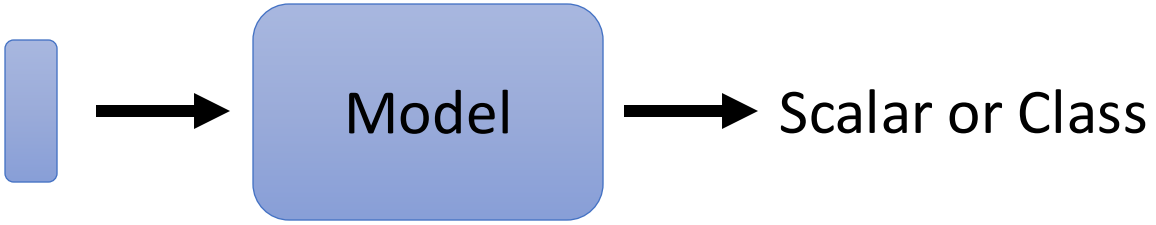
Transformer

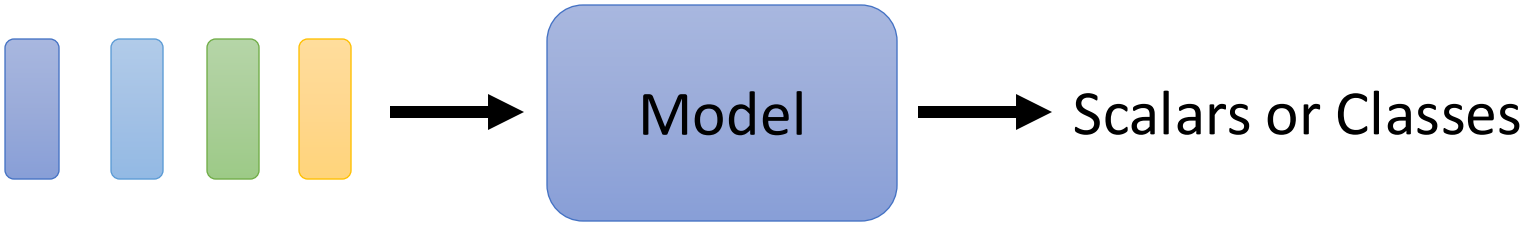
Slides credit: Hung-Yi Lee

Sophisticated Input

- Input is **a vector**

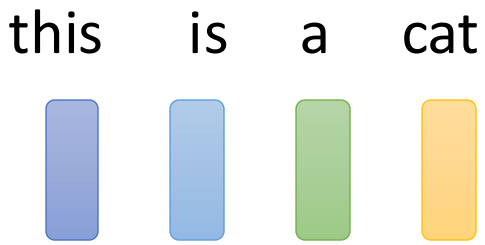


- Input is **a set of vectors**



(may change length)

Vector Set as Input



One-hot Encoding

apple = [1 0 0 0 0 ……]

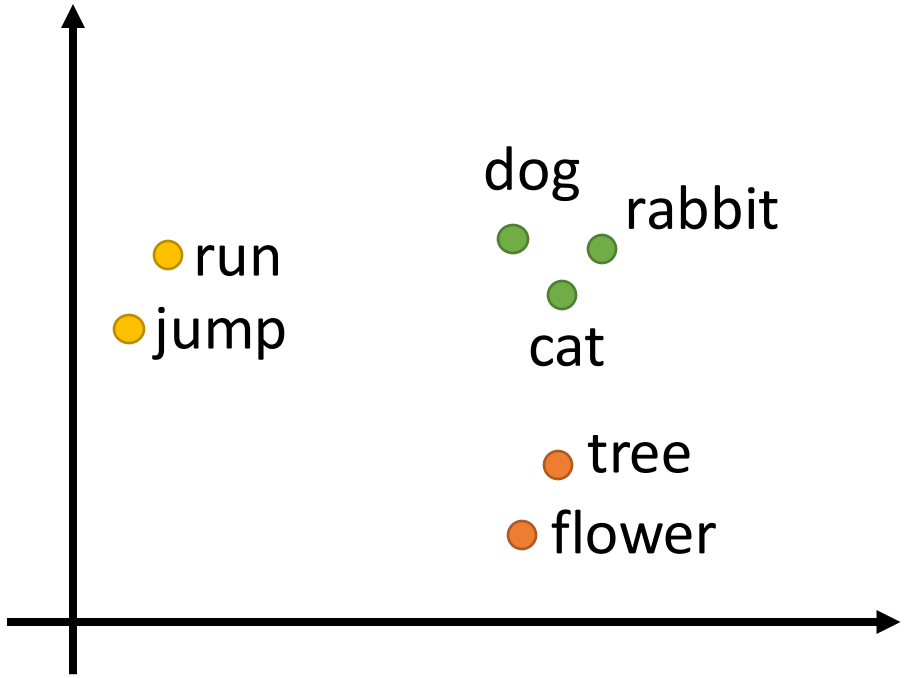
bag = [0 1 0 0 0 ……]

cat = [0 0 1 0 0 ……]

dog = [0 0 0 1 0 ……]

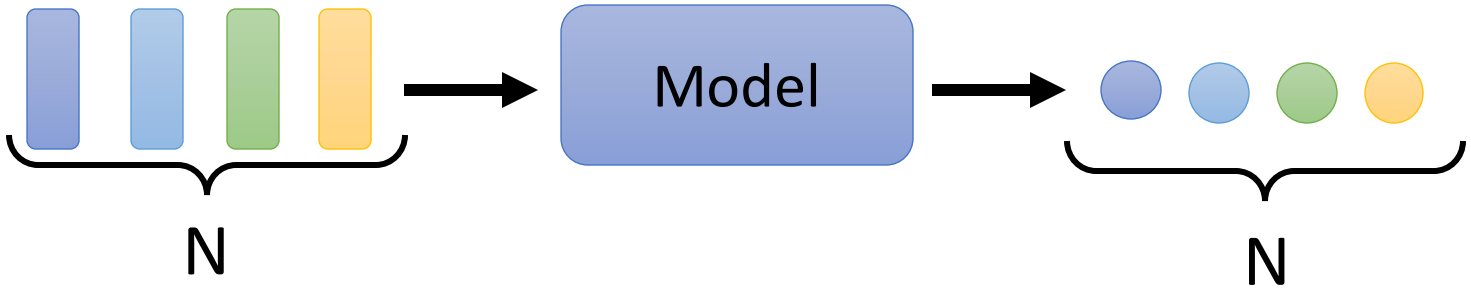
elephant = [0 0 0 0 1 ……]

Word Embedding

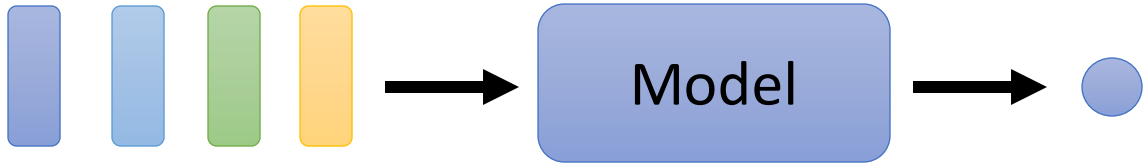


What is the output?

- Each vector has a label.

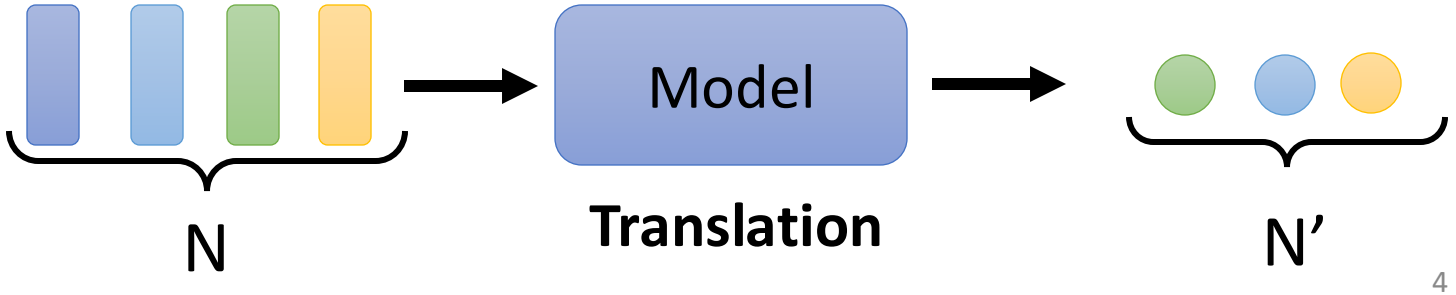


- The whole sequence has a label.



- Model decides the number of labels itself.

seq2seq



Sequence Labeling

FC

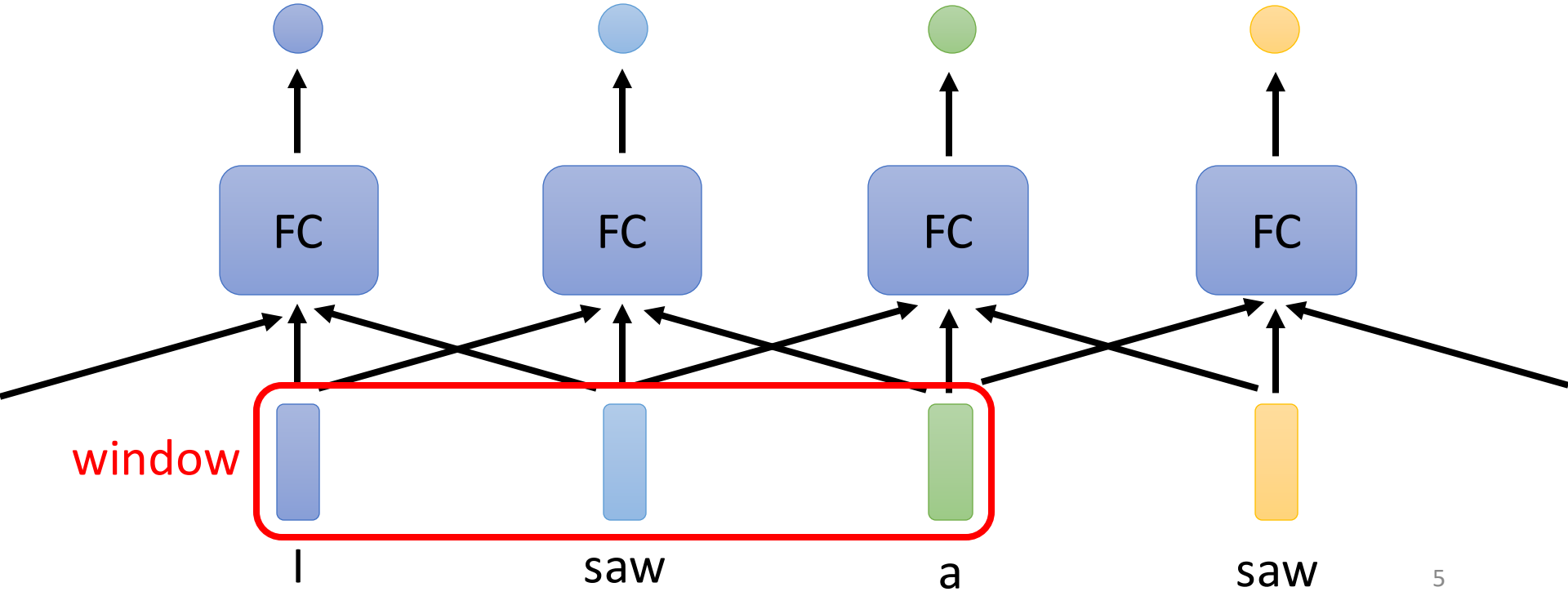
Fully-connected

Is it possible to consider the context?

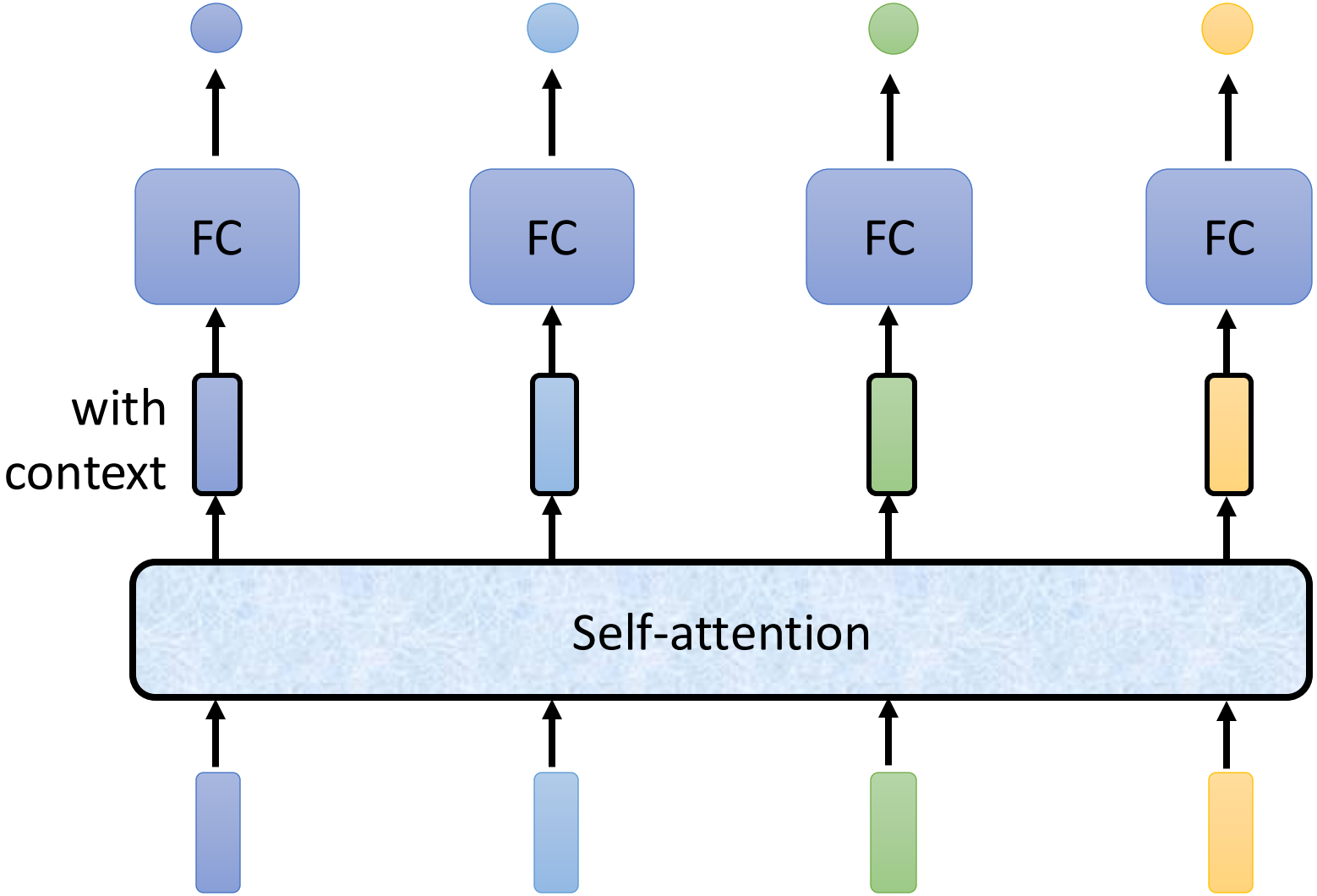
FC can consider the neighbor

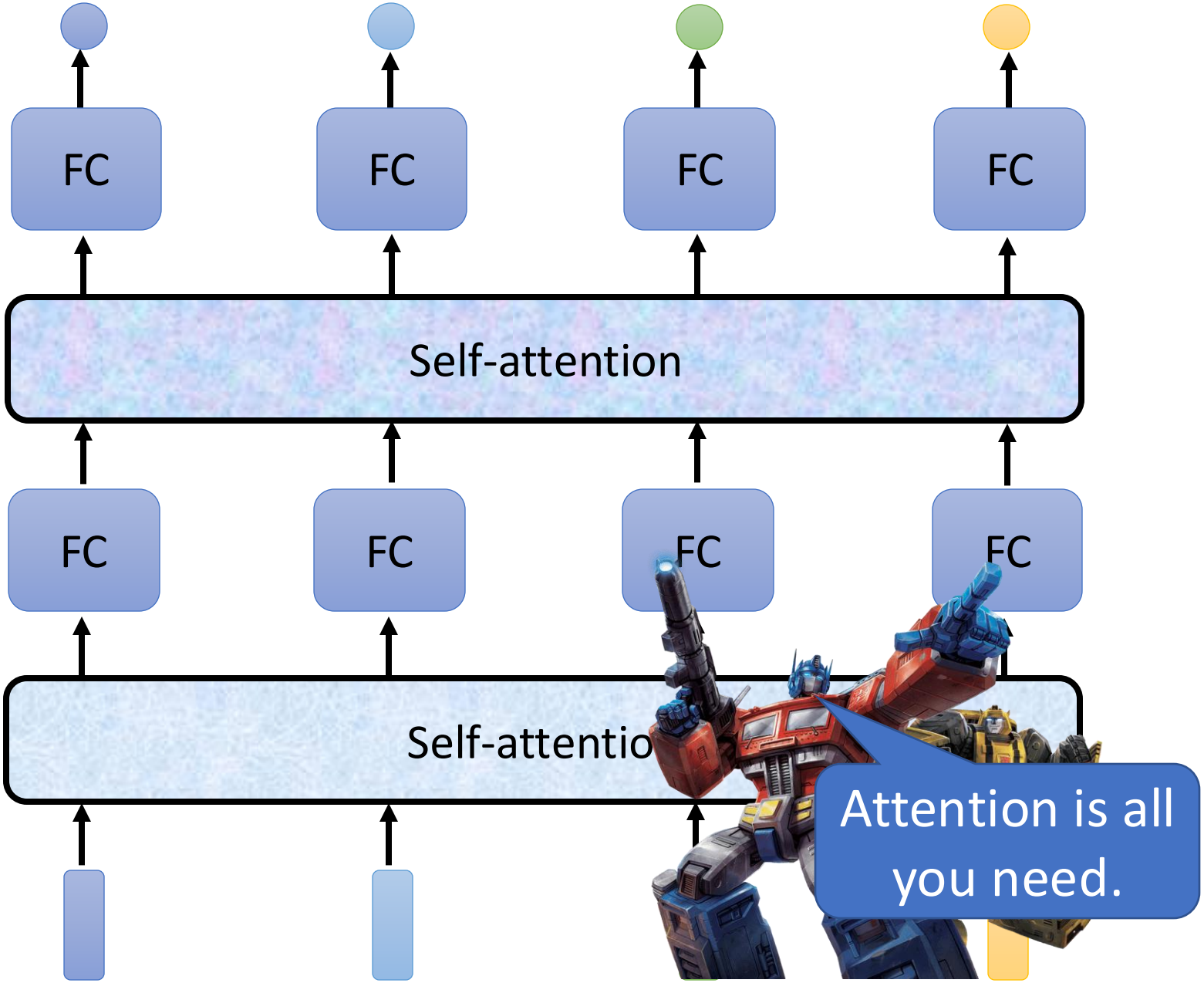
How to consider the whole sequence?

a window covers the whole sequence?



