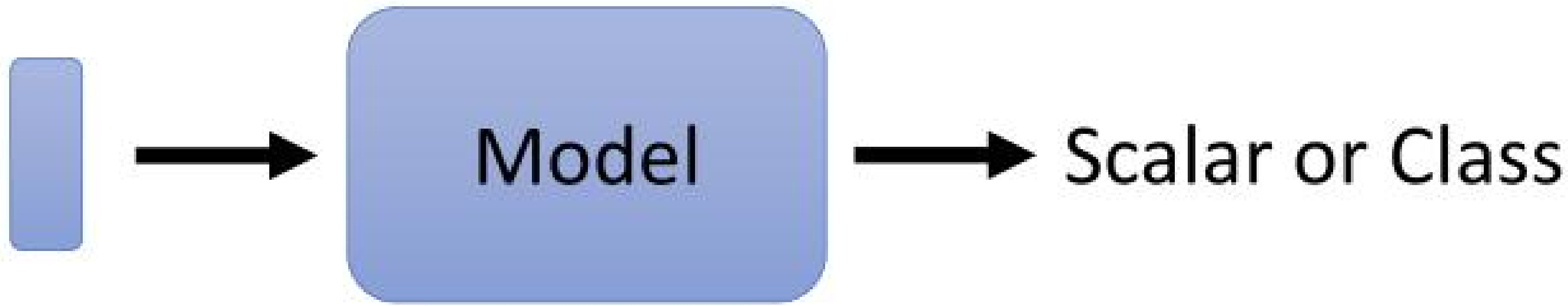
人工智能技术及应用

Artificial Intelligence and Application

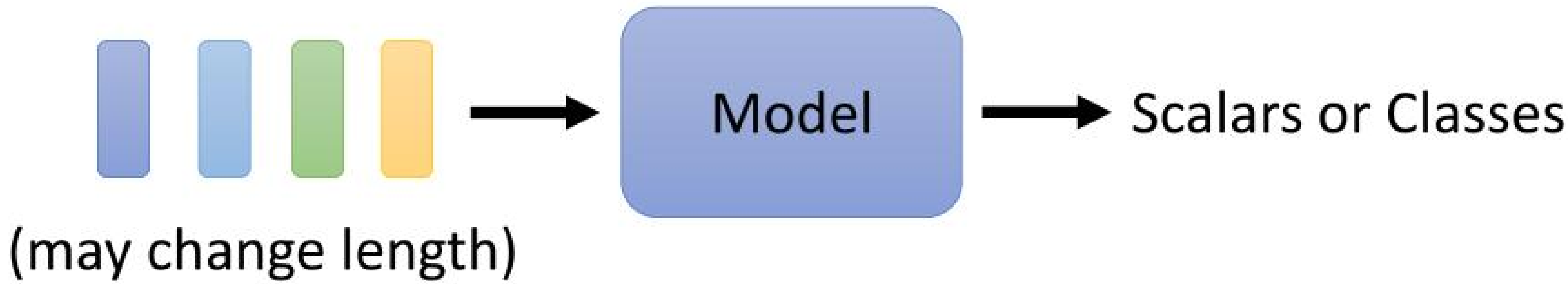
Self-attention

Sophisticated Input

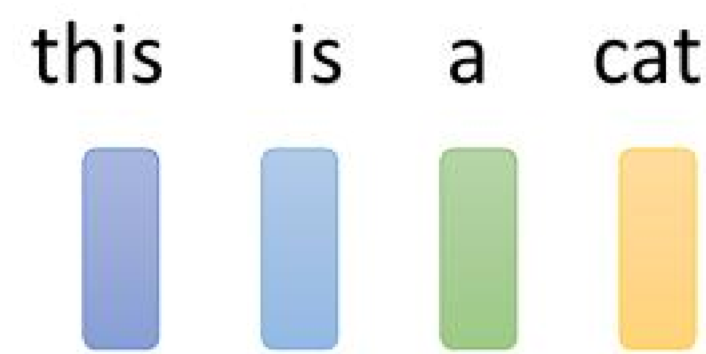
- Input is **a vector**



- Input is **a set of vectors**



Vector Set as Input



One-hot Encoding

apple = [1 0 0 0 0 ……]

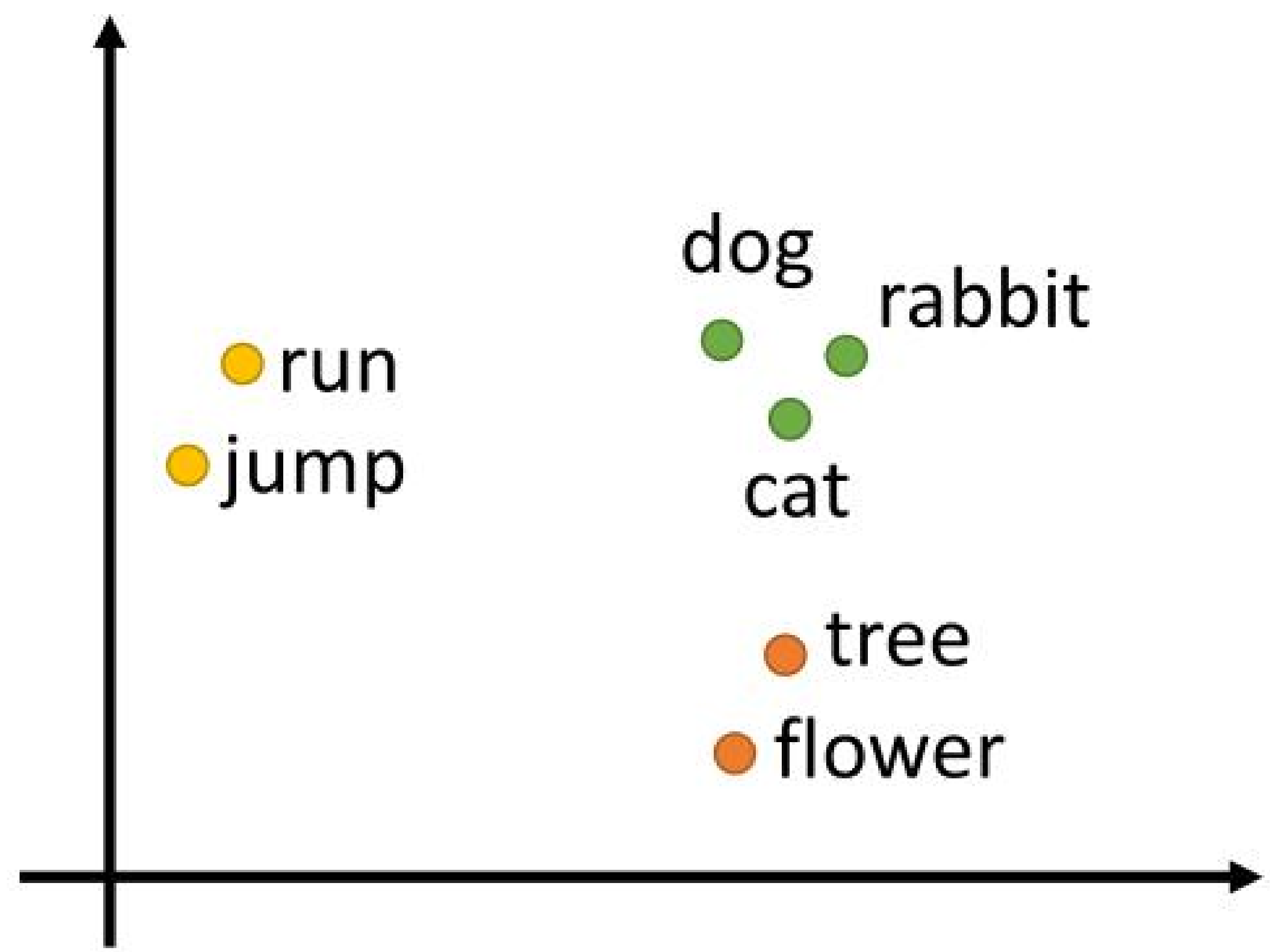
bag = [0 1 0 0 0 ……]

cat = [0 0 1 0 0 ……]

dog = [0 0 0 1 0 ……]

