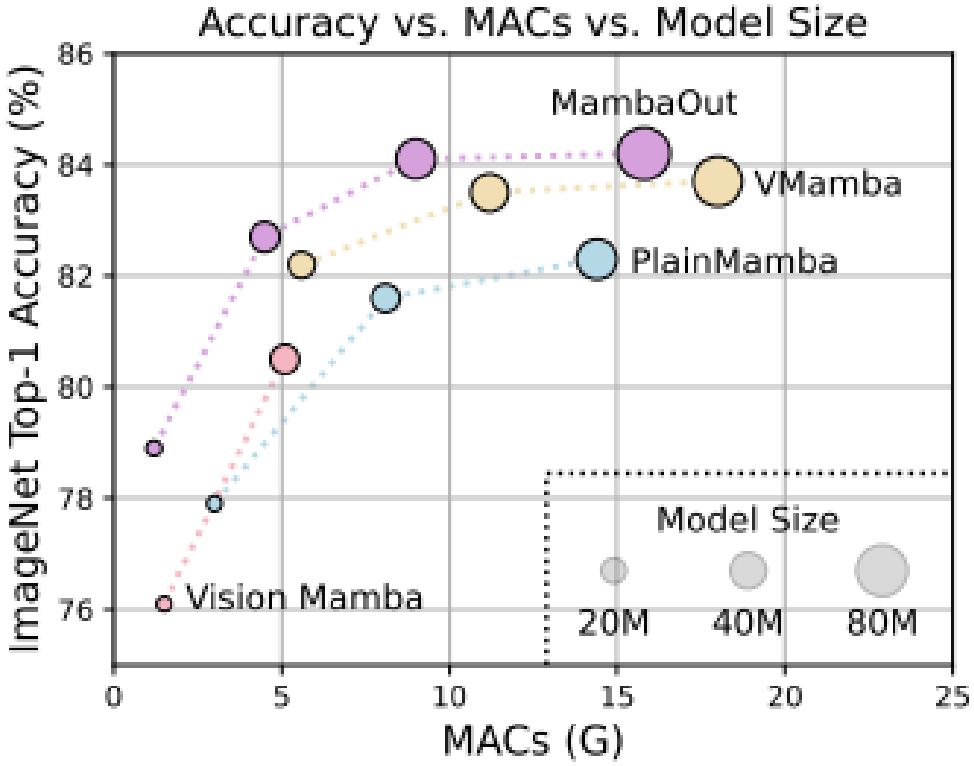
"What can I say, Mamba out." — *Kobe Bryant's NBA farewell speech in 2016.*



Gated CNN block
(e.g. Our MambaOut)

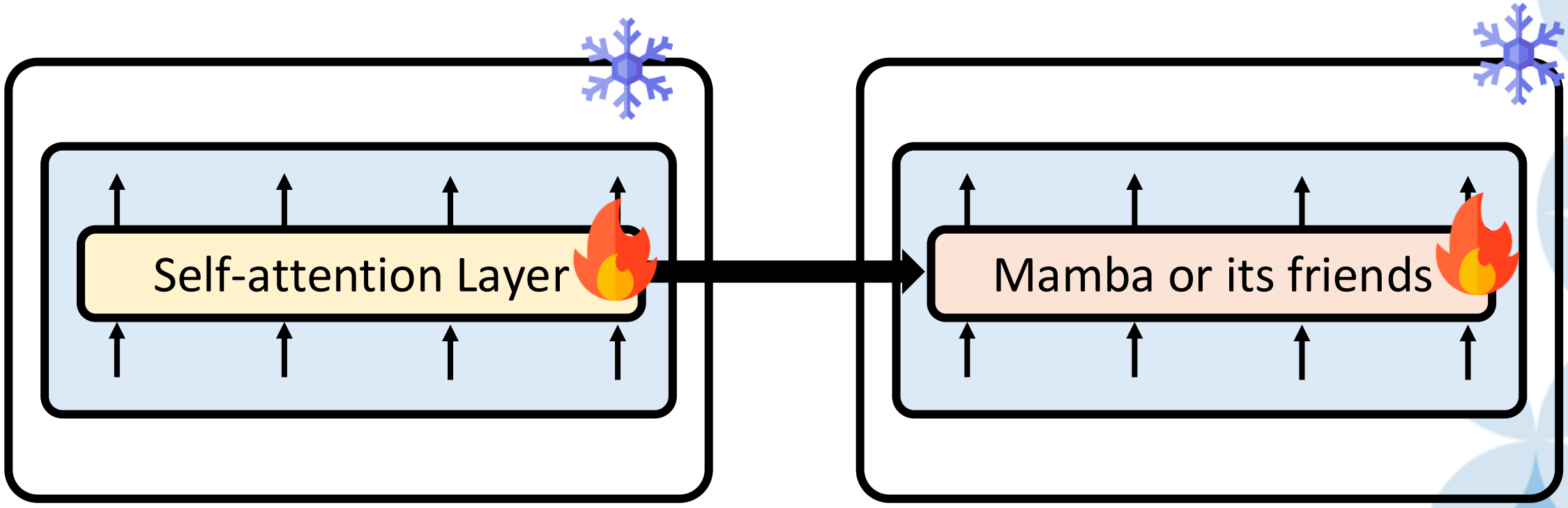


Mamba block
(e.g. Vision Mamba)



Do not train from scratch

Low-rank Linear Conversion via Attention Transfer (LoLCATs), https://arxiv.org/abs/2410.10254
The Mamba in the Llama, https://arxiv.org/abs/2408.15237
Transformers to SSMs, https://arxiv.org/abs/2408.10189
Linger, https://arxiv.org/abs/2503.01496



Is Attention All You Need?

Current Status: Yes

Time Remaining: 302d 20h 44m 32s

https://www.isattentionallyouneed.com/

Proposition:

On January 1, 2027, a Transformer-like model will continue to hold the state-of-the-art position in most benchmarked tasks in natural language processing.

For the Motion

Jonathan Frankle
@jefrankle
Harvard Professor
Chief Scientist Mosaic ML

Against the Motion

Sasha Rush
@srush_nlp
Cornell Professor
Research Scientist Hugging Face 🤗

https://www.isattentionallyouneed.com/

Questions?