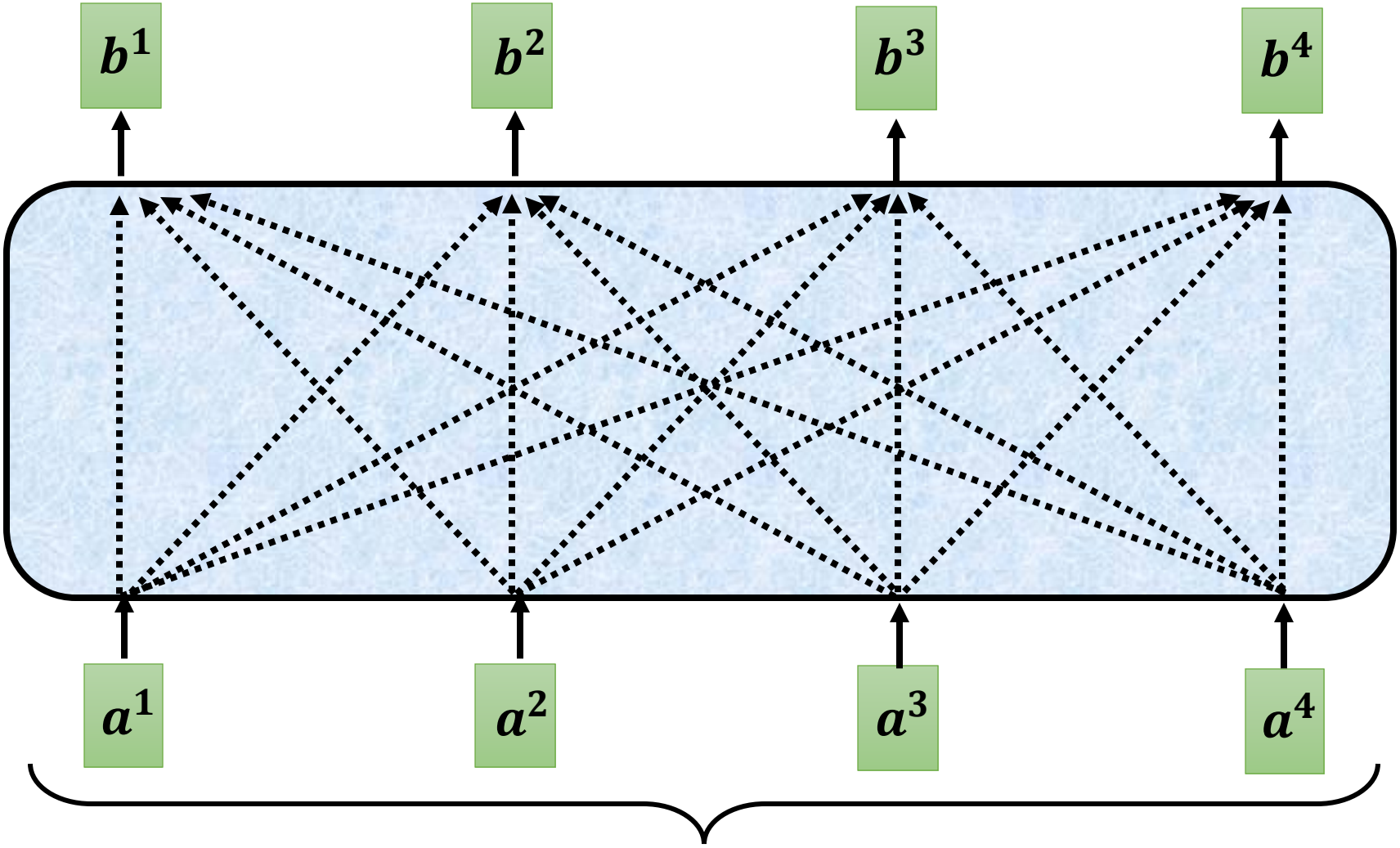
Self-attention





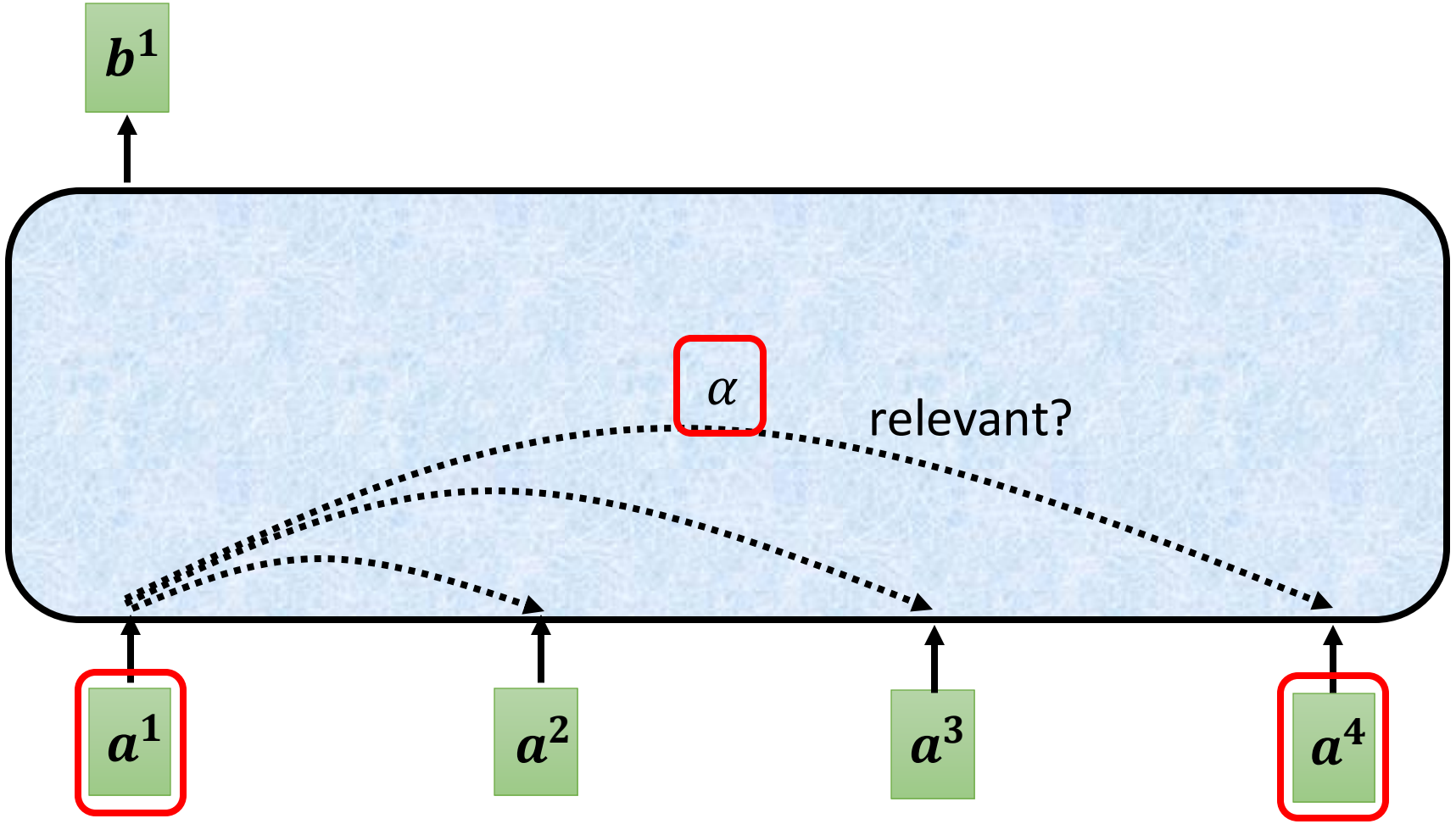
https://arxiv.org/abs/1706.03762

Self-attention



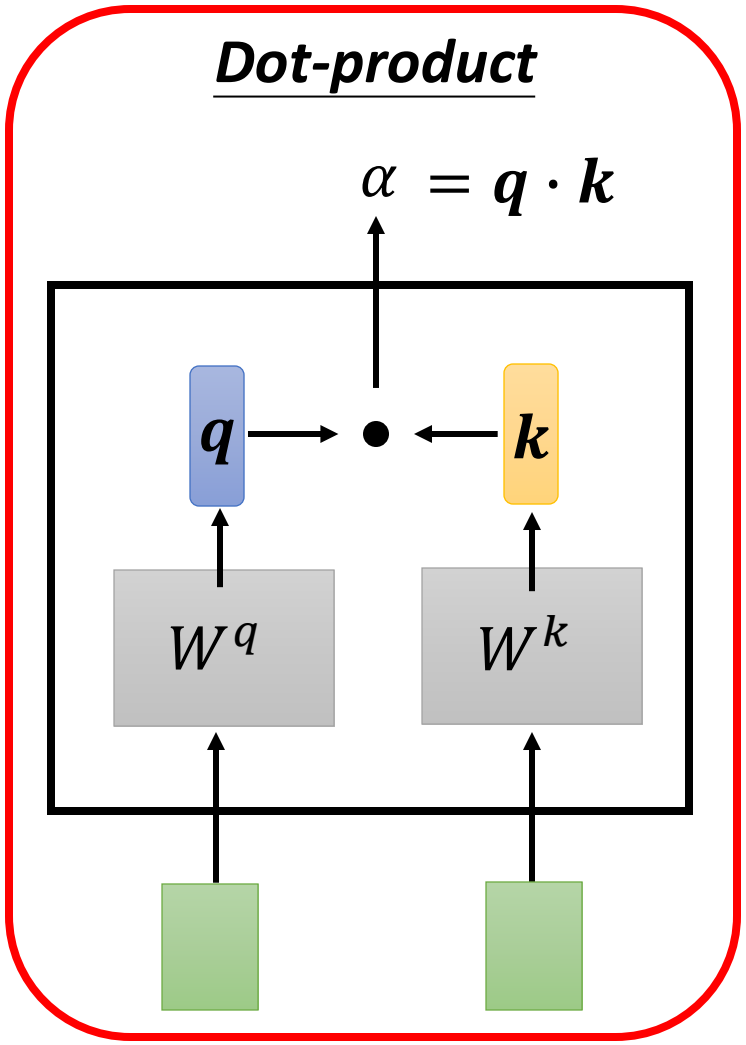
Can be either **input** or **a hidden layer**