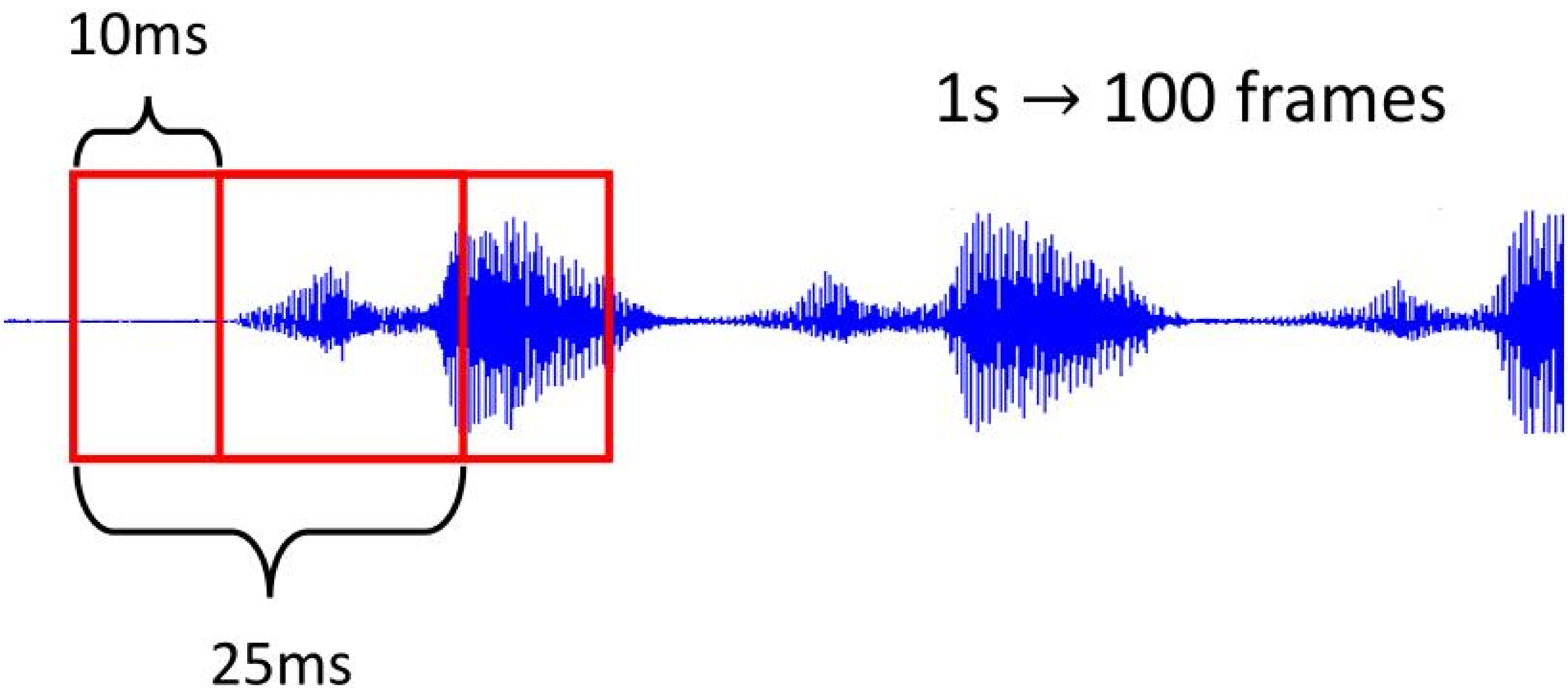
elephant = [0 0 0 0 1 ……]

Word Embedding



Vector Set as Input

1s → 100 frames



frame

400 sample points (16KHz)
39-dim MFCC
80-dim filter bank output

https://medium.com/analytics-vidhya/social-network-analytics-f082f4e21b16

Vector Set as Input

- Graph is also a set of vectors (consider each **node** as **a vector**)

Each profile is a vector

http://www.twword.com/wiki/%E5%88%86%E5%AD%90%90

Vector Set as Input

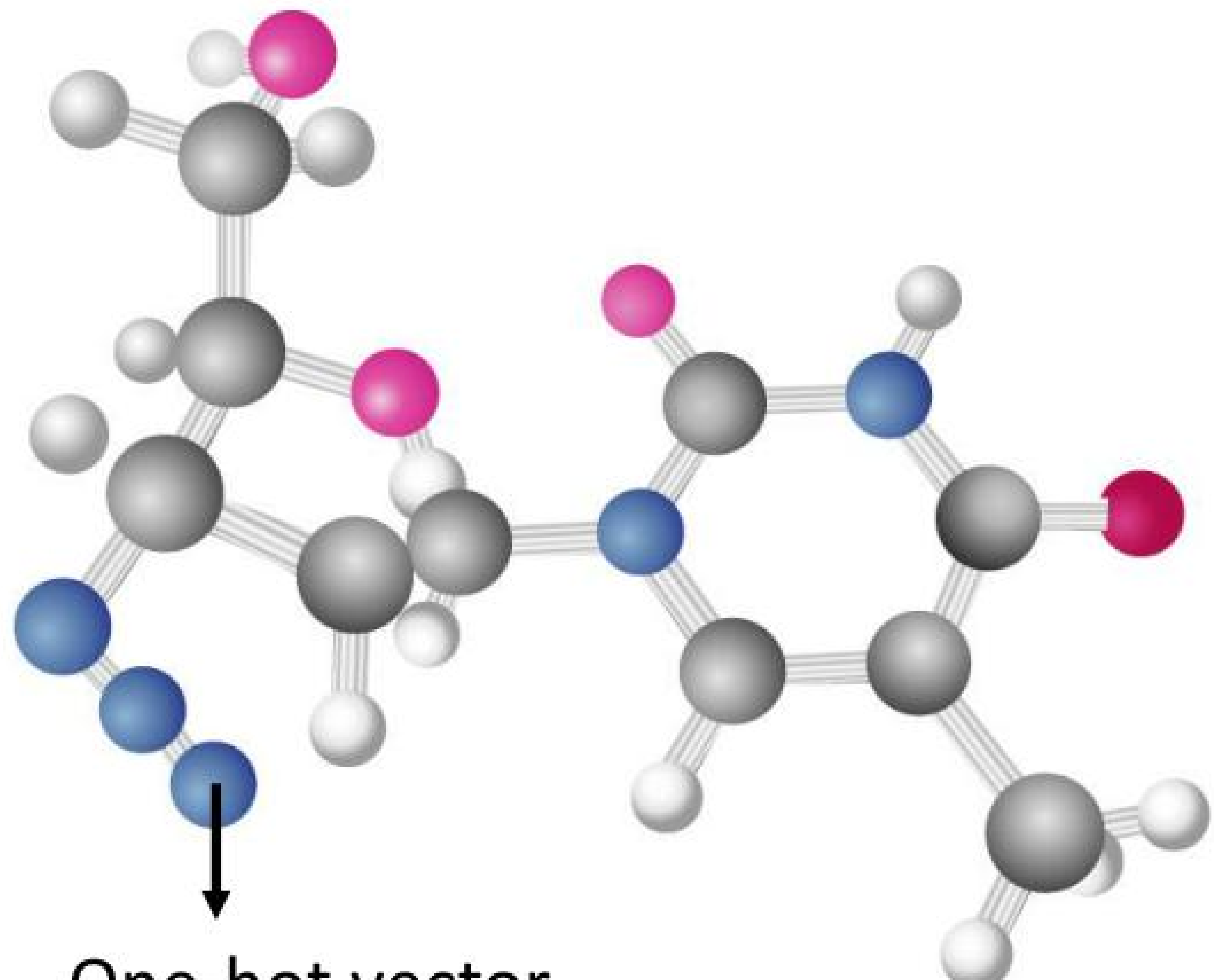
- Graph is also a set of vectors (consider each **node** as **a vector**)

H = [1 0 0 0 0 ……]

C = [0 1 0 0 0 ……]

O = [0 0 1 0 0 ……]

⋮

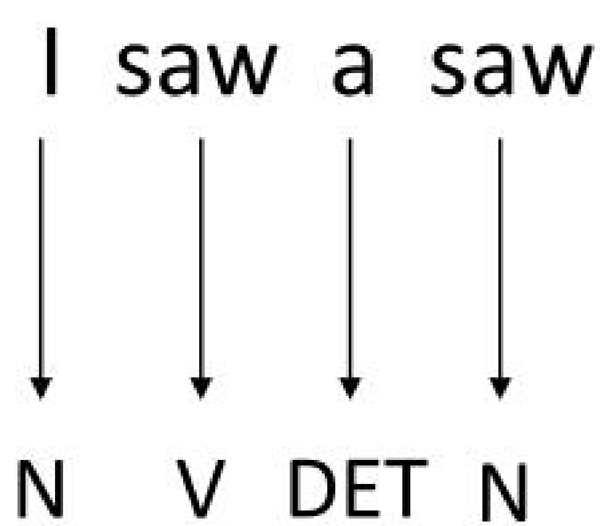


One-hot vector

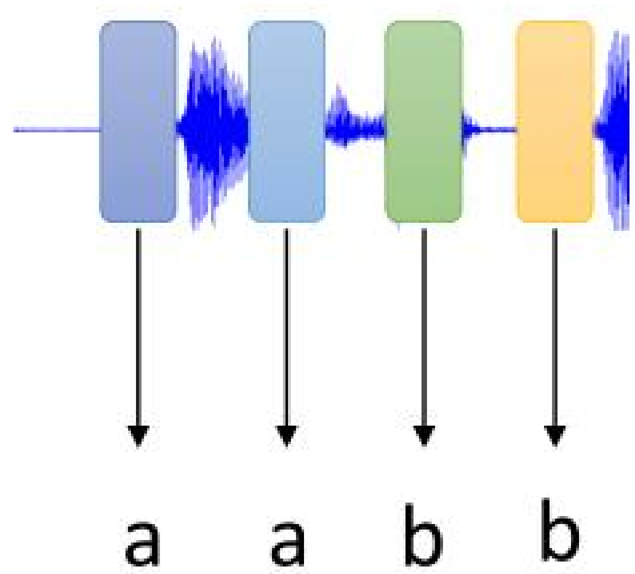
What is the output?