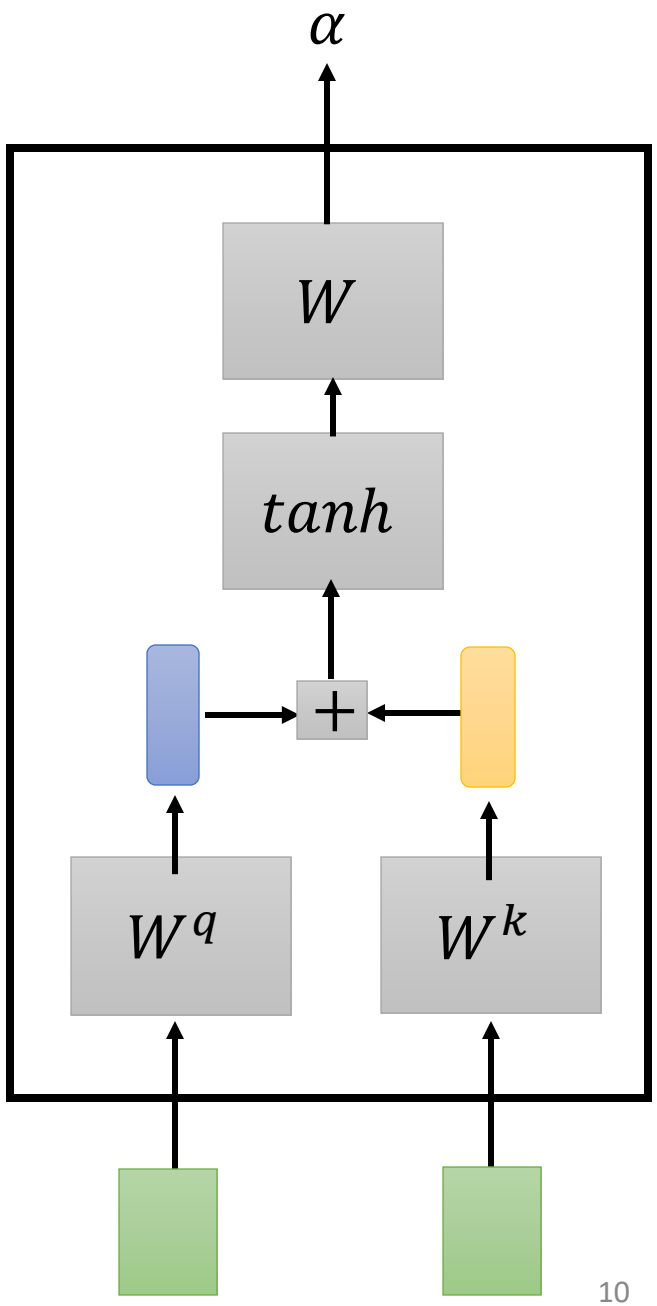
Self-attention



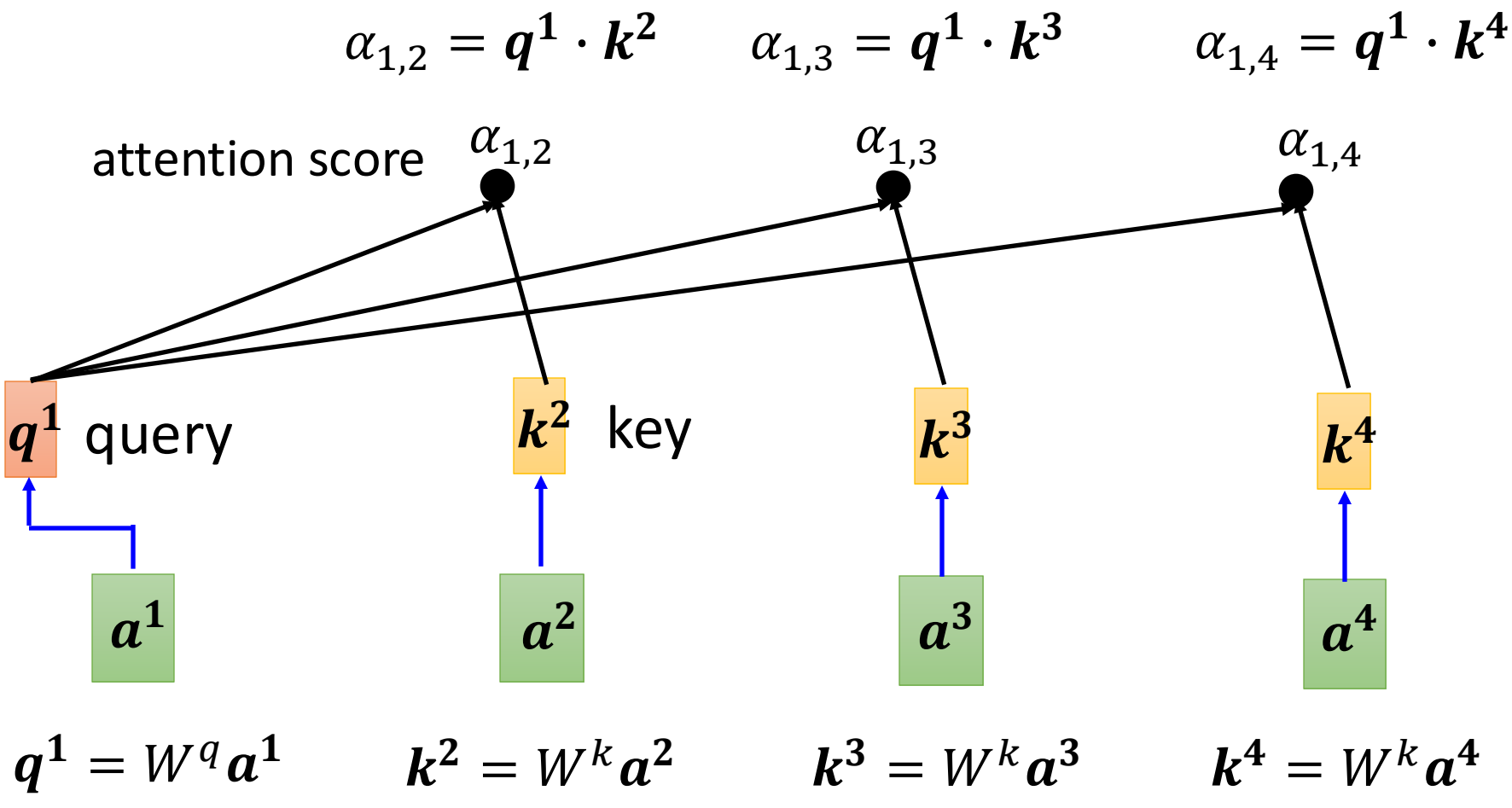
Find the relevant vectors in a sequence

Self-attention



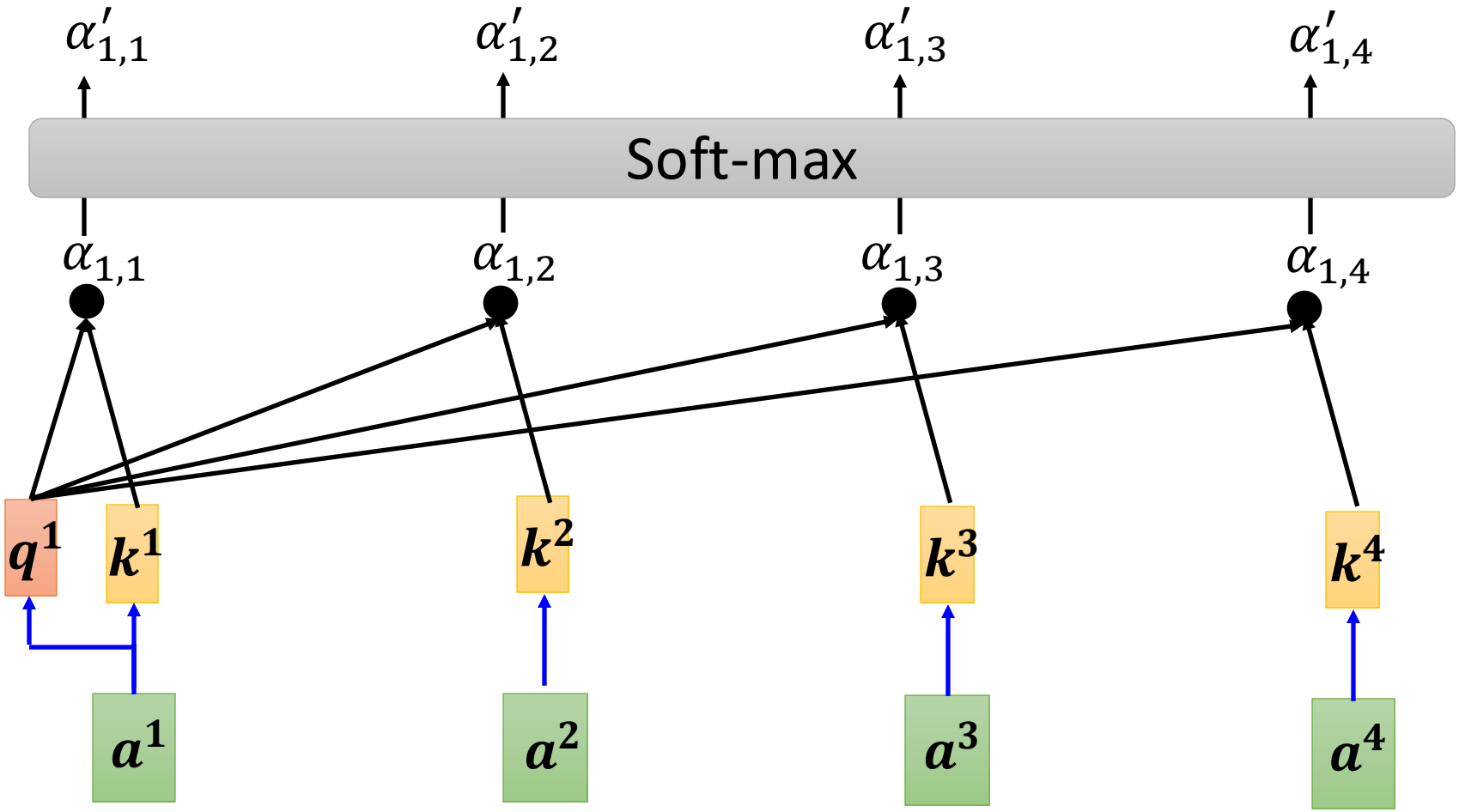


Self-attention



Self-attention

$$\alpha'_{1,i} = exp(\alpha_{1,i}) / \sum_j exp(\alpha_{1,j})$$



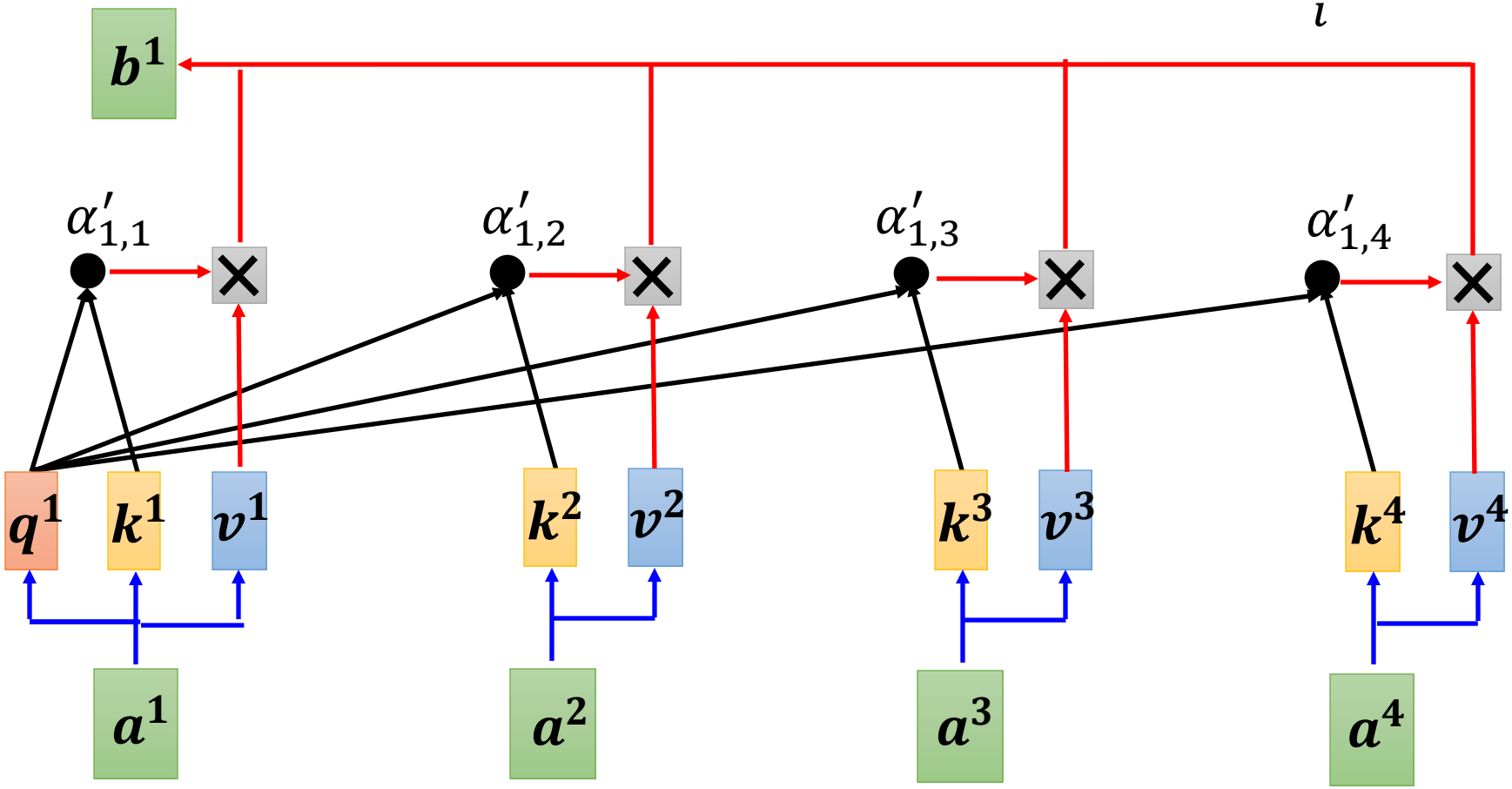
$q^1 = W^q a^1$ $k^2 = W^k a^2$ $k^3 = W^k a^3$ $k^4 = W^k a^4$

$k^1 = W^k a^1$

Self-attention

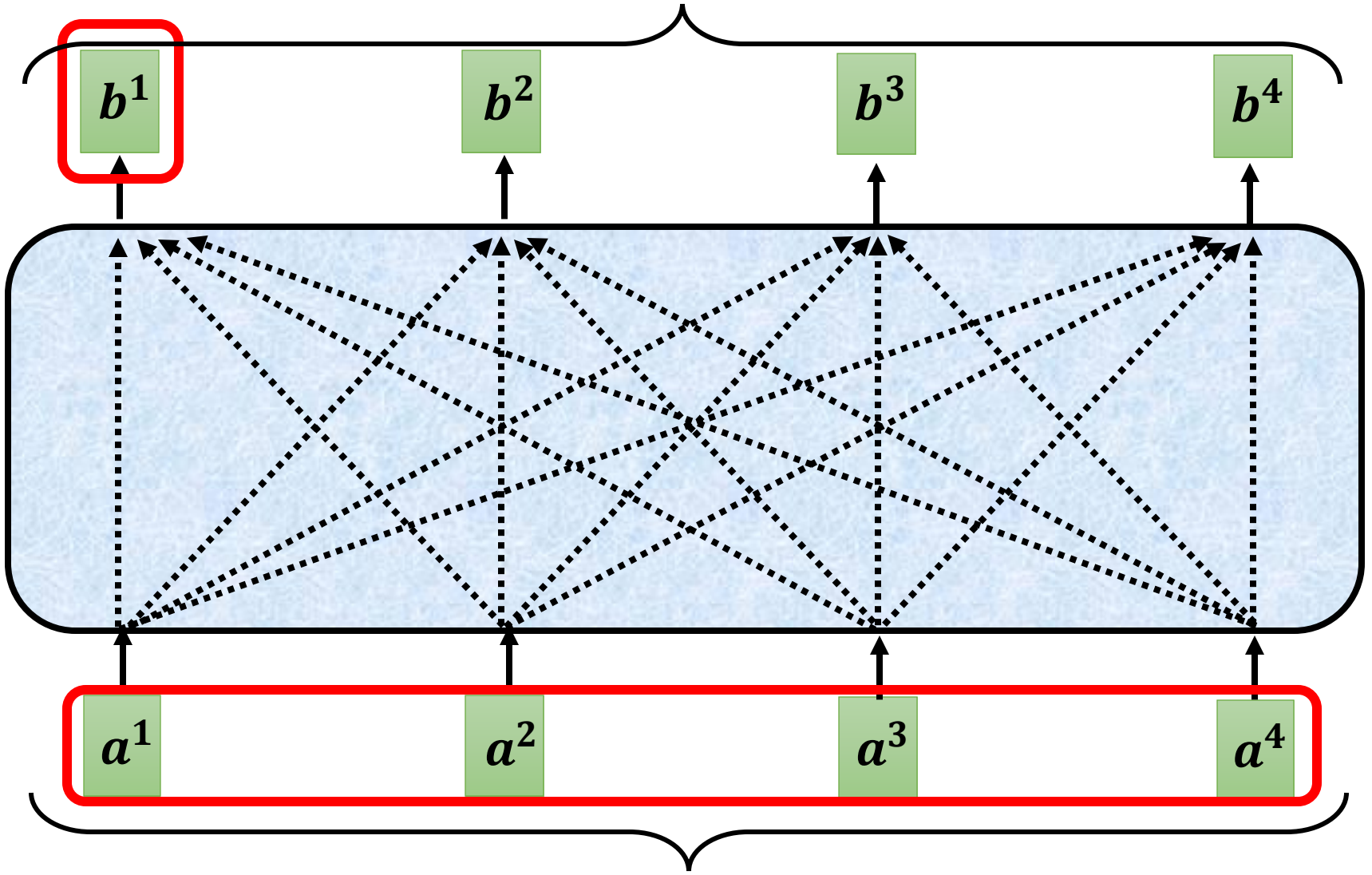
Extract information based on attention scores

$$b^1 = \sum_i \alpha'_{1,i} v^i$$



$v^1 = W^v a^1$ $v^2 = W^v a^2$ $v^3 = W^v a^3$ $v^4 = W^v a^4$

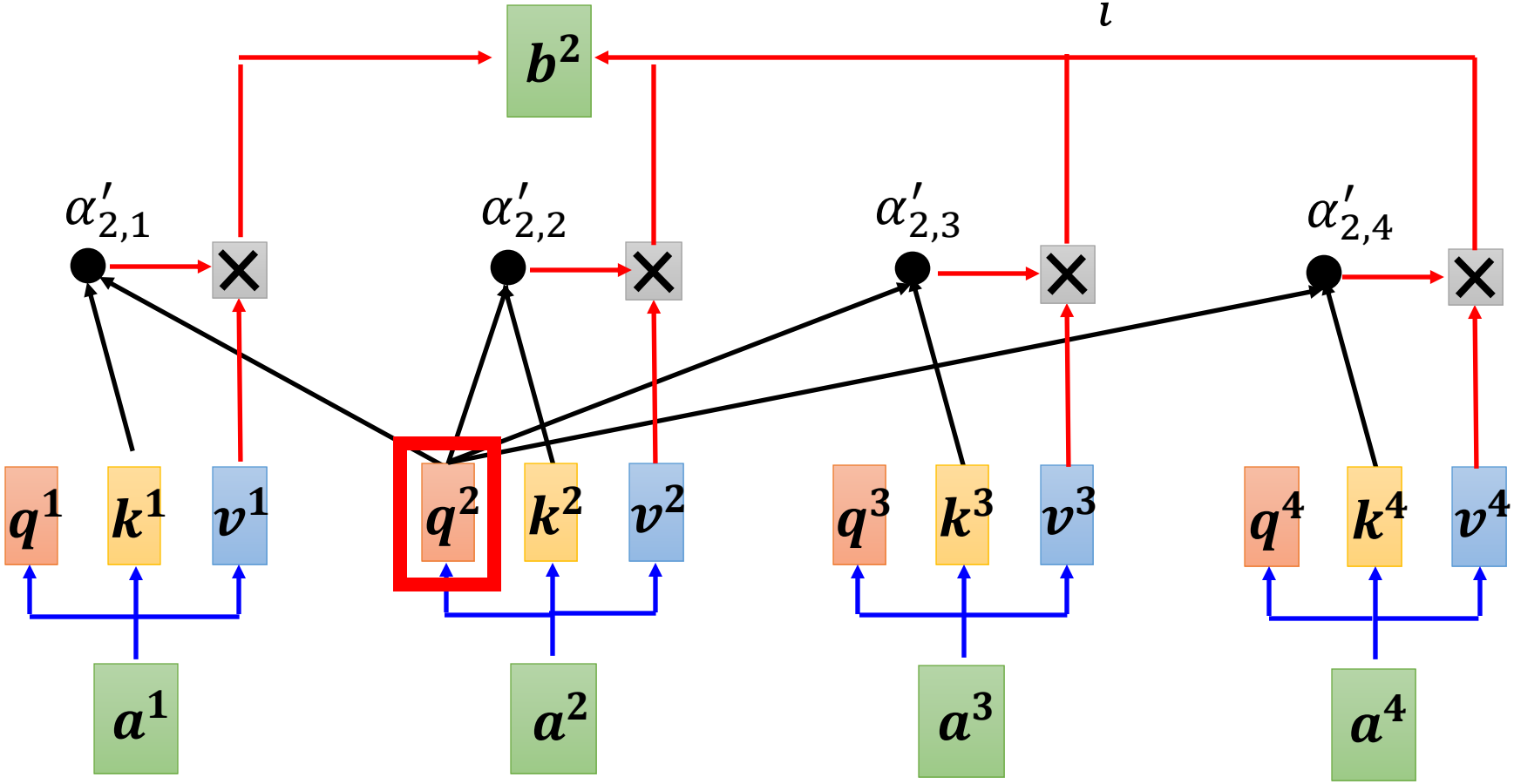
Self-attention



Can be either **input** or **a hidden layer**

Self-attention

$$b^2 = \sum_i \alpha'_{2,i} v^i$$



Self-attention

$$q^i = W^q a^i$$

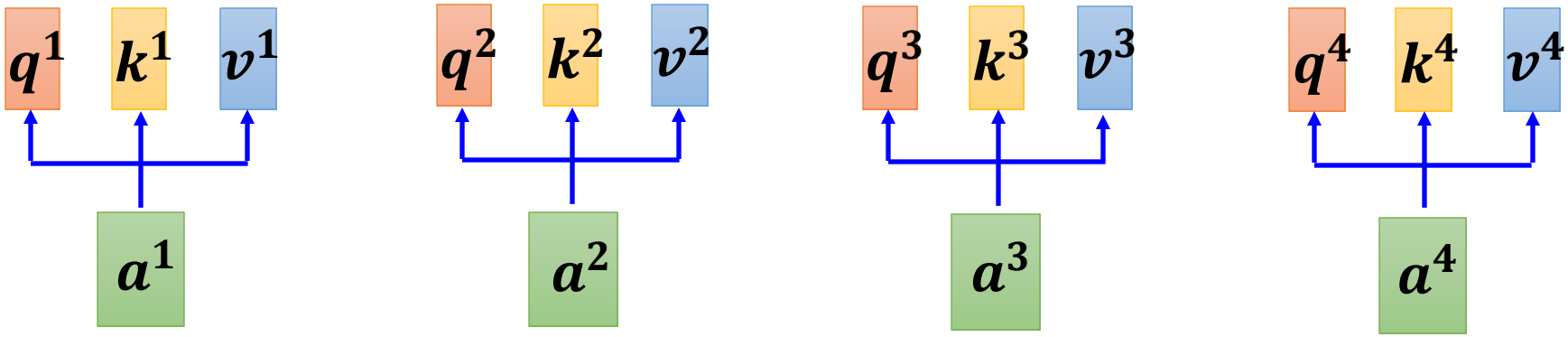
$$\begin{matrix} q^1 & q^2 & q^3 & q^4 \\ Q \end{matrix} = W^q \begin{matrix} a^1 & a^2 & a^3 & a^4 \\ I \end{matrix}$$

$$k^i = W^k a^i$$

$$\begin{matrix} k^1 & k^2 & k^3 & k^4 \\ K \end{matrix} = W^k \begin{matrix} a^1 & a^2 & a^3 & a^4 \\ I \end{matrix}$$

$$v^i = W^v a^i$$

$$\begin{matrix} v^1 & v^2 & v^3 & v^4 \\ V \end{matrix} = W^v \begin{matrix} a^1 & a^2 & a^3 & a^4 \\ I \end{matrix}$$

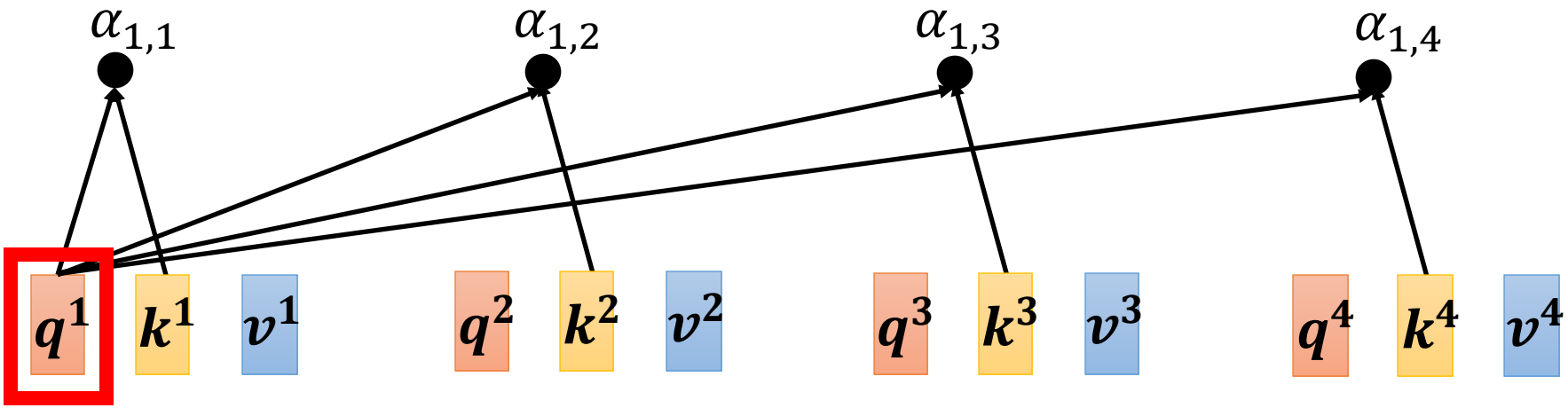


Self-attention

$\alpha_{1,1} = k^1 q^1$ $\alpha_{1,2} = k^2 q^1$

$\alpha_{1,3} = k^3 q^1$ $\alpha_{1,4} = k^4 q^1$

$$\begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{bmatrix} = \begin{bmatrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{bmatrix} q^1$$

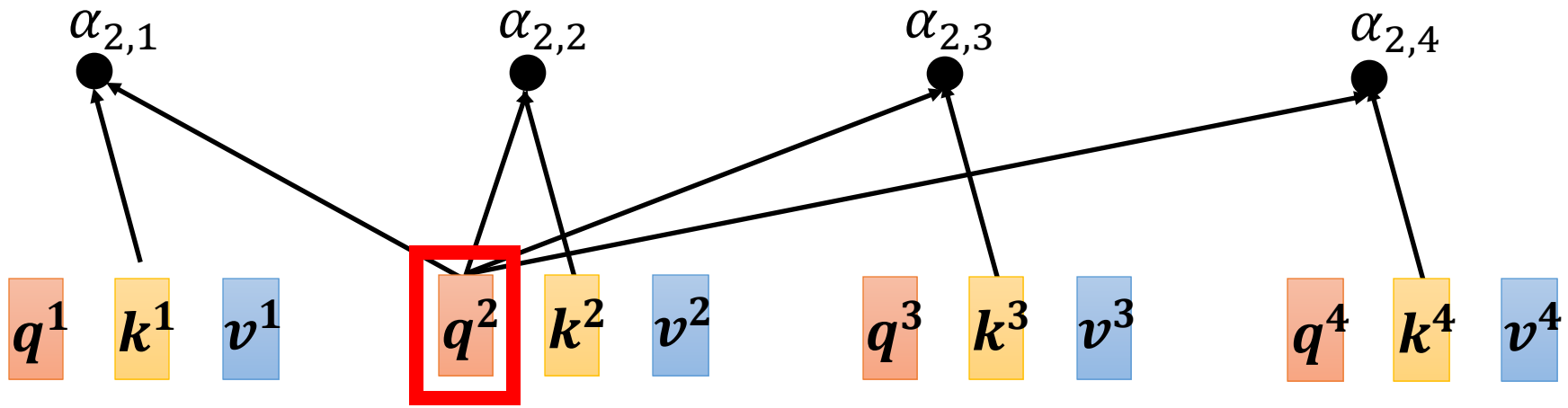


Self-attention

$\alpha_{1,1} = \boldsymbol{k^1}\ \boldsymbol{q^1}$ $\alpha_{1,2} = \boldsymbol{k^2}\ \boldsymbol{q^1}$

$\alpha_{1,3} = \boldsymbol{k^3}\ \boldsymbol{q^1}$ $\alpha_{1,4} = \boldsymbol{k^4}\ \boldsymbol{q^1}$

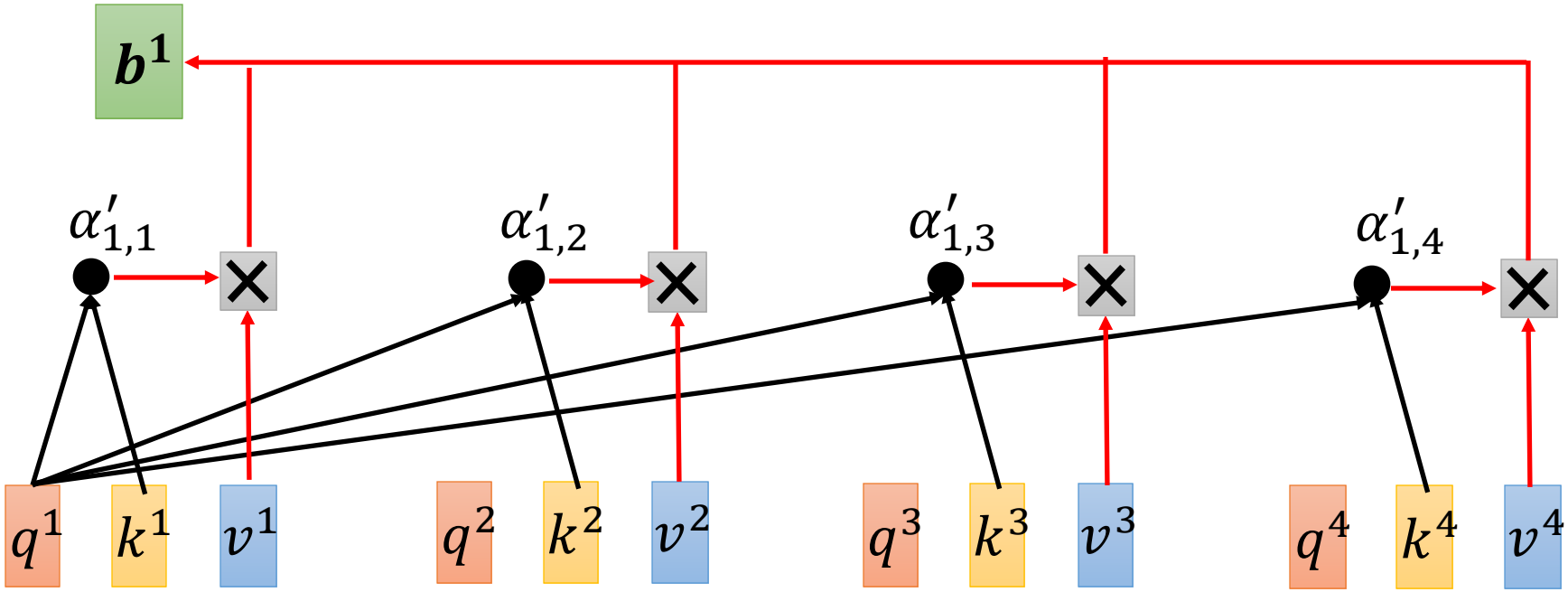
$$\begin{bmatrix}\alpha_{1,1}\\ \alpha_{1,2}\\ \alpha_{1,3}\\ \alpha_{1,4}\end{bmatrix} = \begin{bmatrix}\boldsymbol{k^1}\\ \boldsymbol{k^2}\\ \boldsymbol{k^3}\\ \boldsymbol{k^4}\end{bmatrix} \boldsymbol{q^1}$$



$$\begin{bmatrix}\alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1}\\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2}\\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3}\\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4}\end{bmatrix} = A' \quad \text{softmax} \quad \begin{bmatrix}\alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \alpha_{4,1}\\ \alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \alpha_{4,2}\\ \alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \alpha_{4,3}\\ \alpha_{1,4} & \alpha_{2,4} & \alpha_{3,4} & \alpha_{4,4}\end{bmatrix} = A$$

$$\begin{bmatrix}\boldsymbol{k^1}\\ \boldsymbol{k^2}\\ \boldsymbol{k^3}\\ \boldsymbol{k^4}\end{bmatrix} = K^T \qquad \begin{bmatrix}\boldsymbol{q^1} & \boldsymbol{q^2} & \boldsymbol{q^3} & \boldsymbol{q^4}\end{bmatrix} = Q$$

Self-attention

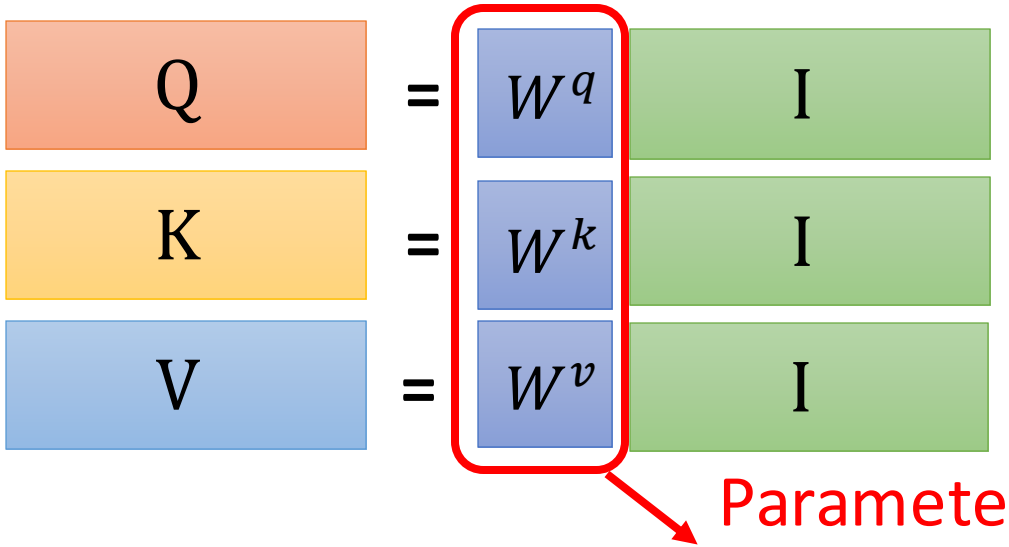


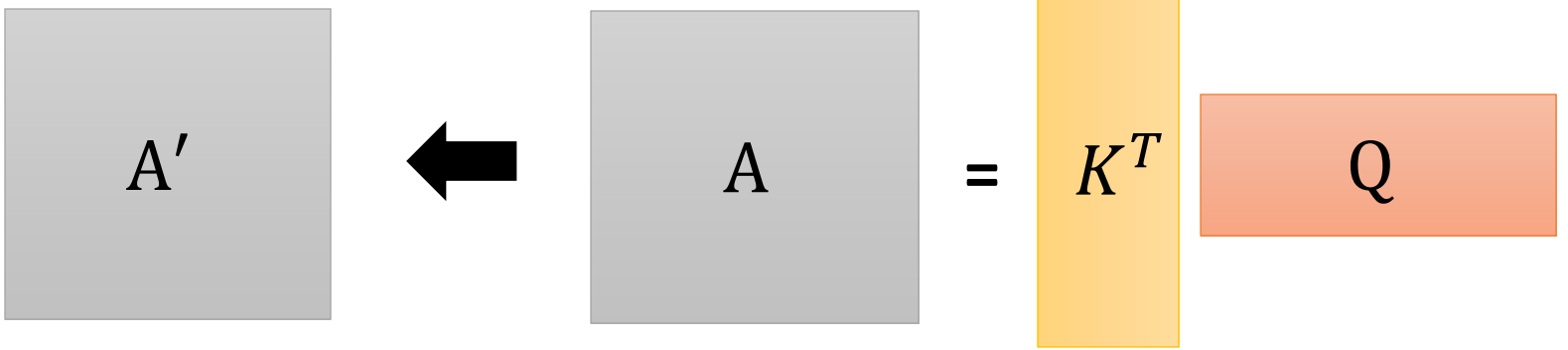
$b^1\ b^2\ b^3\ b^4$ = $v^1\ v^2\ v^3\ v^4$

O V

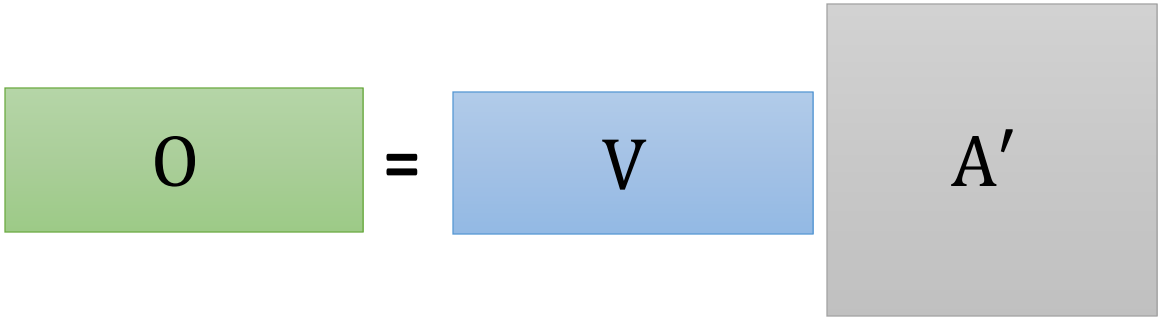
$$A' = \begin{bmatrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{bmatrix}$$

Self-attention





Attention Matrix



Many applications …

Transformer

https://arxiv.org/abs/1706.03762

BERT

https://arxiv.org/abs/1810.04805

Widely used in Natural Langue Processing (NLP)!

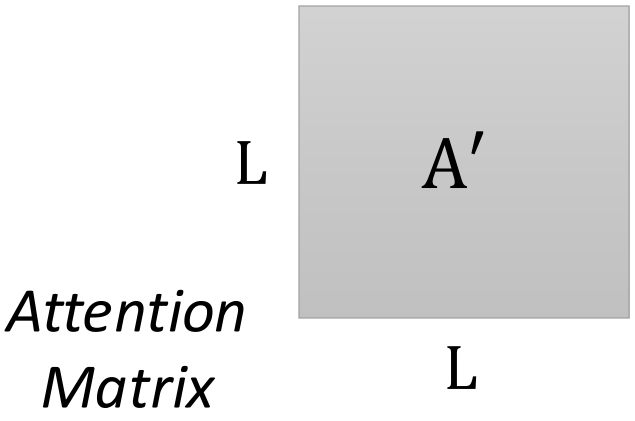
https://arxiv.org/abs/1910.12977

Self-attention for Speech

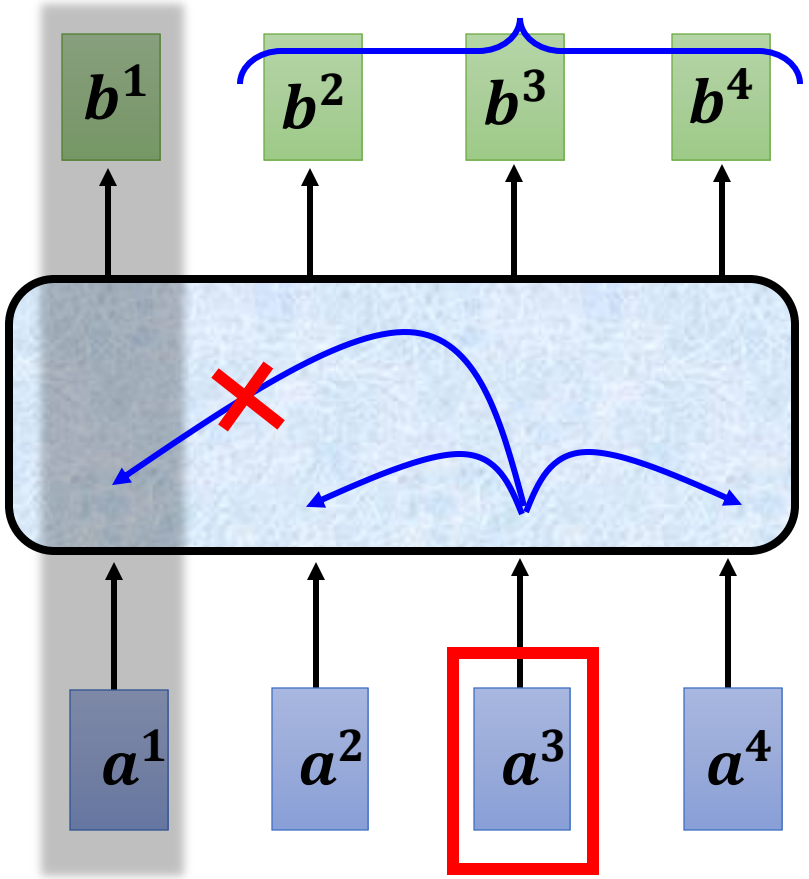
Speech is a very long vector sequence.

10ms

If input sequence is length L



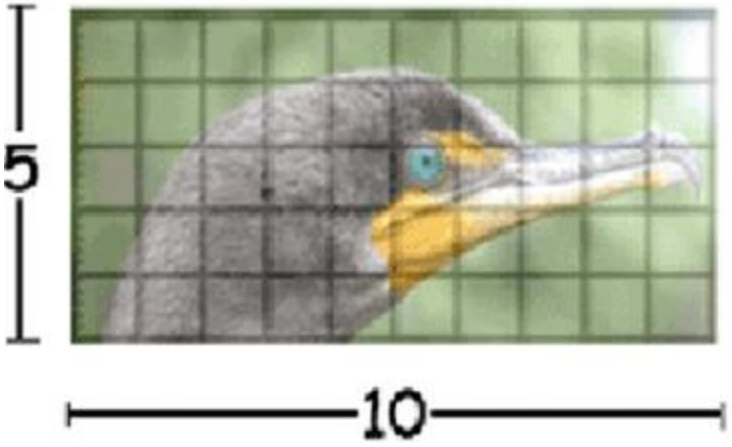
Attention in a range



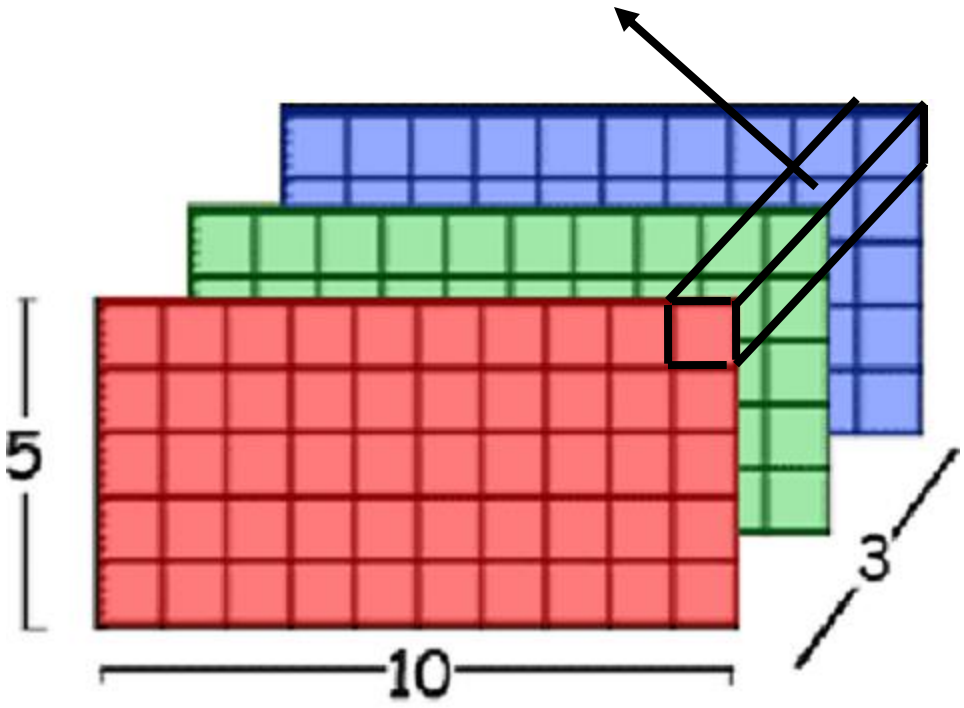
Truncated Self-attention

Self-attention for Image

An **image** can also be considered as a **vector set**.

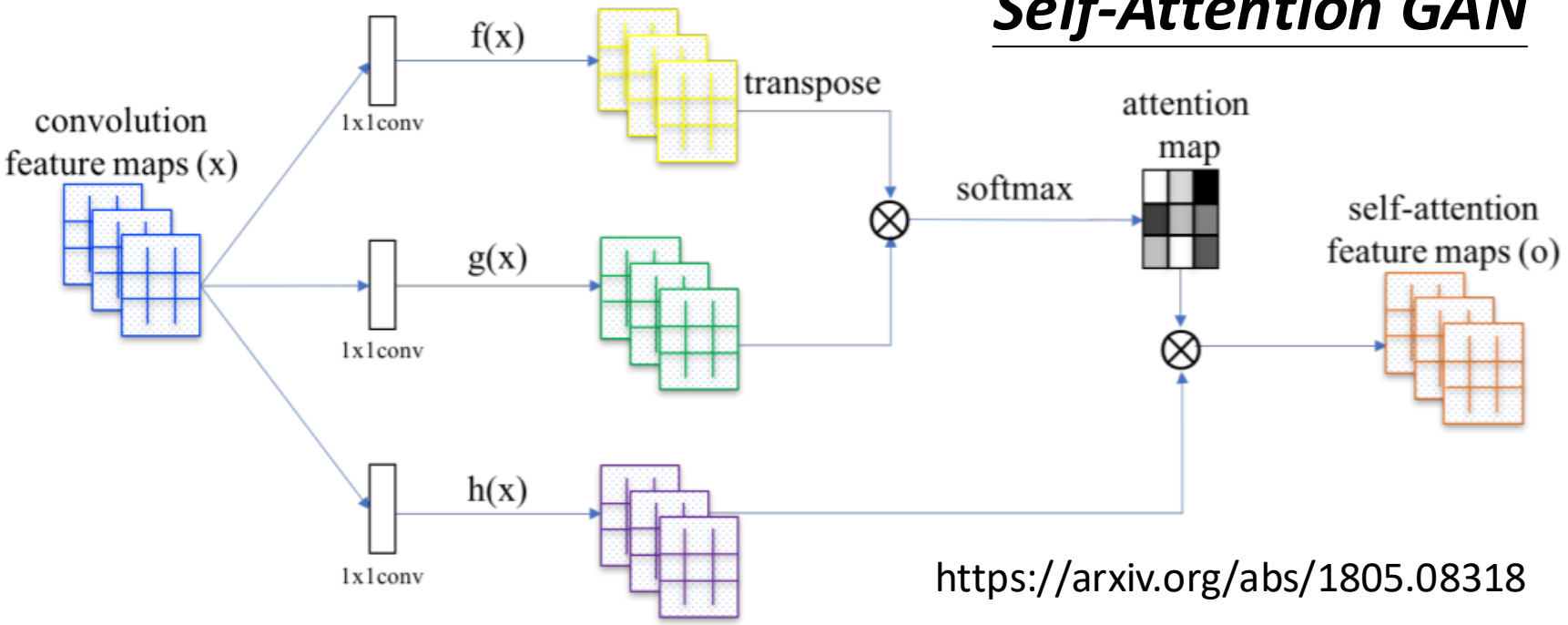


This is a vector.