- Each vector has a label.

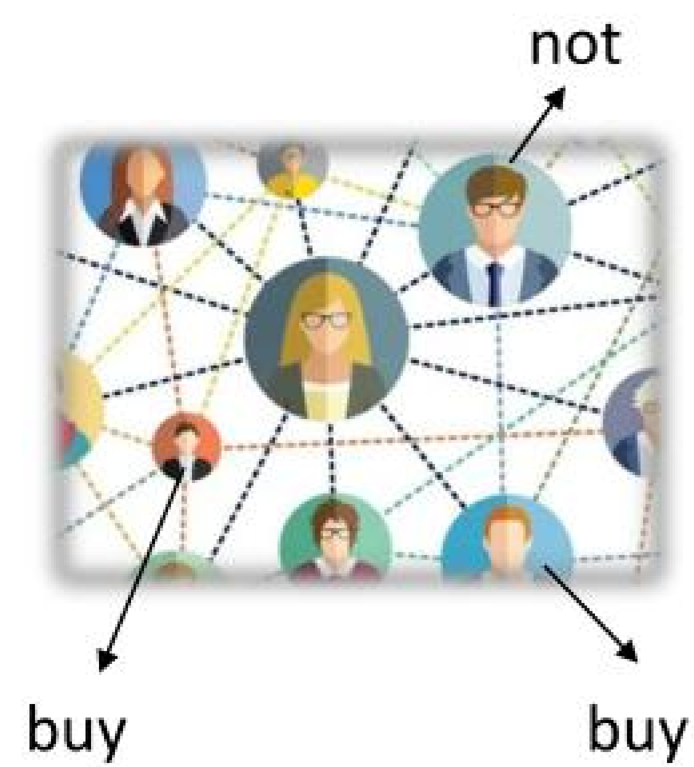
Example Applications



POS tagging



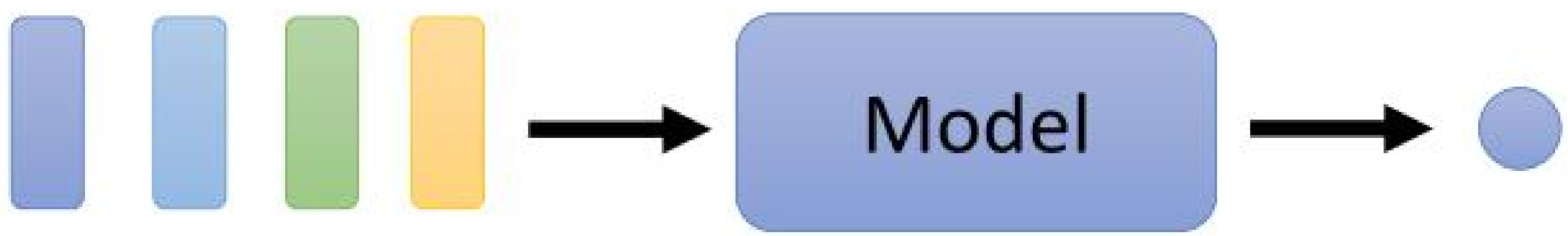
Phoneme recognition



What is the output?

- Each vector has a label.

- The whole sequence has a label.

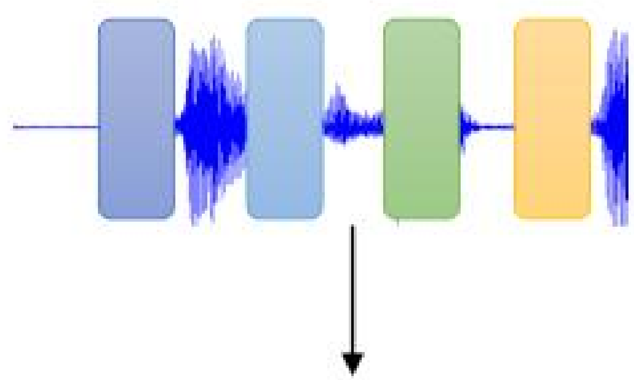


Example Applications

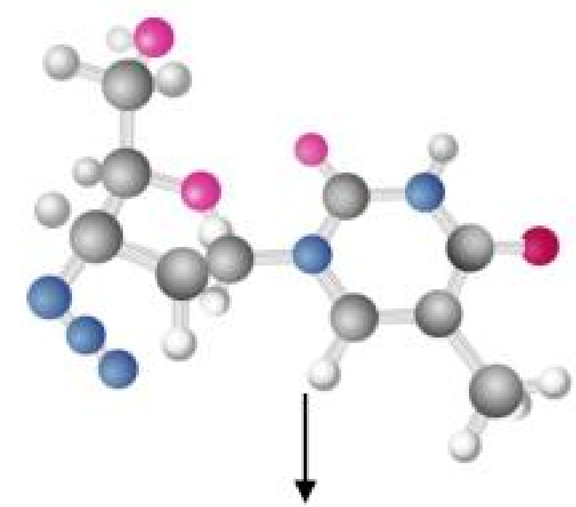
this is good

Sentiment analysis

positive



speaker

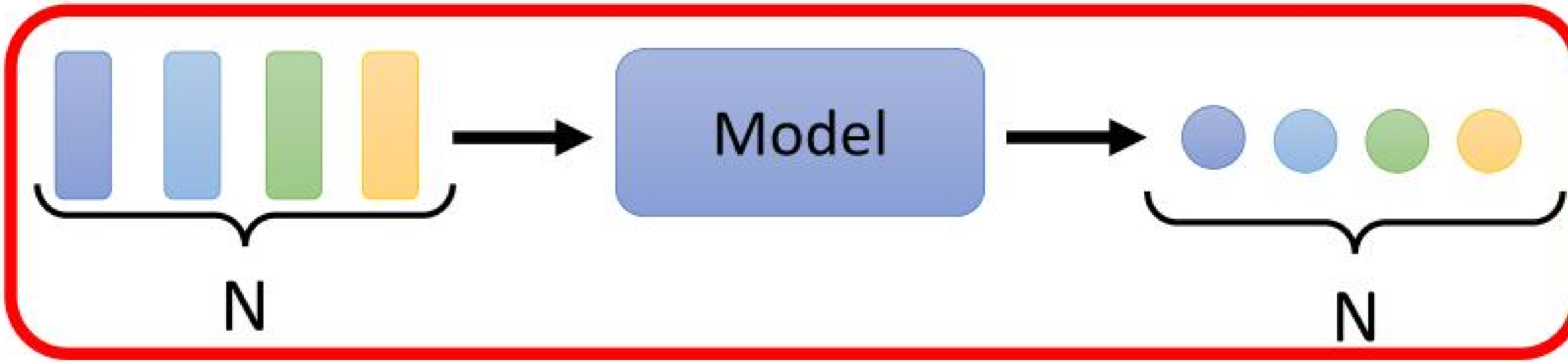


hydrophilicity

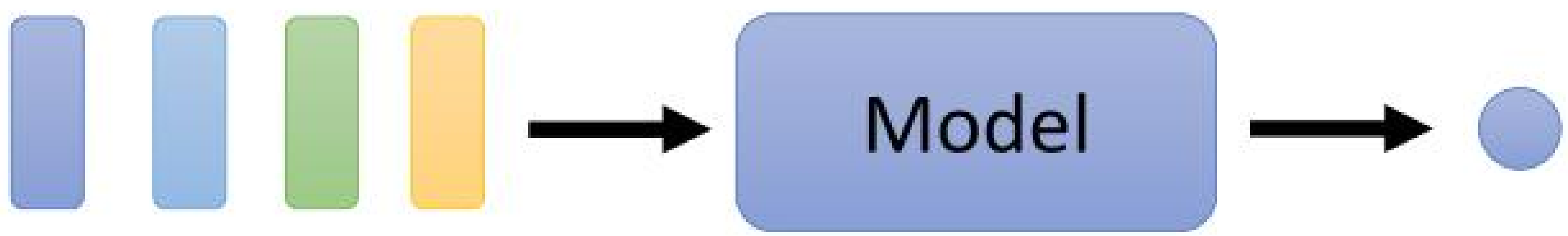
What is the output?