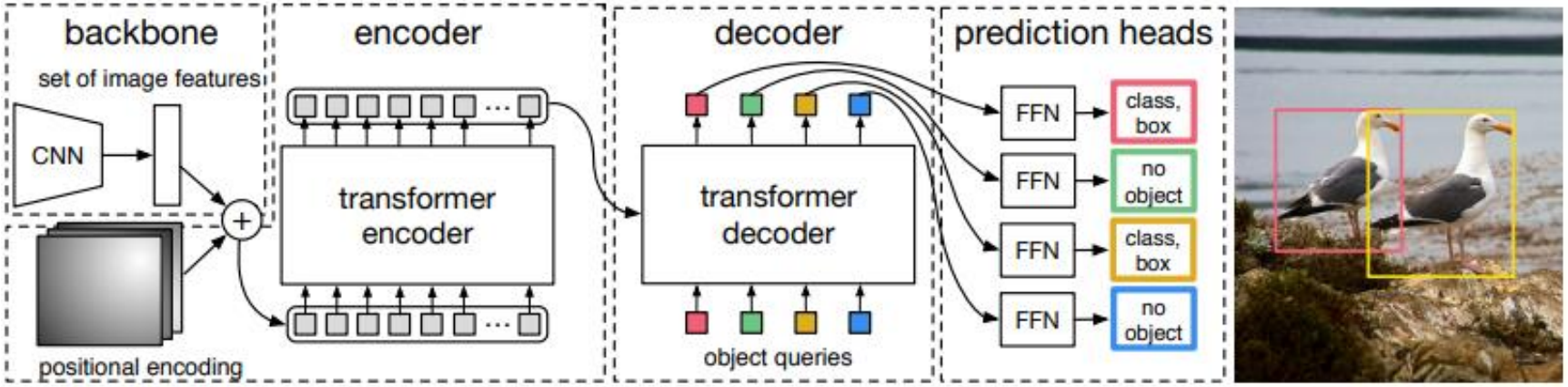
Source of image: https://www.researchgate.net/figure/Color-image-representation-and-RGB-matrix_fig15_282798184

Self-Attention GAN



https://arxiv.org/abs/1805.08318

DEtection Transformer (DETR)

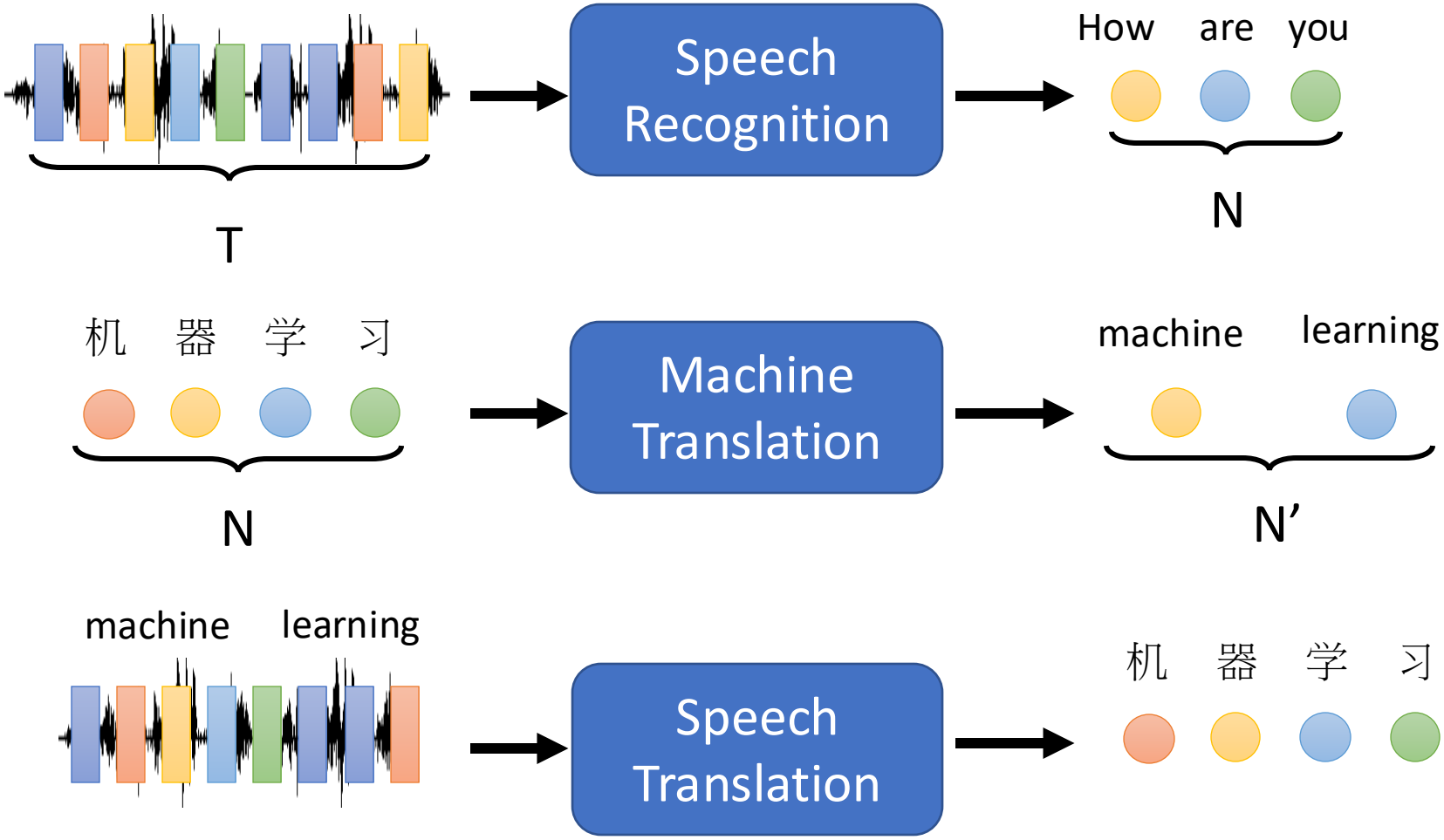


https://arxiv.org/abs/2005.12872

Sequence-to-sequence (Seq2seq)

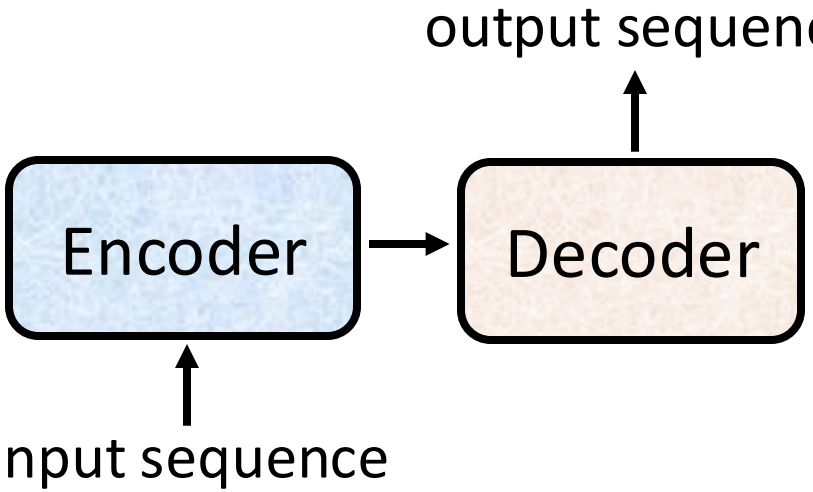
Input a sequence, output a sequence

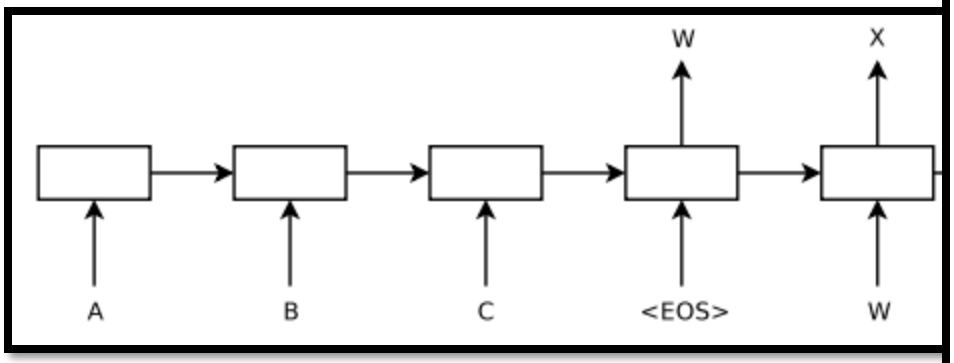
The output length is determined by model.



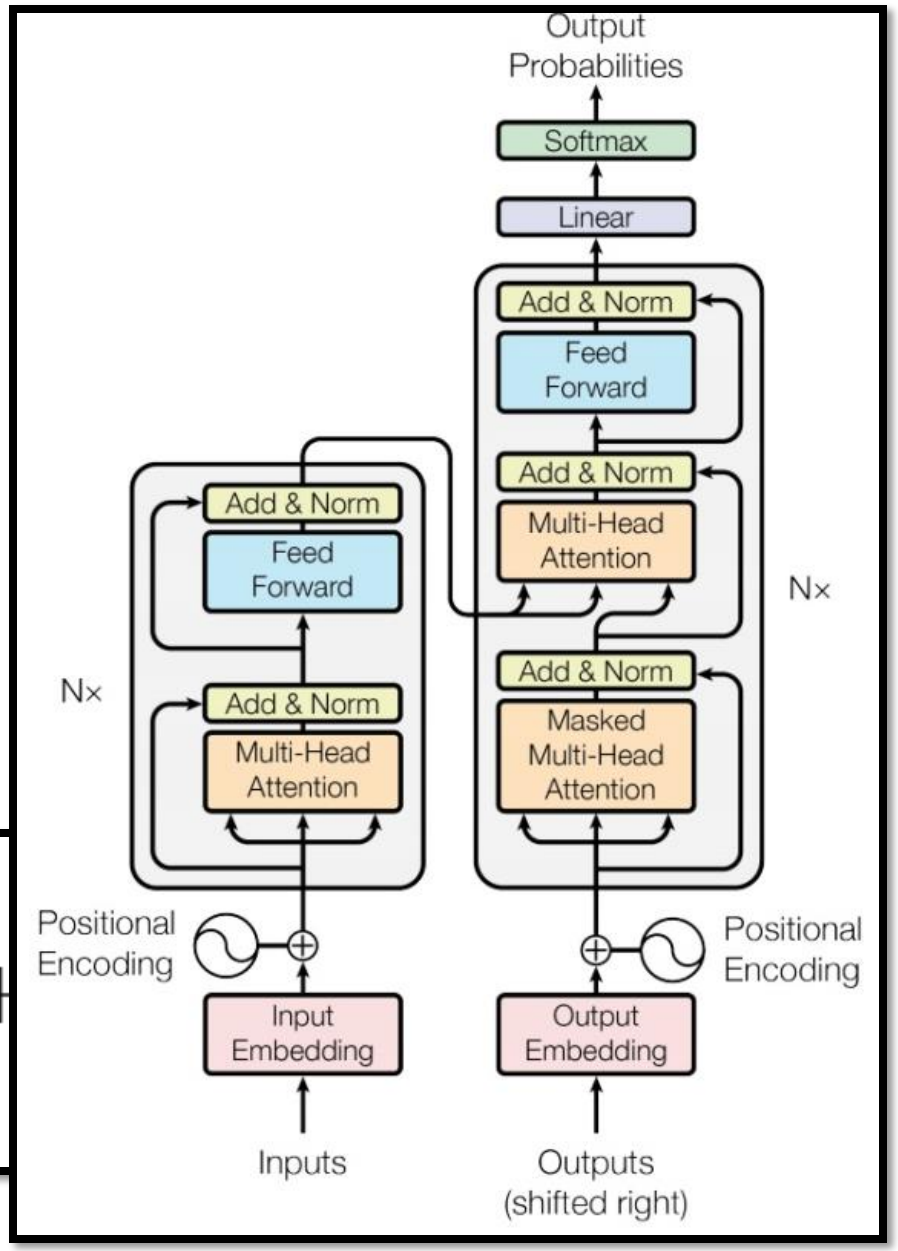
Language without text

Seq2seq



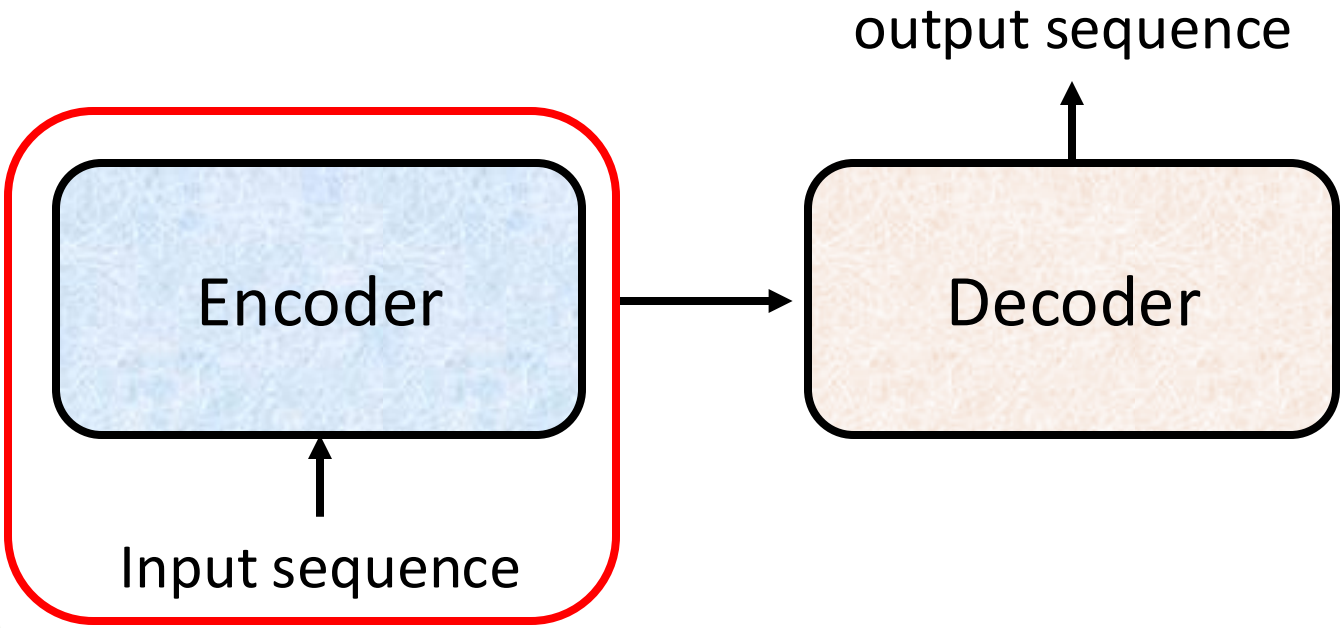


Sequence to Sequence Learning with Neural Networks
https://arxiv.org/abs/1409.3215



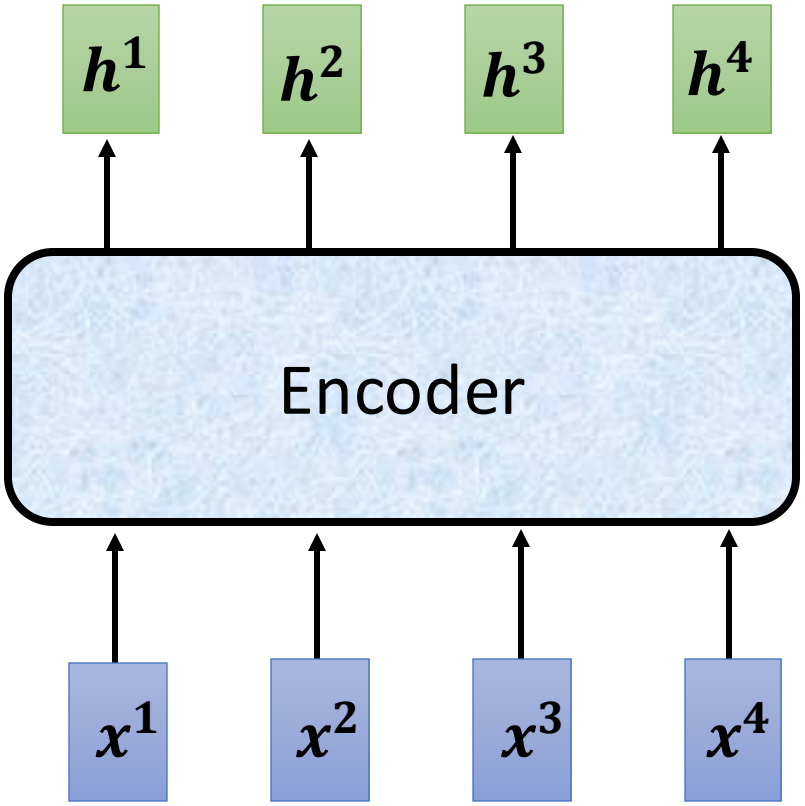
Transformer
https://arxiv.org/abs/1706.03762