- Each vector has a label.

focus of this lecture



- The whole sequence has a label.



- Model decides the number of labels itself.

seq2seq

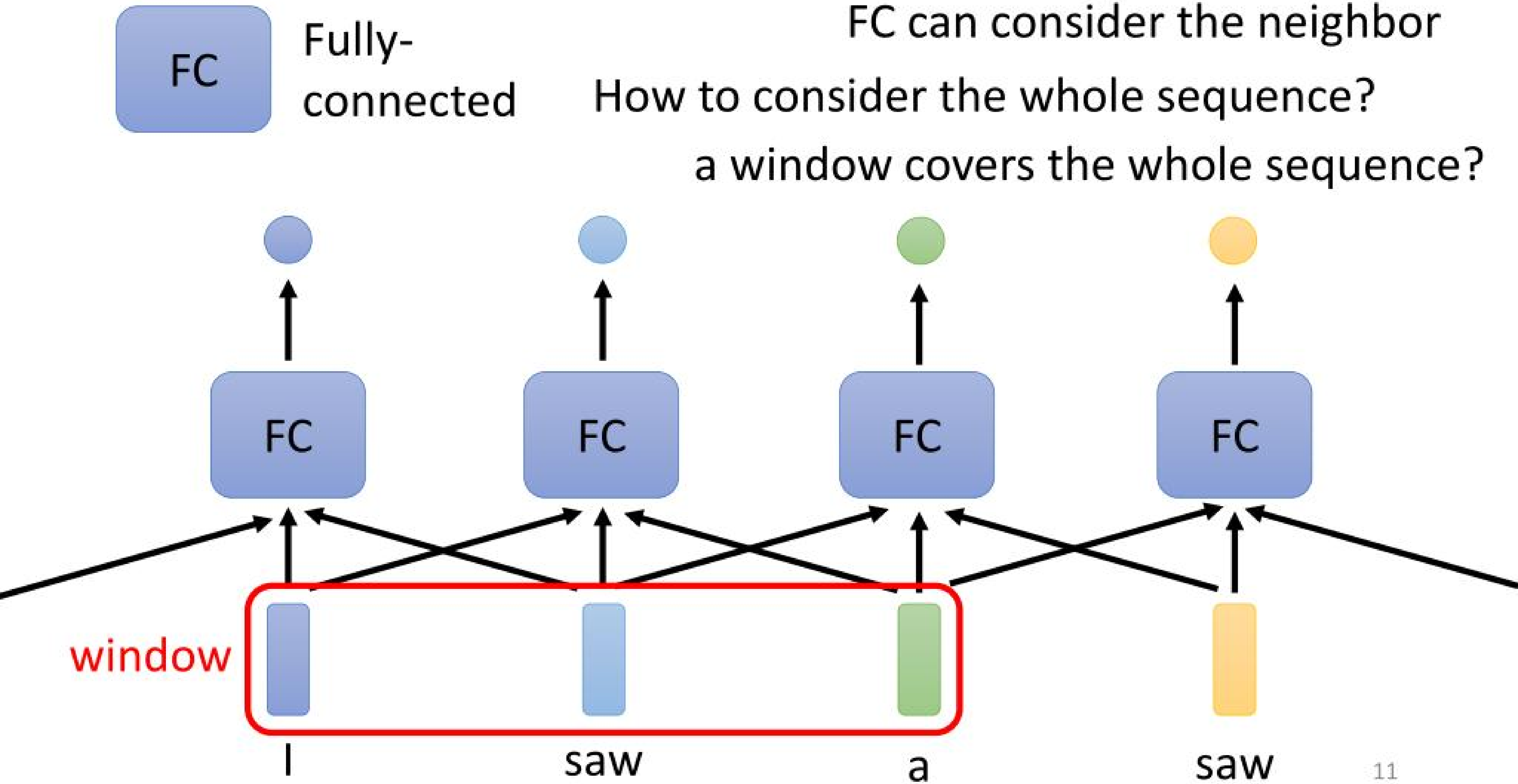
Sequence Labeling

Is it possible to consider the context?

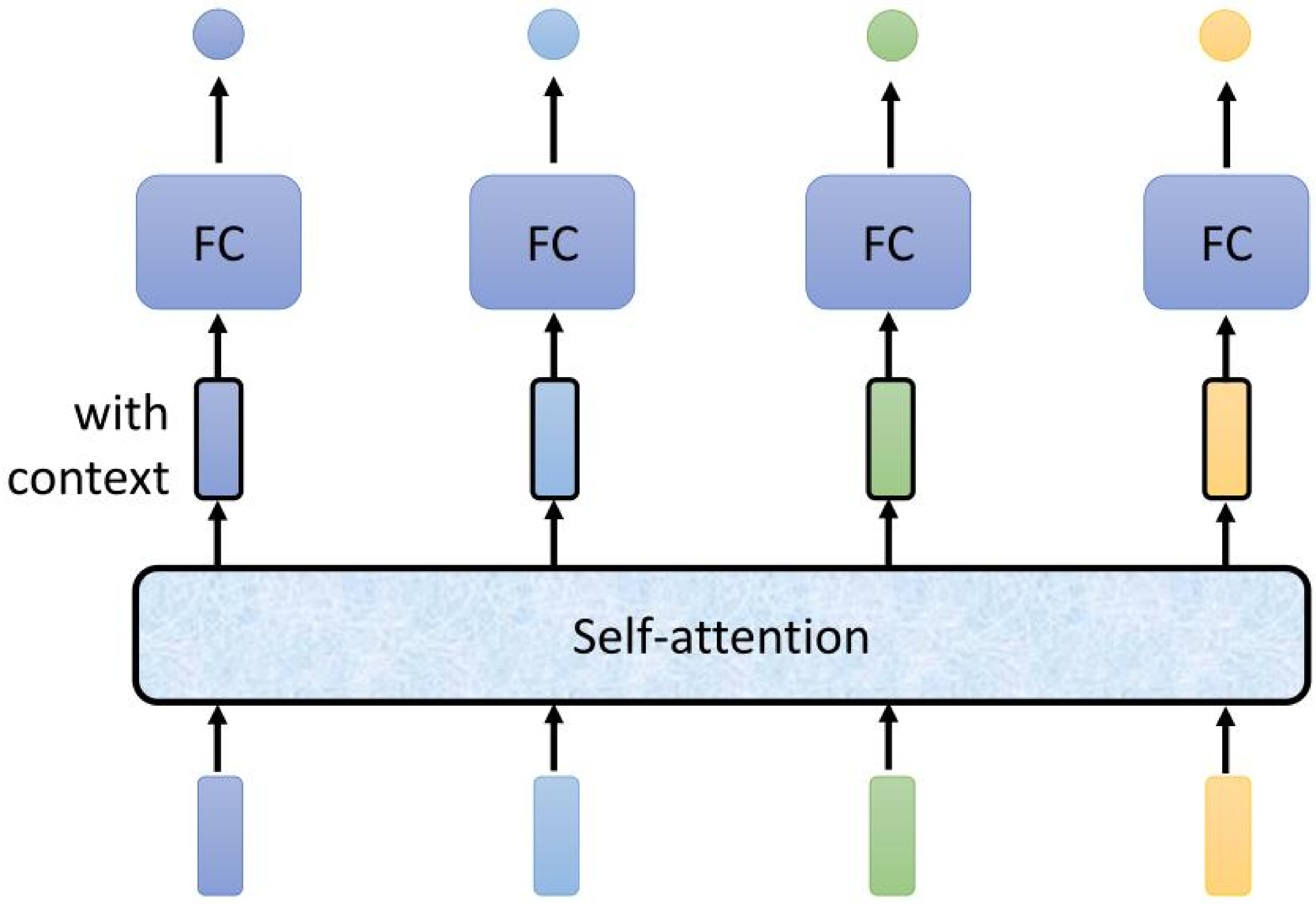
FC can consider the neighbor

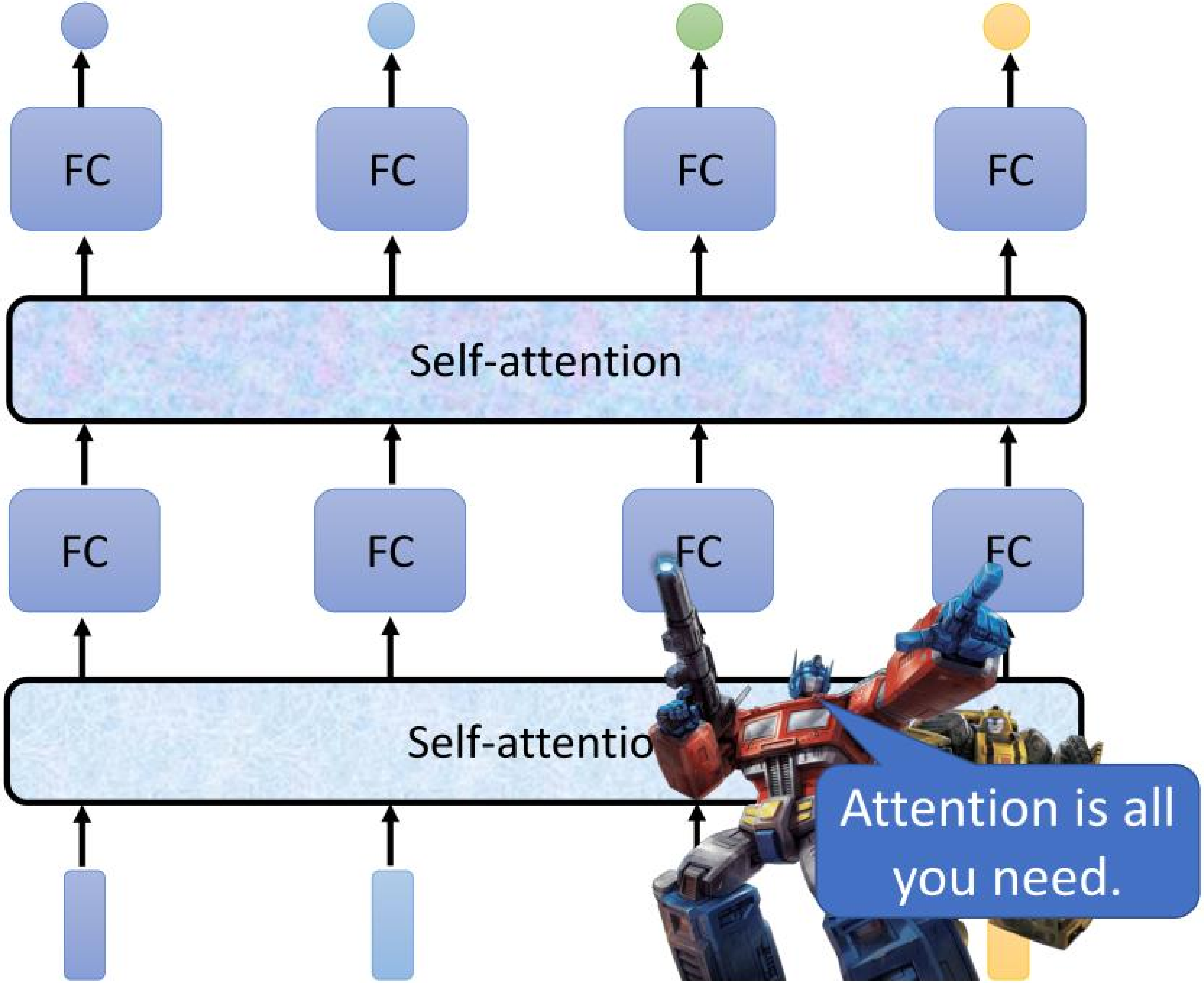
How to consider the whole sequence?

a window covers the whole sequence?