Encoder

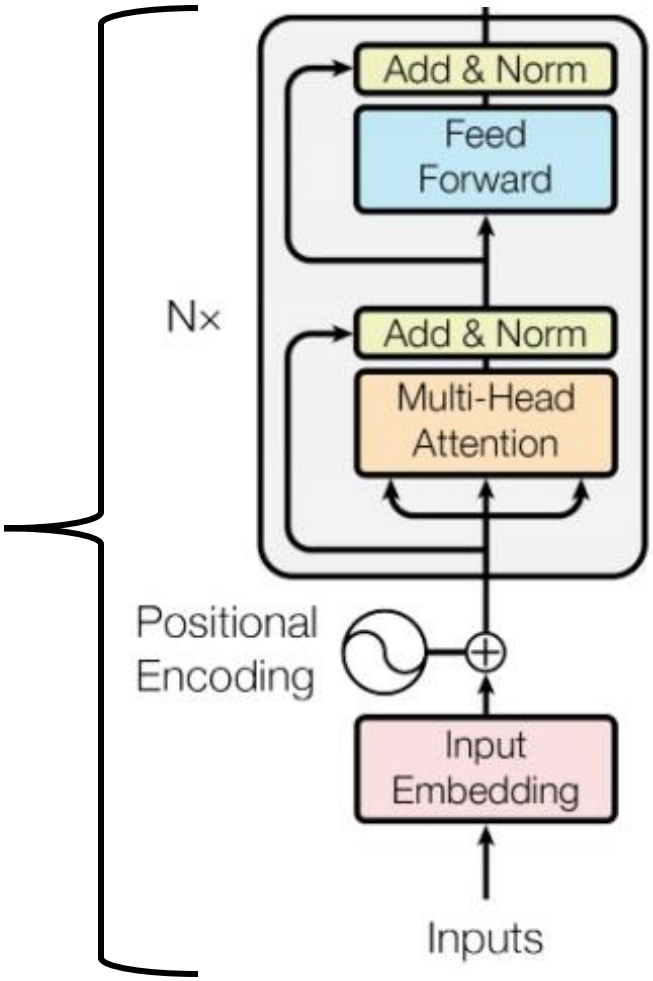


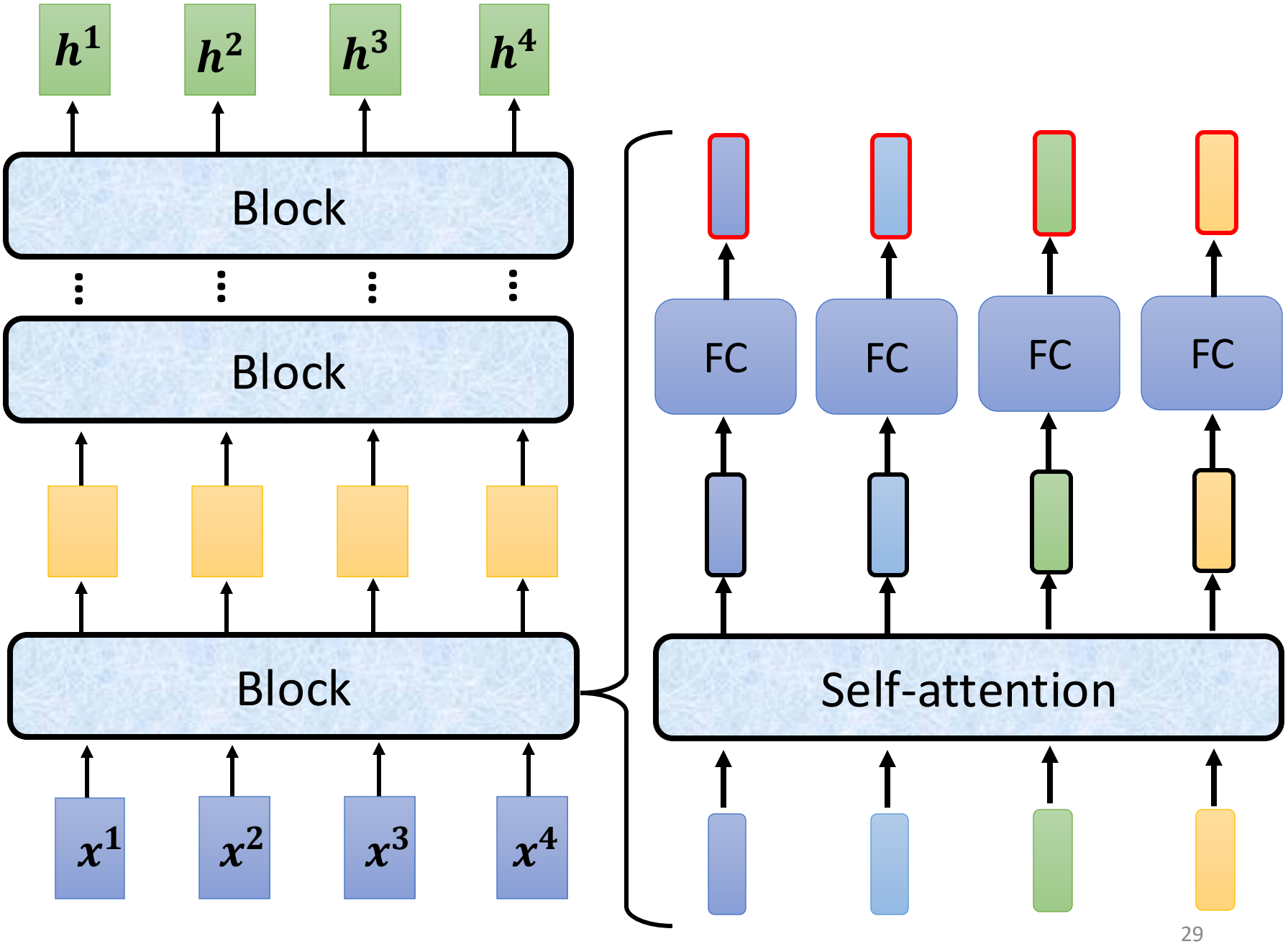
Encoder

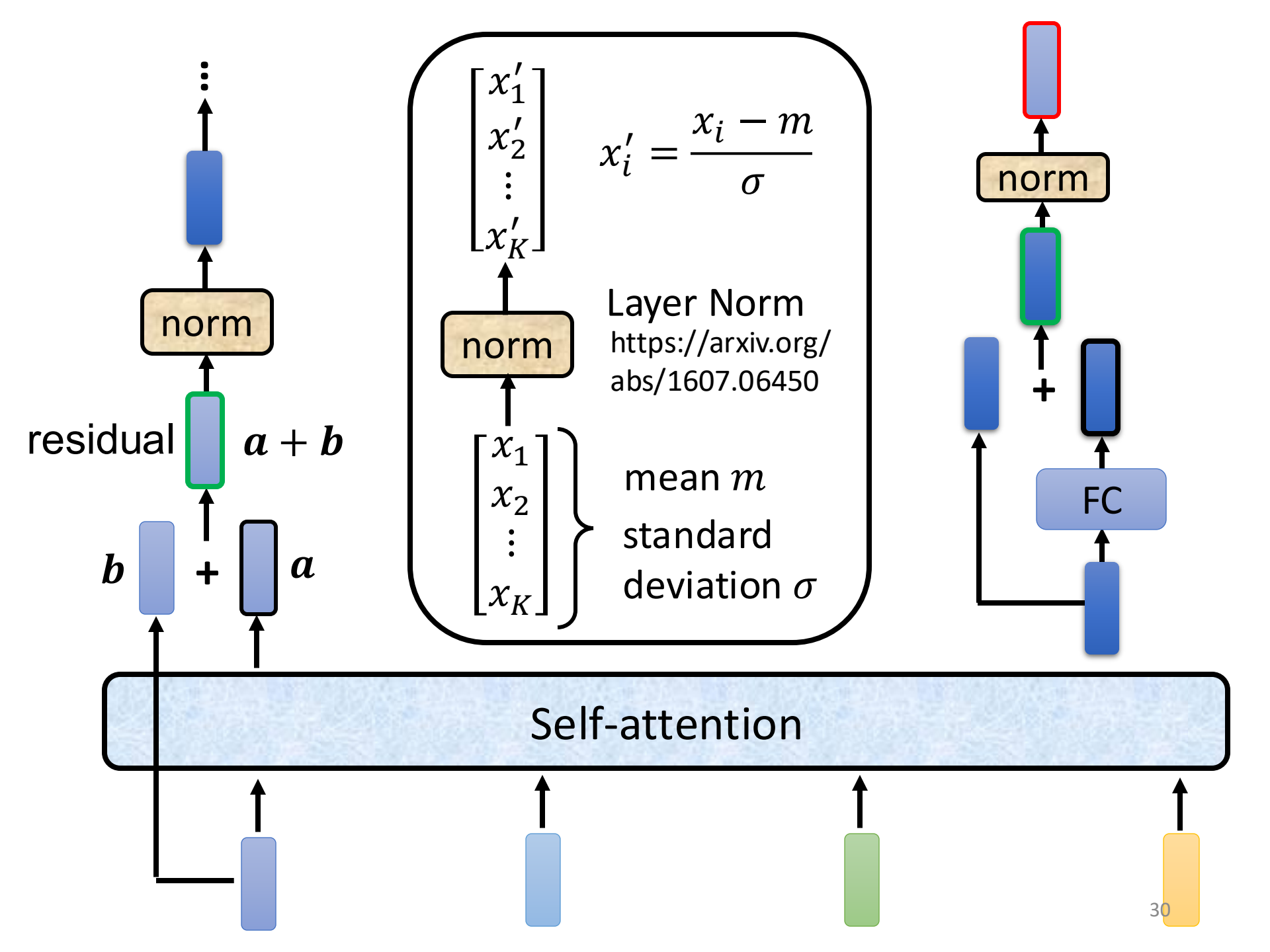
You can use **RNN** or **CNN**.

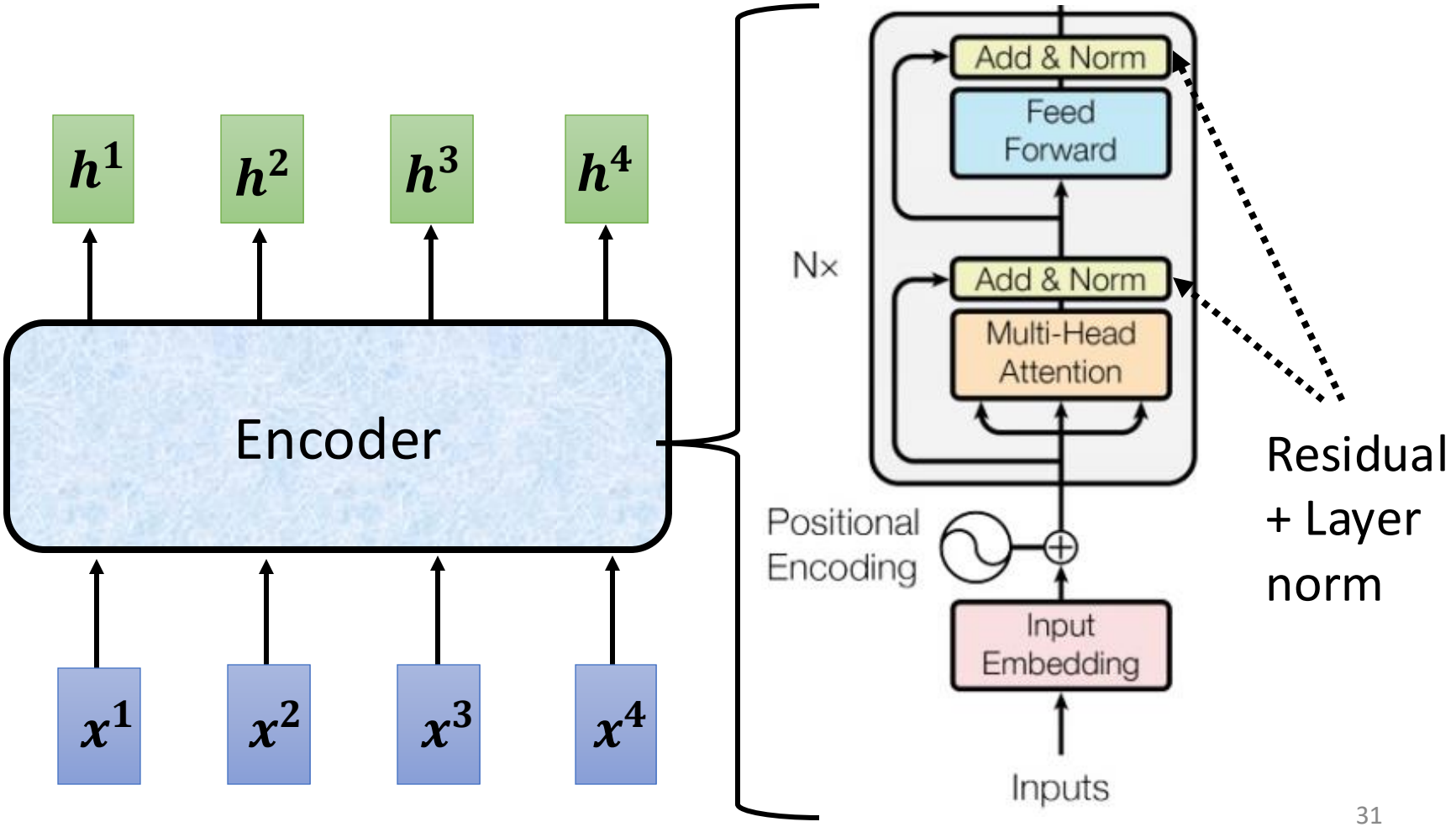


Transformer's Encoder




h^1
h^2
h^3
h^4
Block
Block
Block
x^1
x^2
x^3
x^4
FC
FC
FC
FC
Self-attention

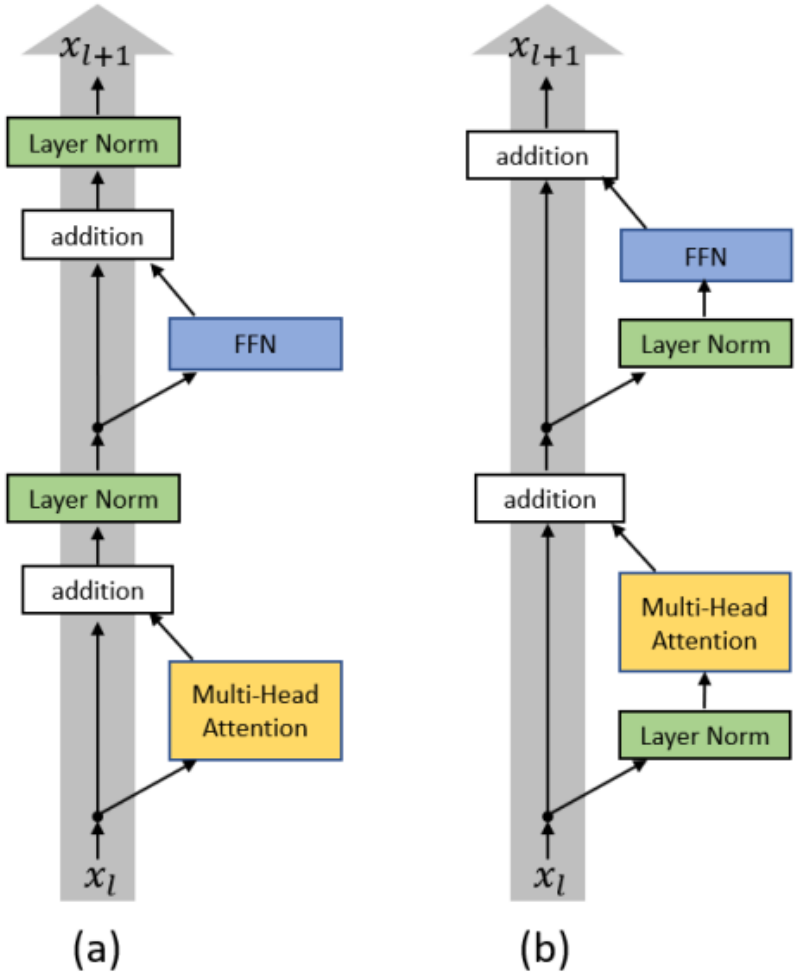

norm
residual
$a + b$
b
+
a
$\begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_K \end{bmatrix}$
$x'_i = \frac{x_i - m}{\sigma}$
norm
Layer Norm
https://arxiv.org/abs/1607.06450
$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix}$
mean m
standard deviation σ
norm
+
FC
Self-attention


h^1
h^2
h^3
h^4
Encoder
x^1
x^2
x^3
x^4
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Positional Encoding
Input Embedding
Inputs
Residual + Layer norm

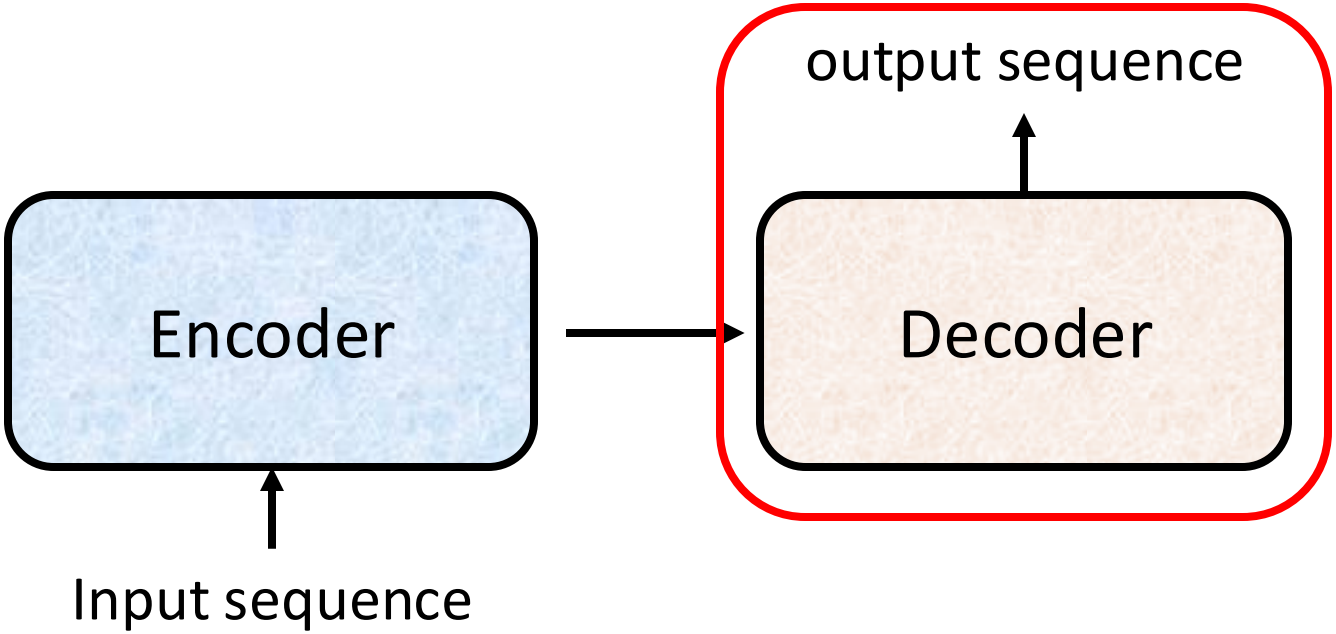
To learn more ……

- On Layer Normalization in the Transformer Architecture
- https://arxiv.org/abs/2002.04745

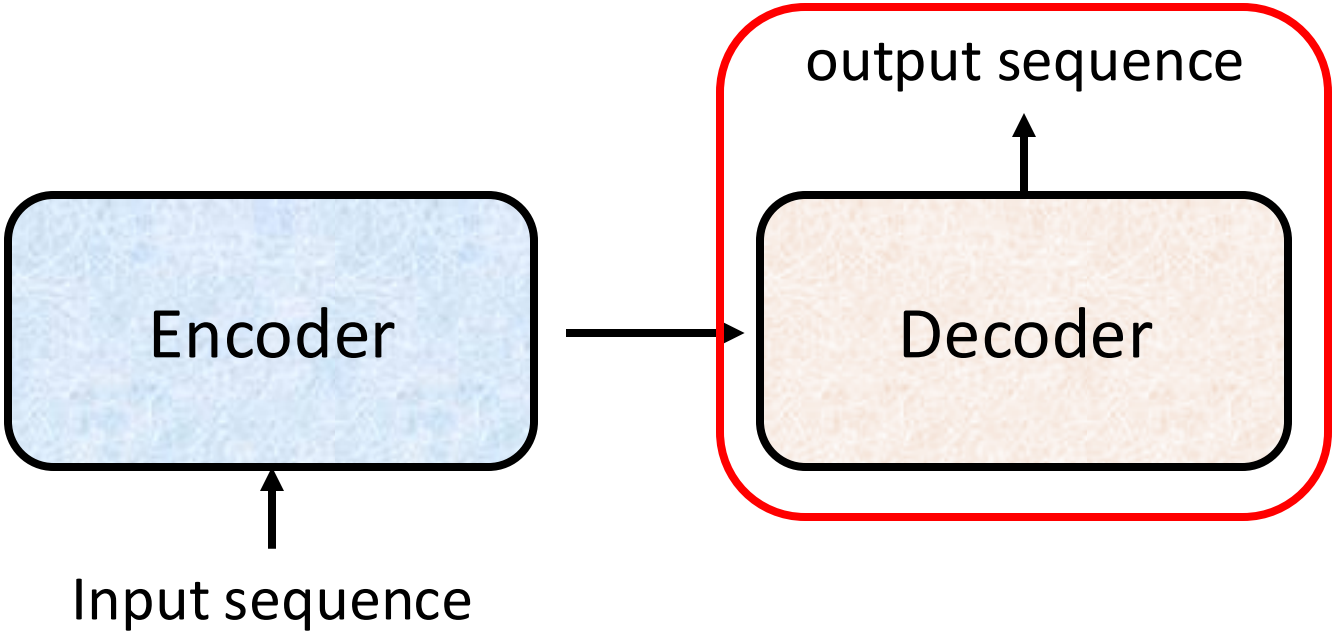
- PowerNorm: Rethinking Batch Normalization in Transformers
- https://arxiv.org/abs/2003.07845