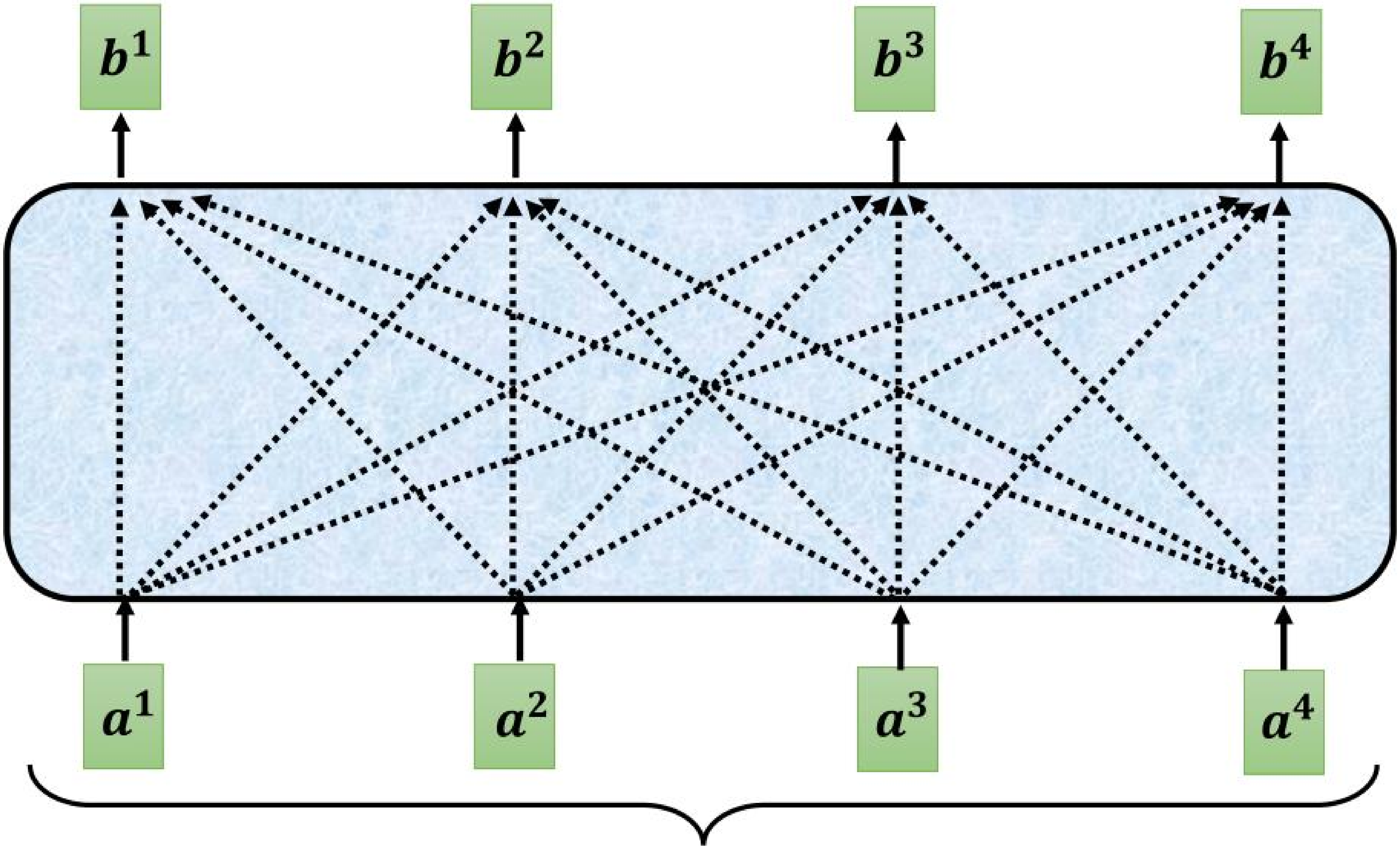
Self-attention





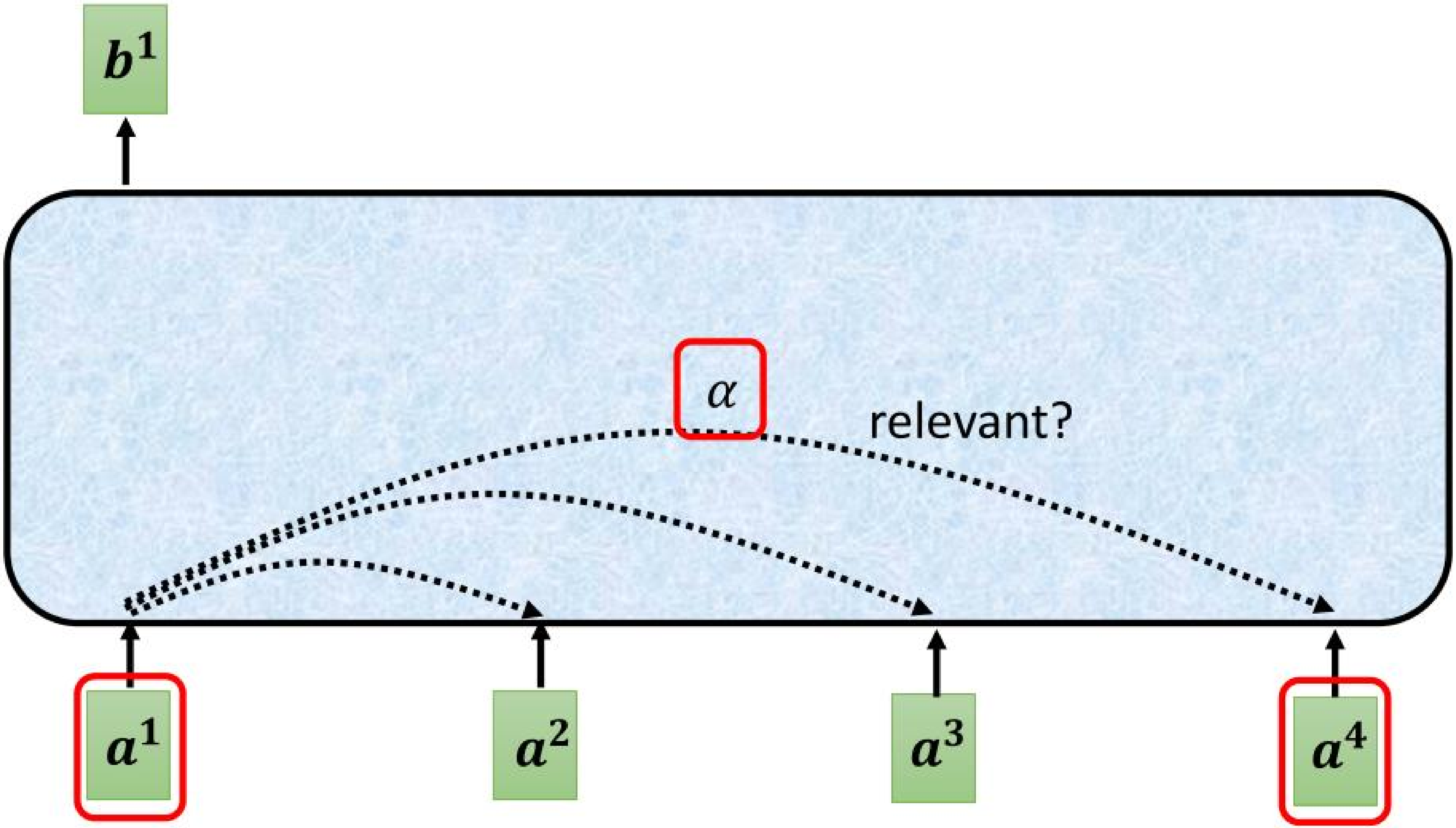
https://arxiv.org/abs/1706.03762

Self-attention



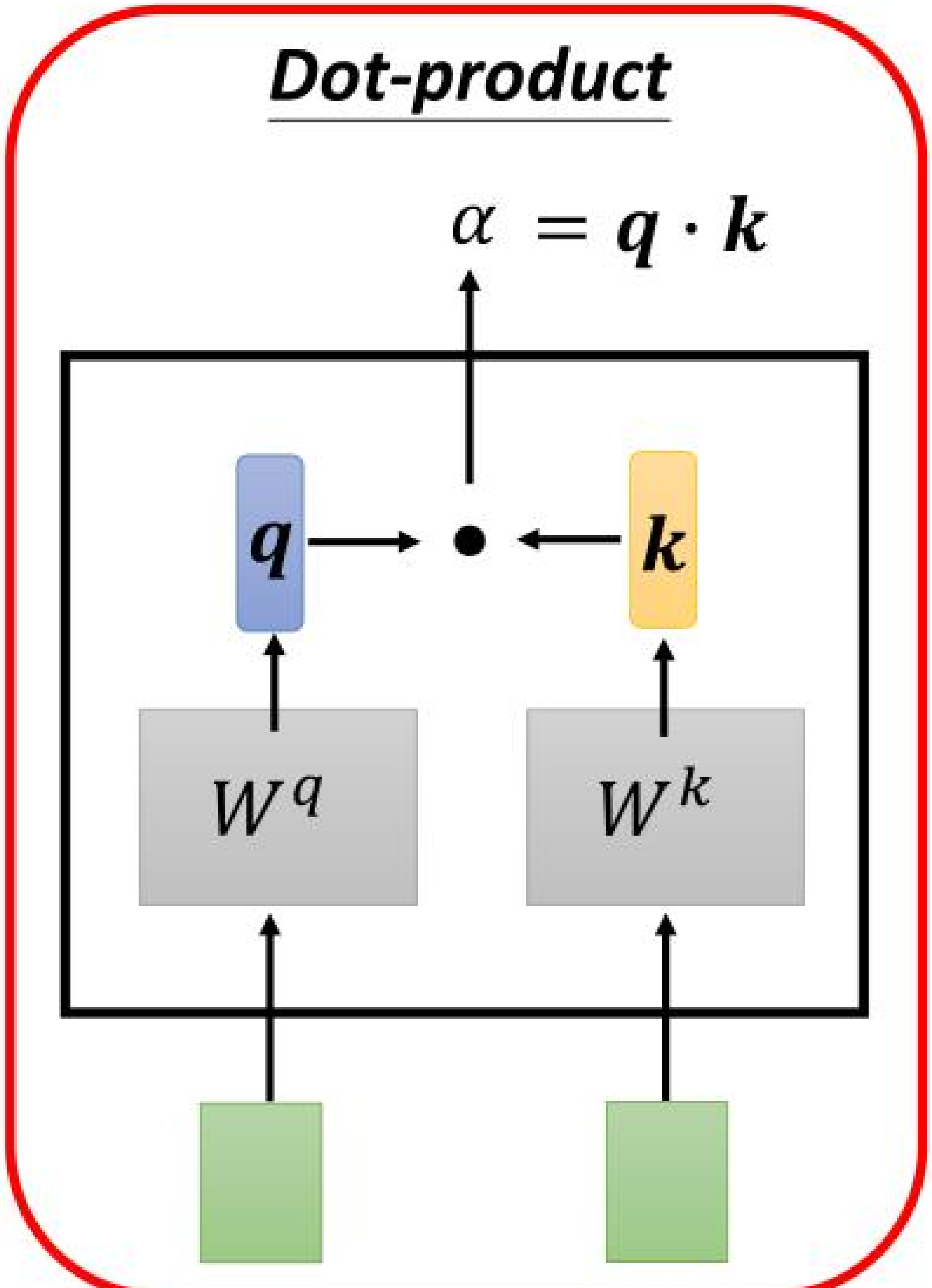
Can be either **input** or **a hidden layer**

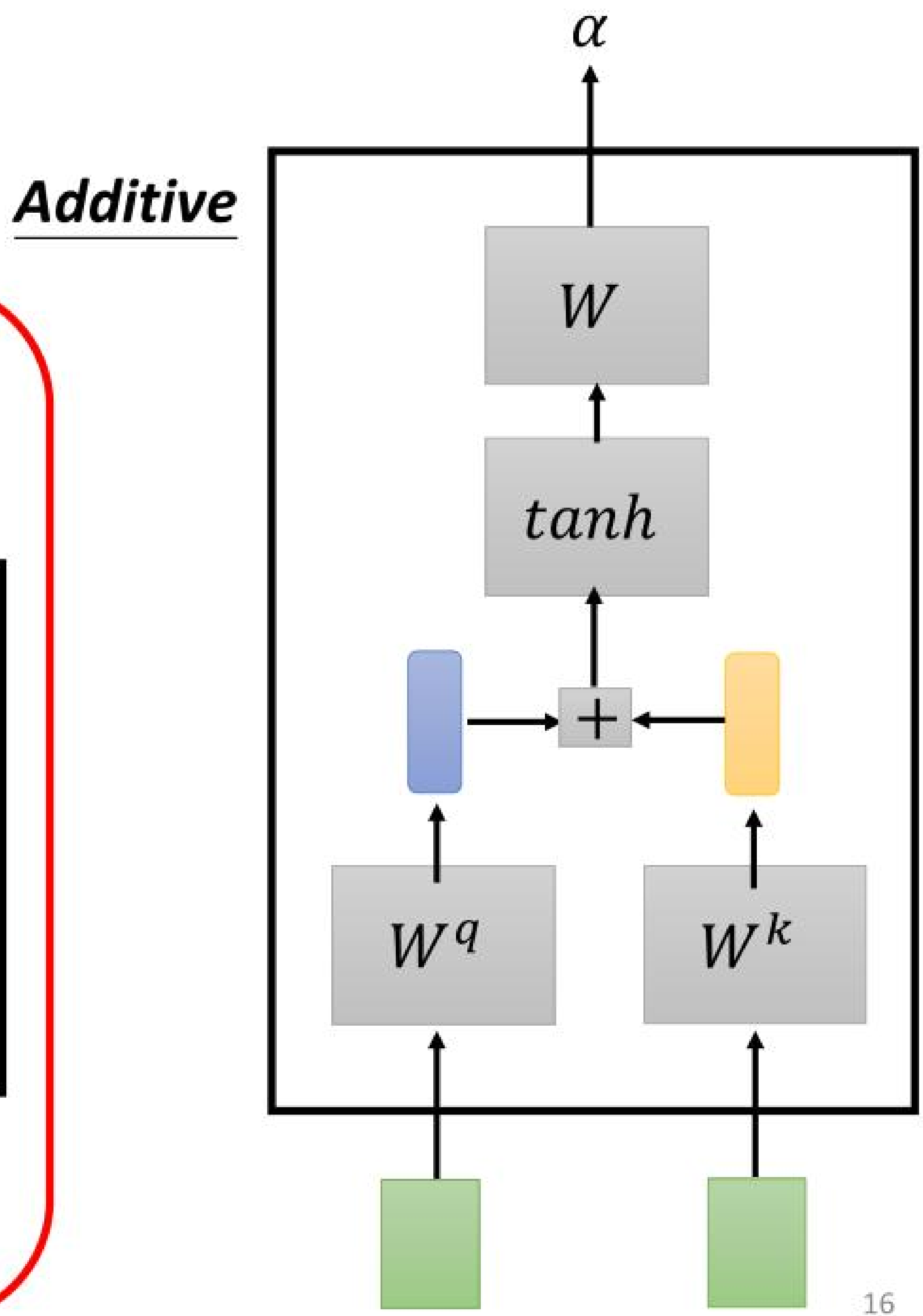
Self-attention



Find the relevant vectors in a sequence

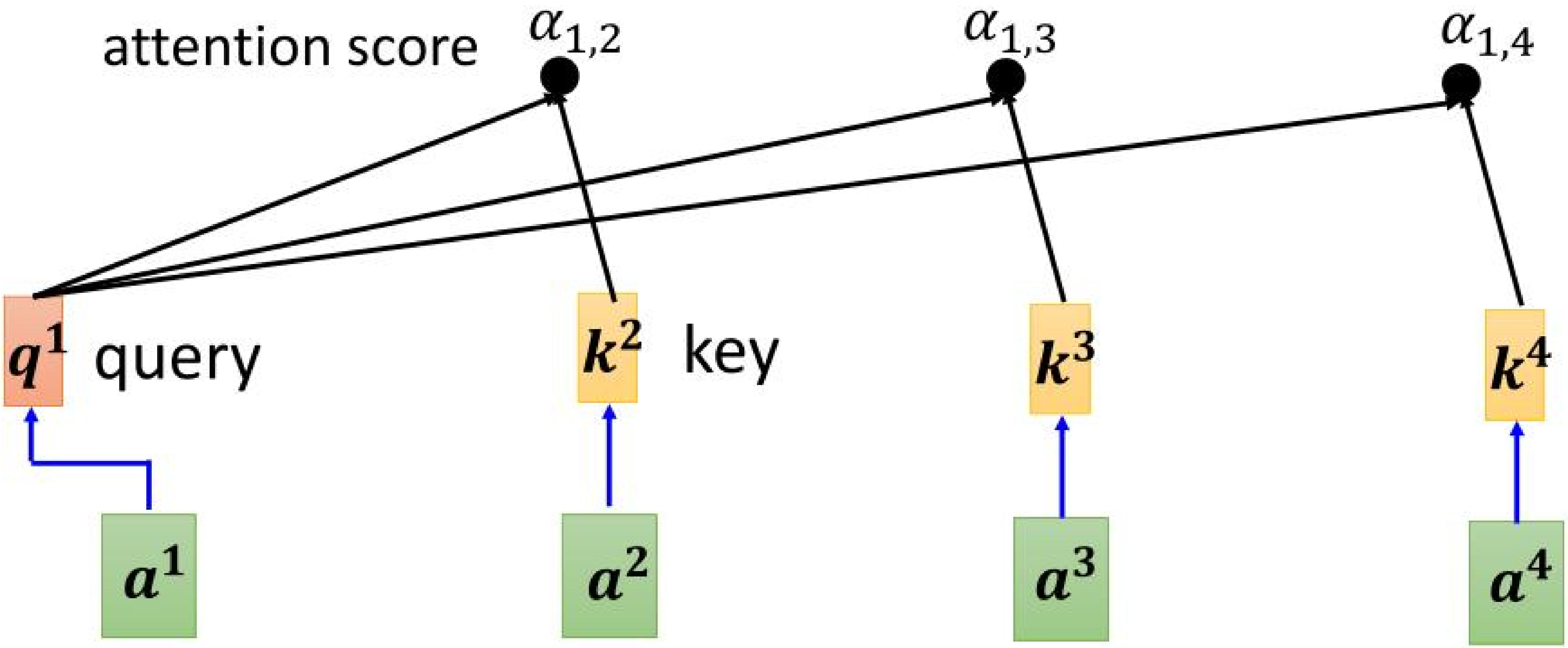
Self-attention





Self-attention

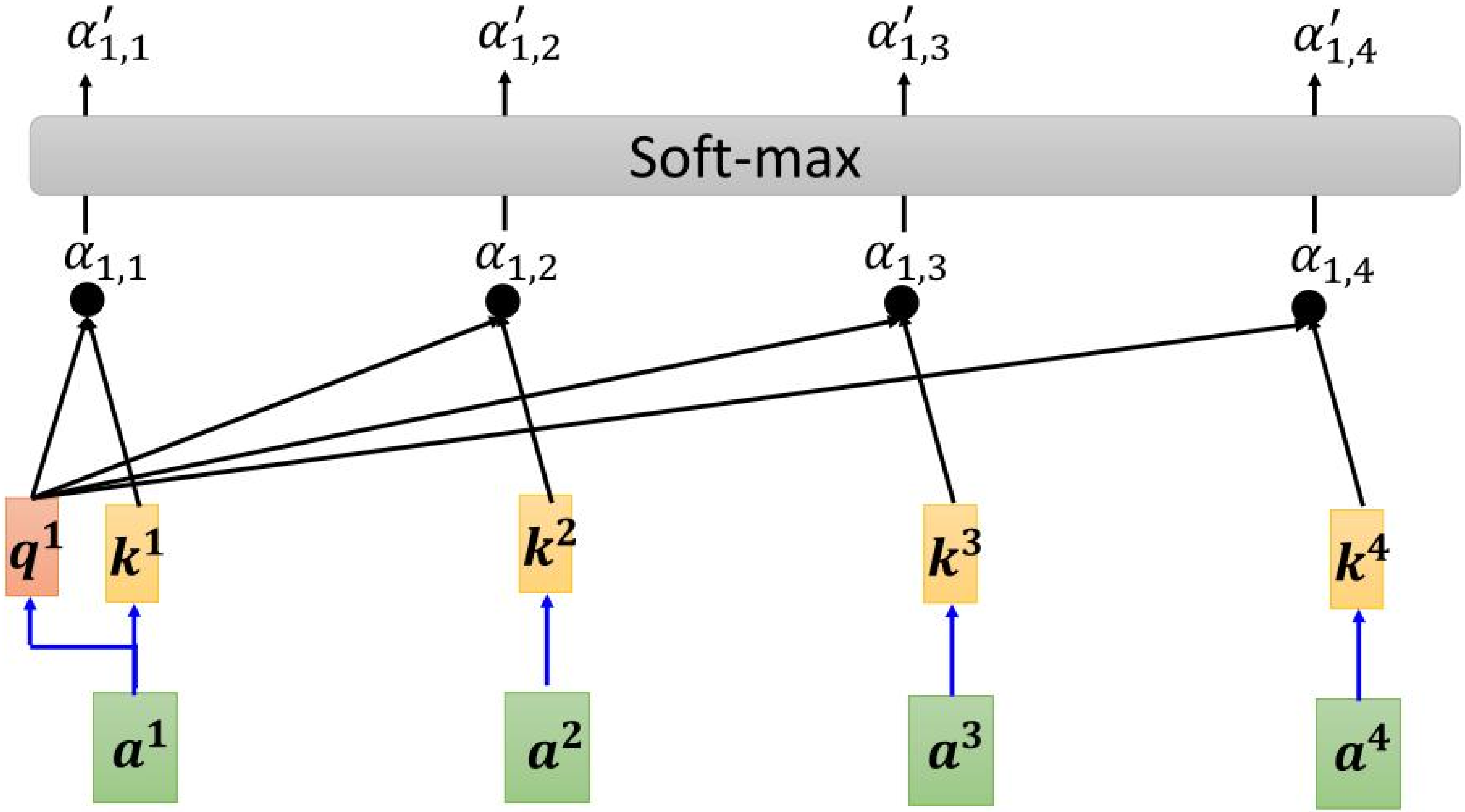
$\alpha_{1,2} = \boldsymbol{q}^1 \cdot \boldsymbol{k}^2$ $\alpha_{1,3} = \boldsymbol{q}^1 \cdot \boldsymbol{k}^3$ $\alpha_{1,4} = \boldsymbol{q}^1 \cdot \boldsymbol{k}^4$