Decoder

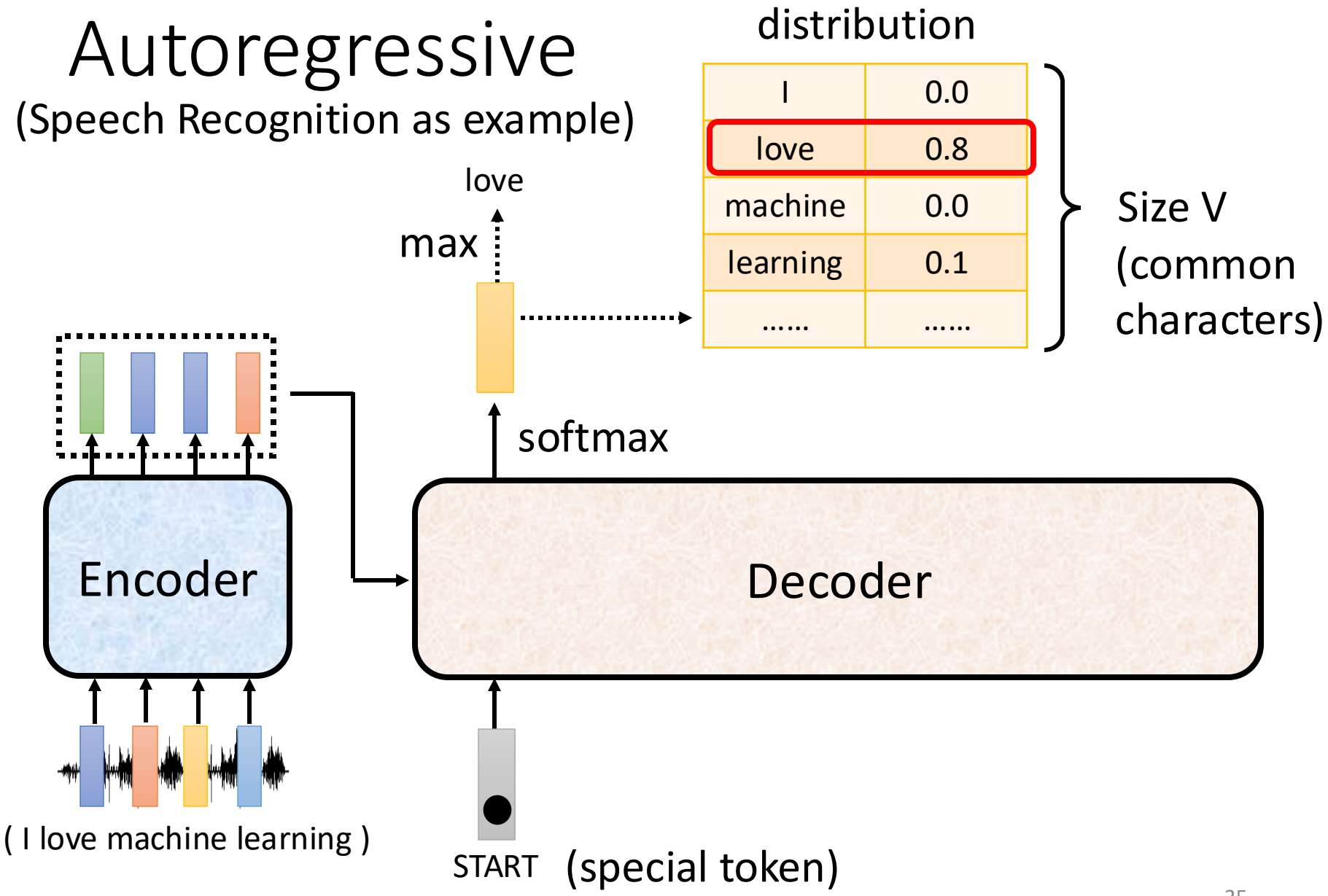


Decoder – Autoregressive (AT)

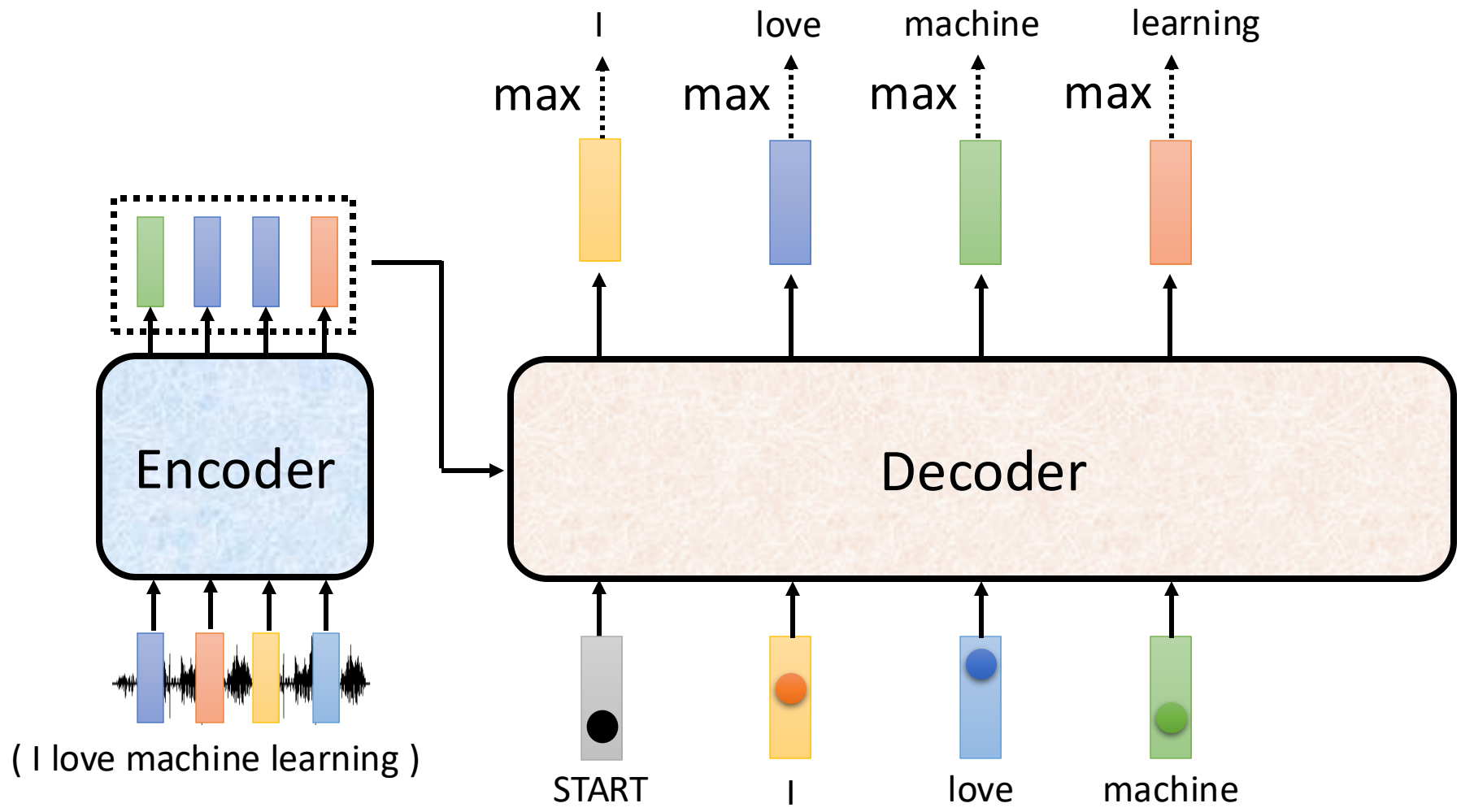


Autoregressive

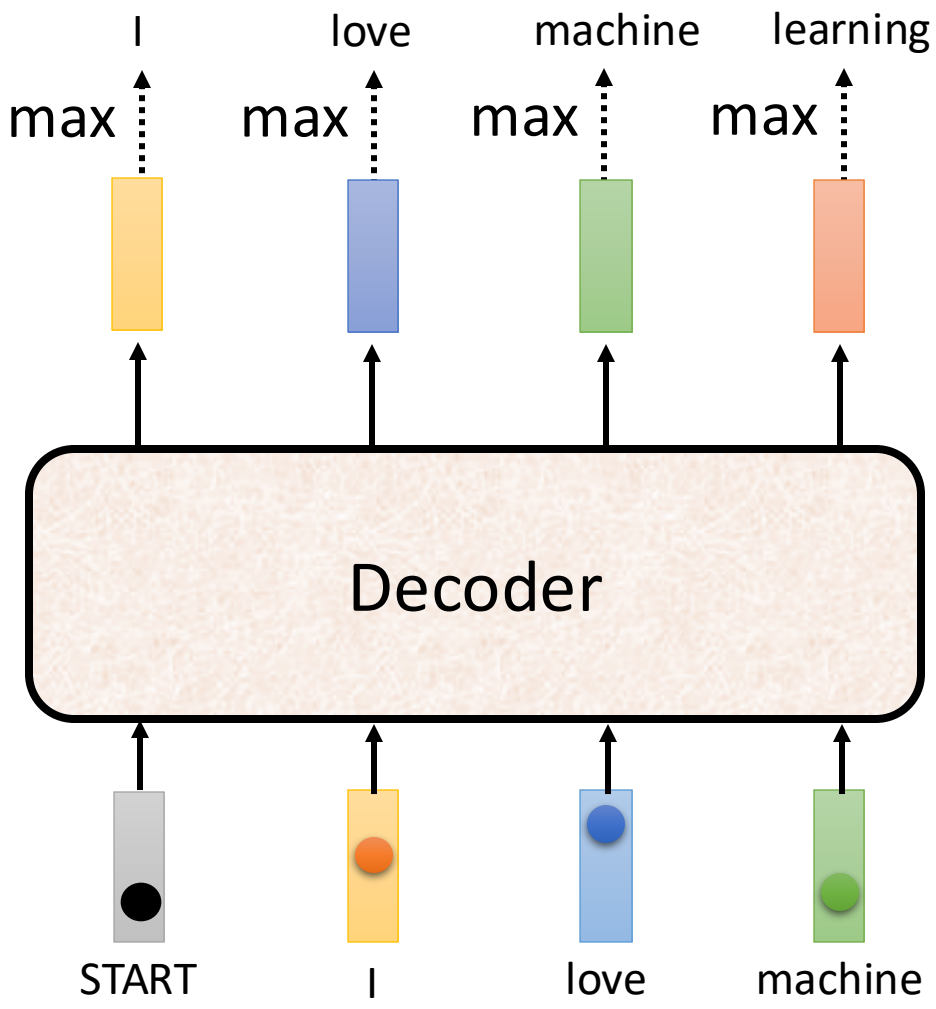
(Speech Recognition as example)

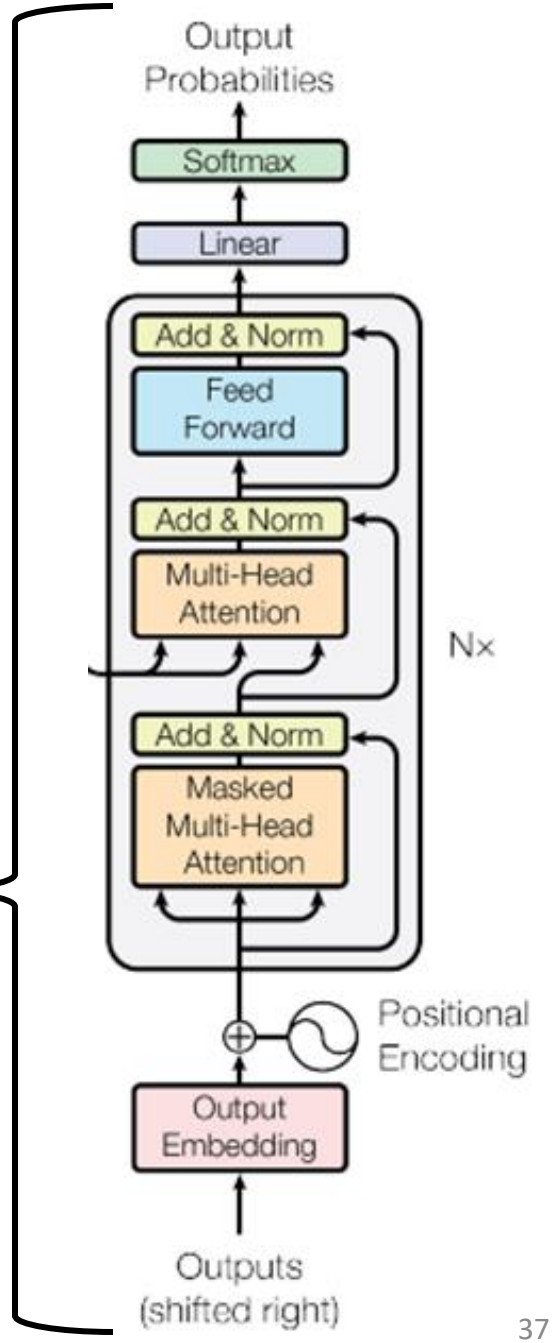


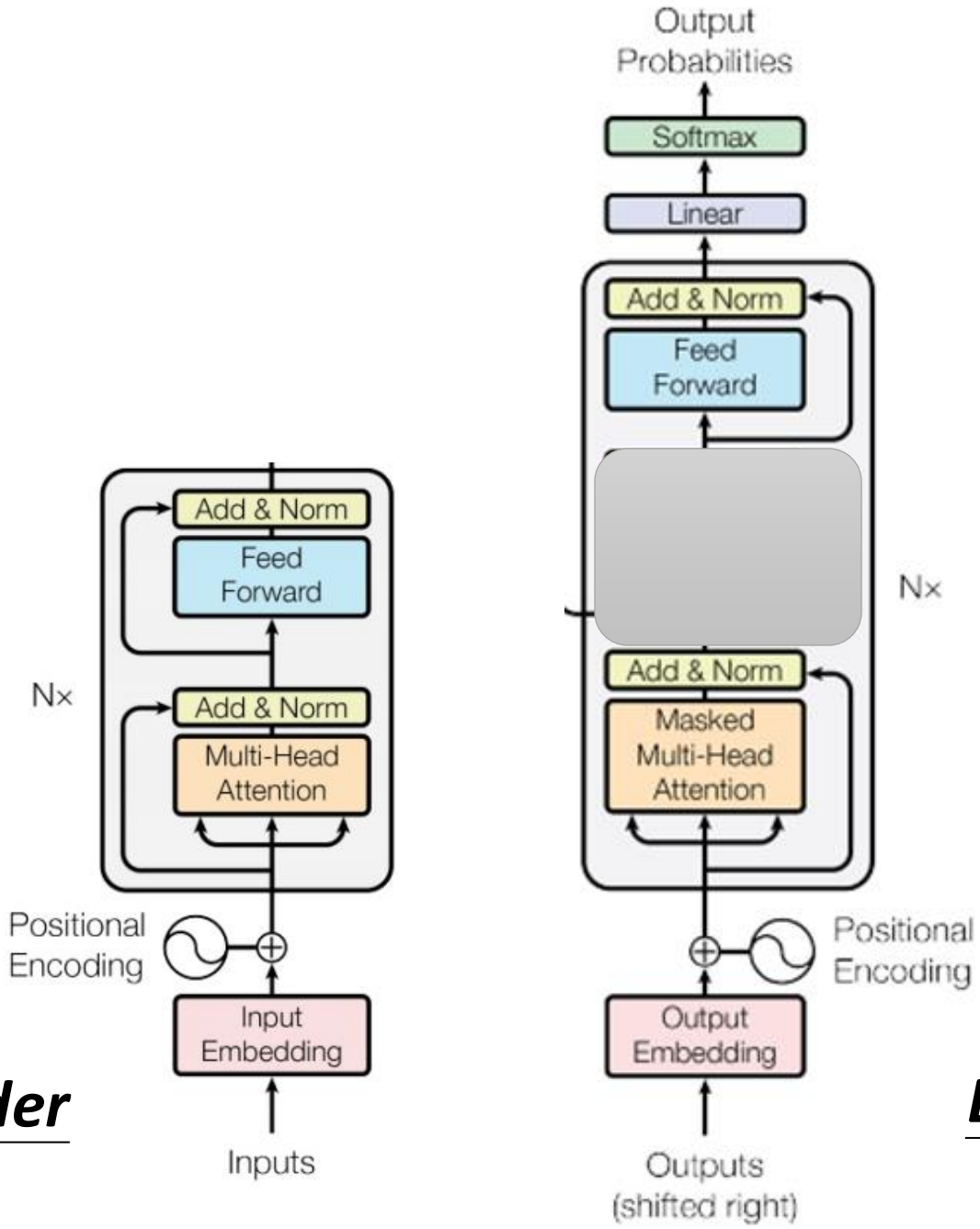
Autoregressive



ignore the input from the encoder here ☺



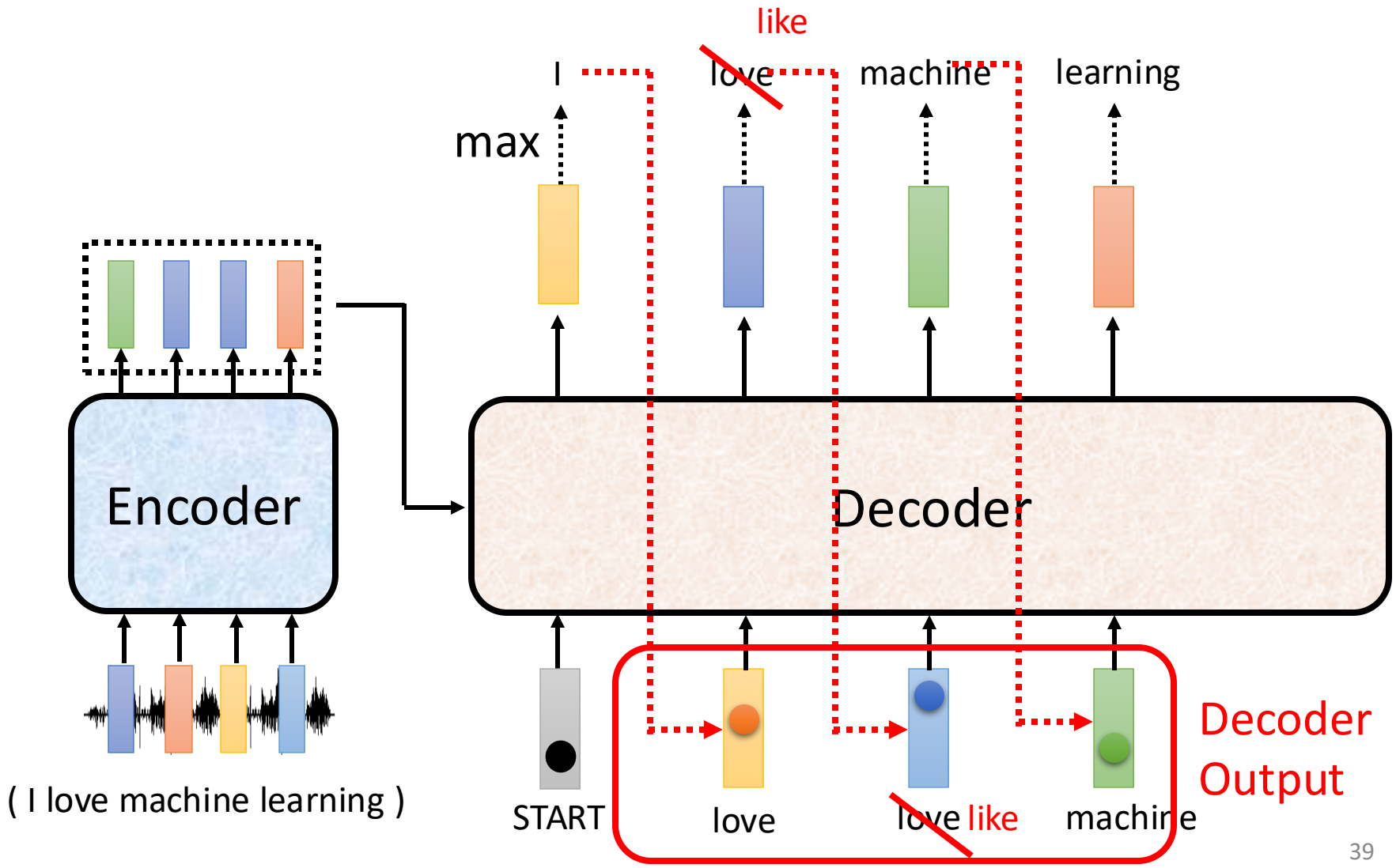




Encoder

Decoder

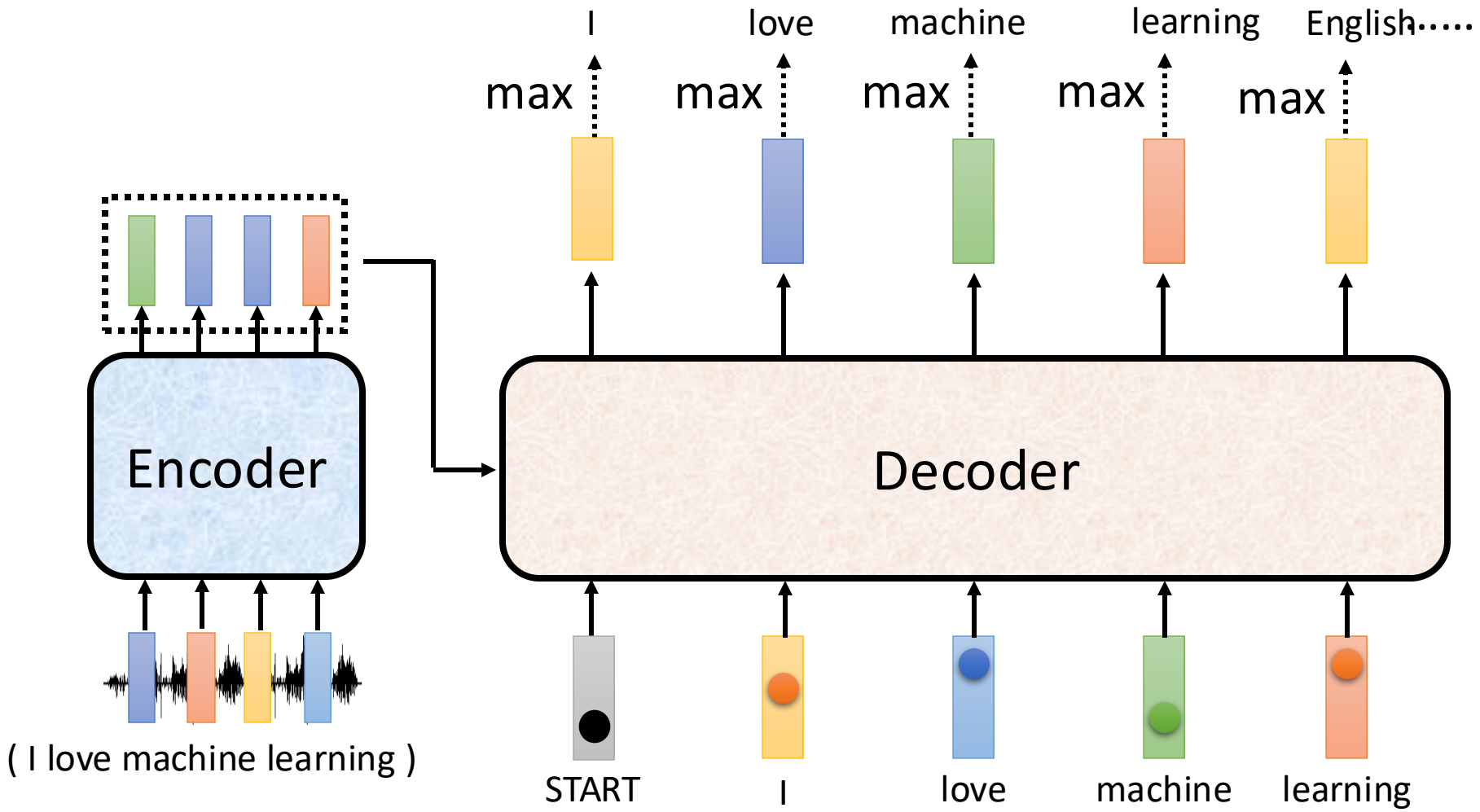
Autoregressive



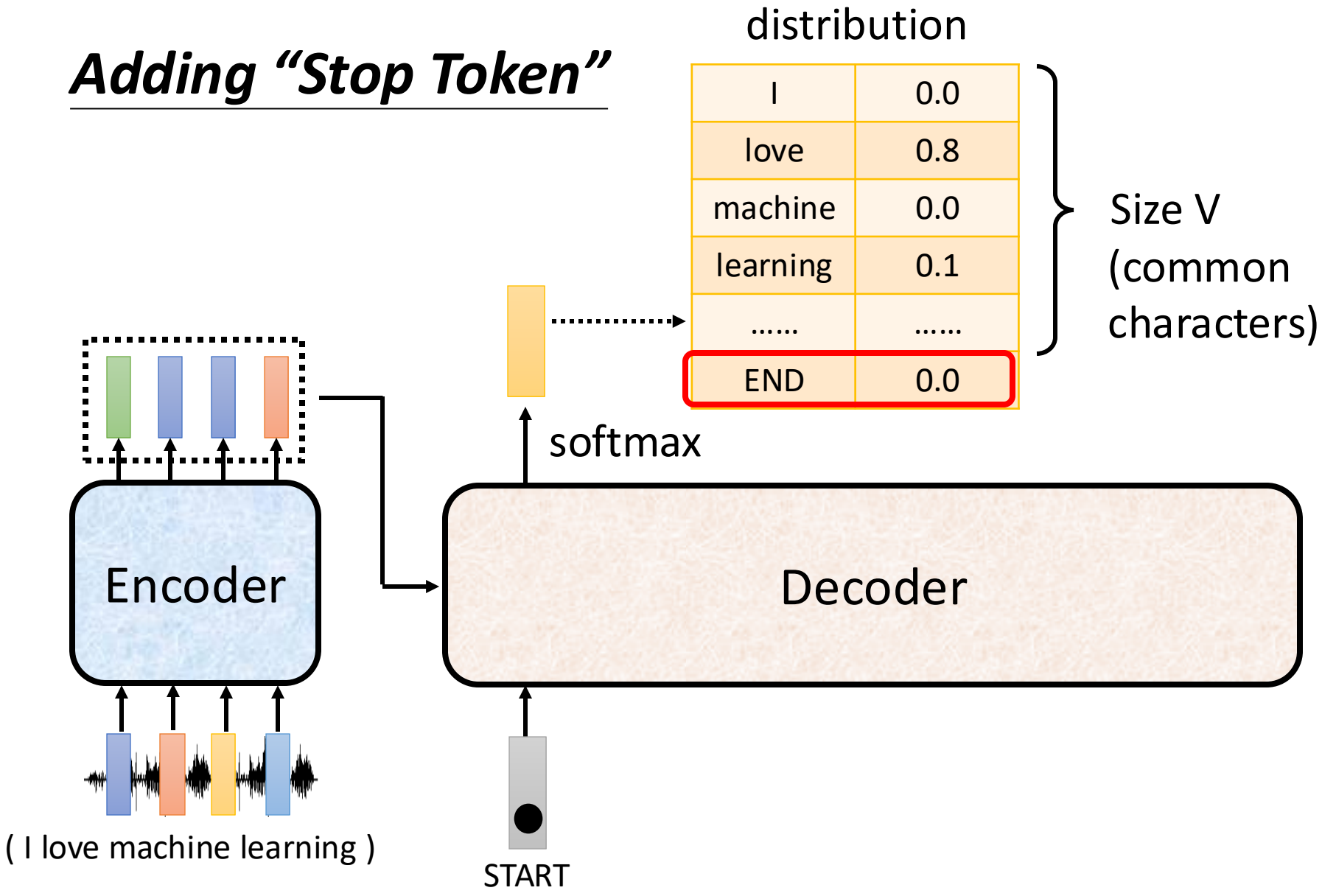
Autoregressive

We do not know the correct output length.

Never stop!

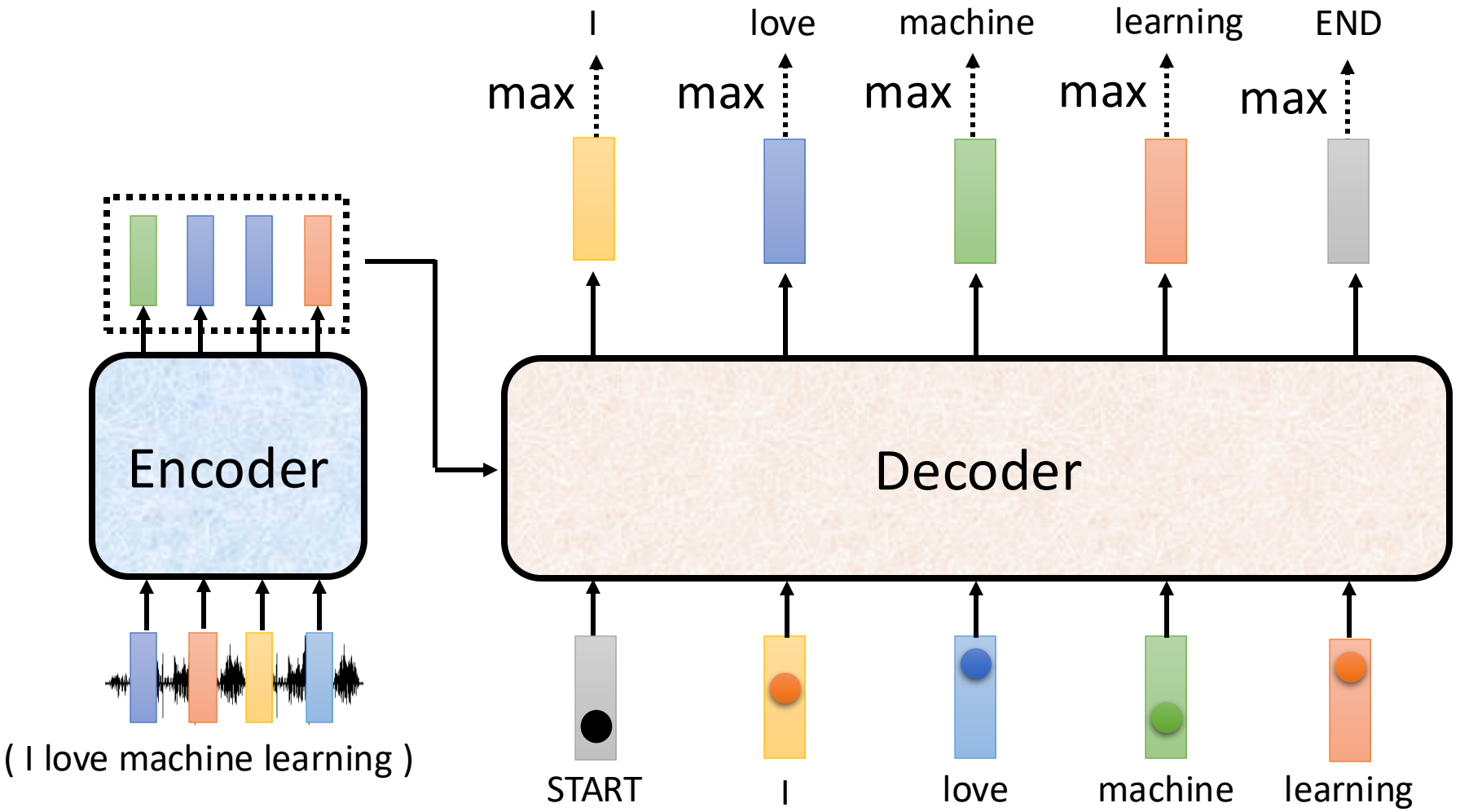


Adding "Stop Token"

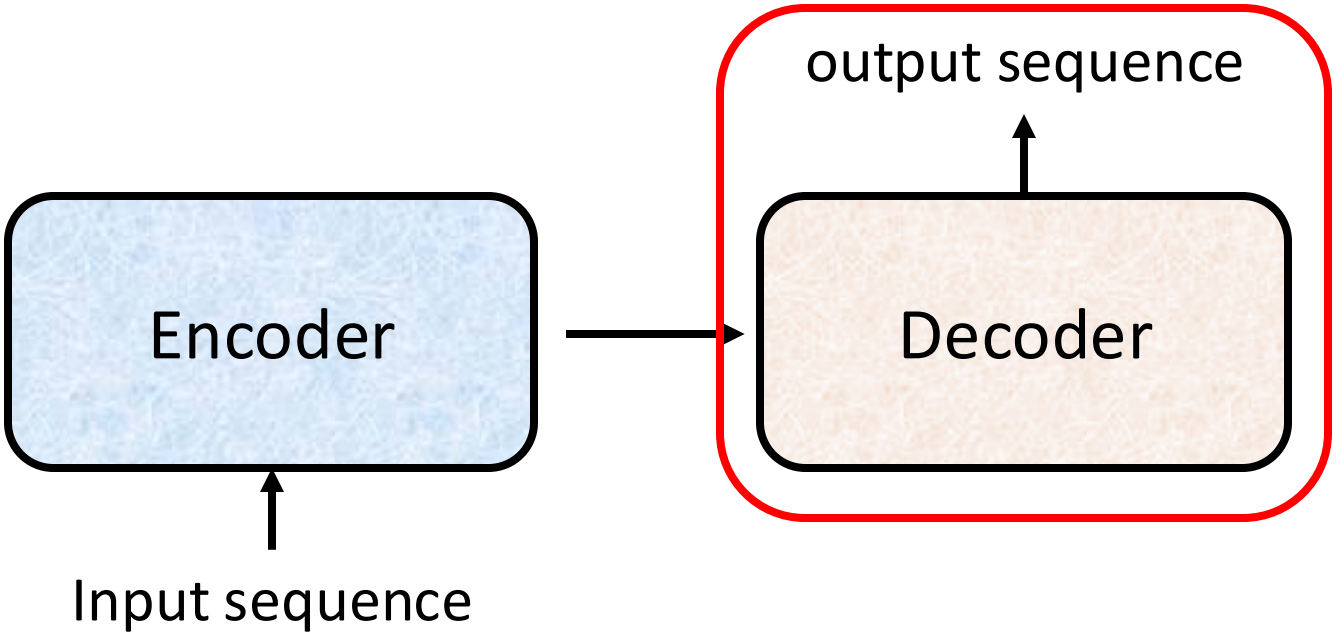


Autoregressive

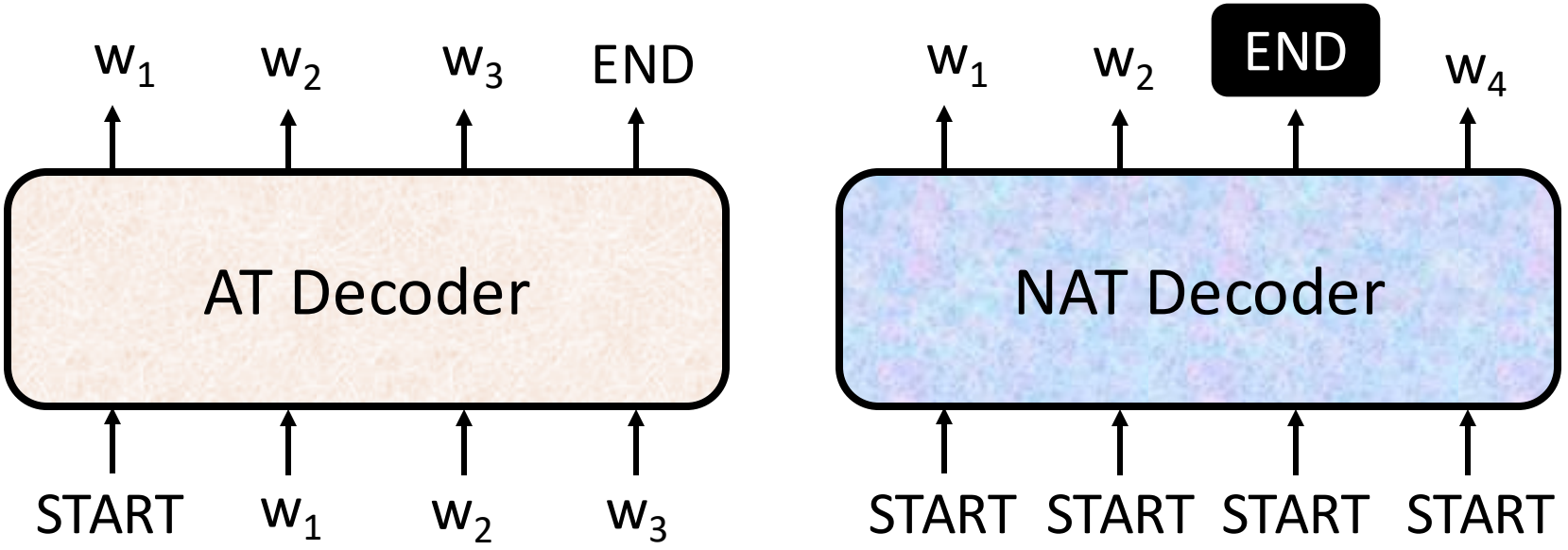
Stop at here!



Decoder
– Non-autoregressive (NAT)

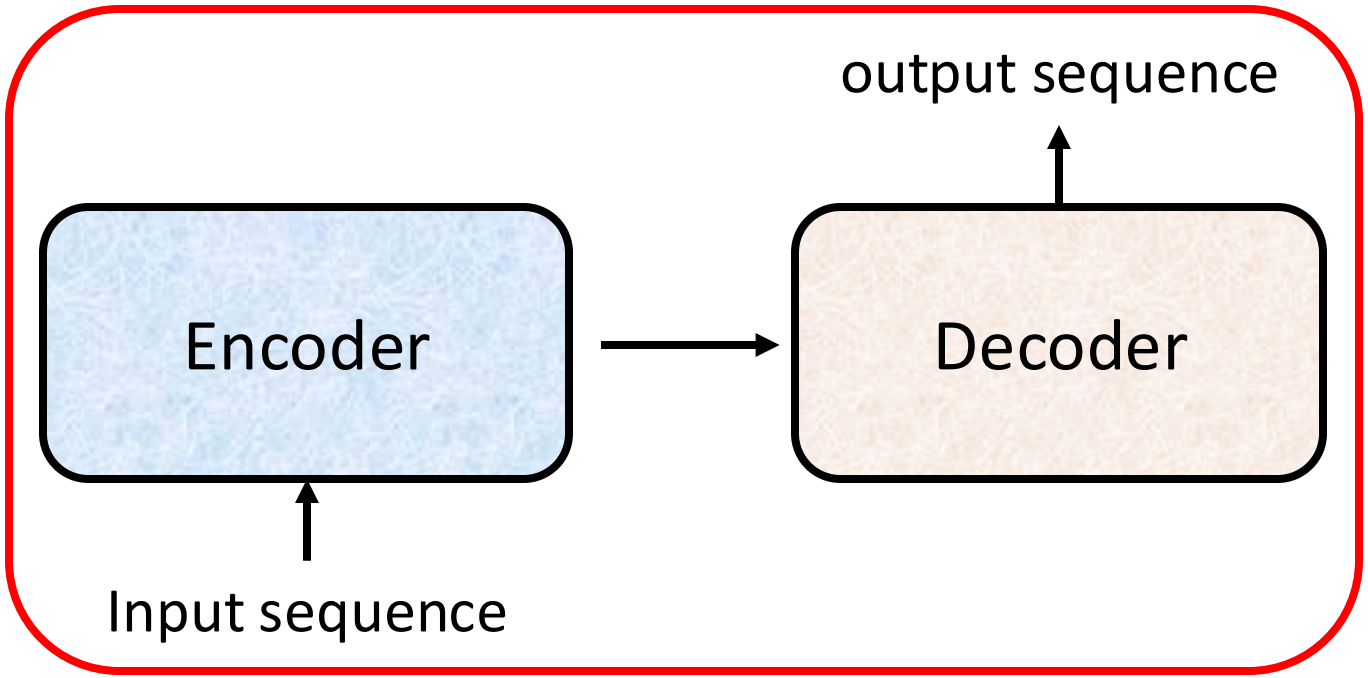


AT v.s. NAT



- How to decide the output length for NAT decoder?
  - Another predictor for output length
  - Output a very long sequence, ignore tokens after END
- Advantage: parallel, more stable generation (e.g., TTS)
- NAT is usually worse than AT (why? **Multi-modality**)

Encoder-Decoder



Concluding Remarks: Transformer