$\boldsymbol{q}^1 = W^q \boldsymbol{a}^1$ $\boldsymbol{k}^2 = W^k \boldsymbol{a}^2$ $\boldsymbol{k}^3 = W^k \boldsymbol{a}^3$ $\boldsymbol{k}^4 = W^k \boldsymbol{a}^4$

Self-attention

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

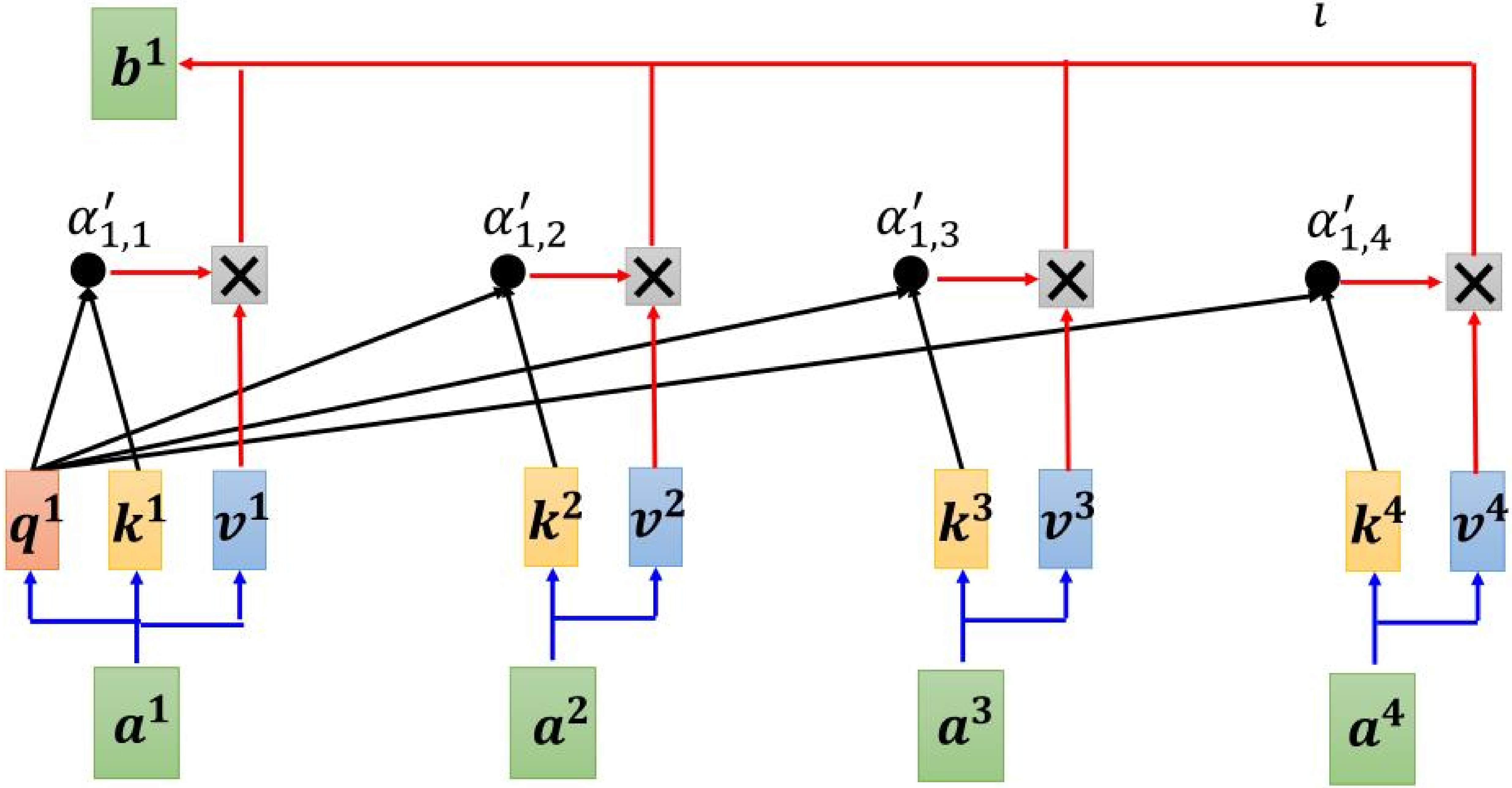


$\boldsymbol{q}^1 = W^q \boldsymbol{a}^1$ $\boldsymbol{k}^2 = W^k \boldsymbol{a}^2$ $\boldsymbol{k}^3 = W^k \boldsymbol{a}^3$ $\boldsymbol{k}^4 = W^k \boldsymbol{a}^4$

$\boldsymbol{k}^1 = W^k \boldsymbol{a}^1$

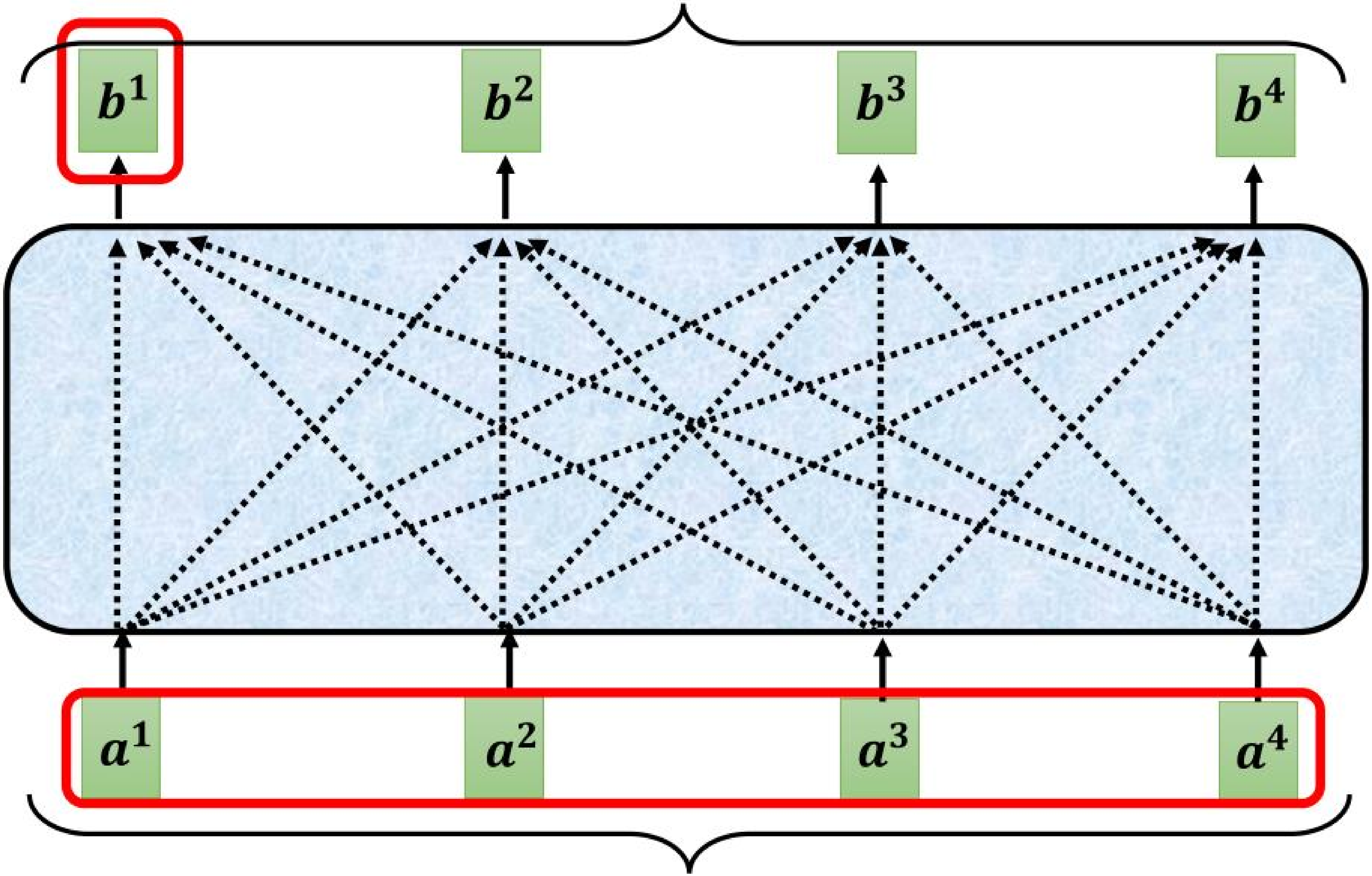
Self-attention Extract information based on attention scores

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



$\boldsymbol{v}^1 = W^v \boldsymbol{a}^1$ $\boldsymbol{v}^2 = W^v \boldsymbol{a}^2$ $\boldsymbol{v}^3 = W^v \boldsymbol{a}^3$ $\boldsymbol{v}^4 = W^v \boldsymbol{a}^4$

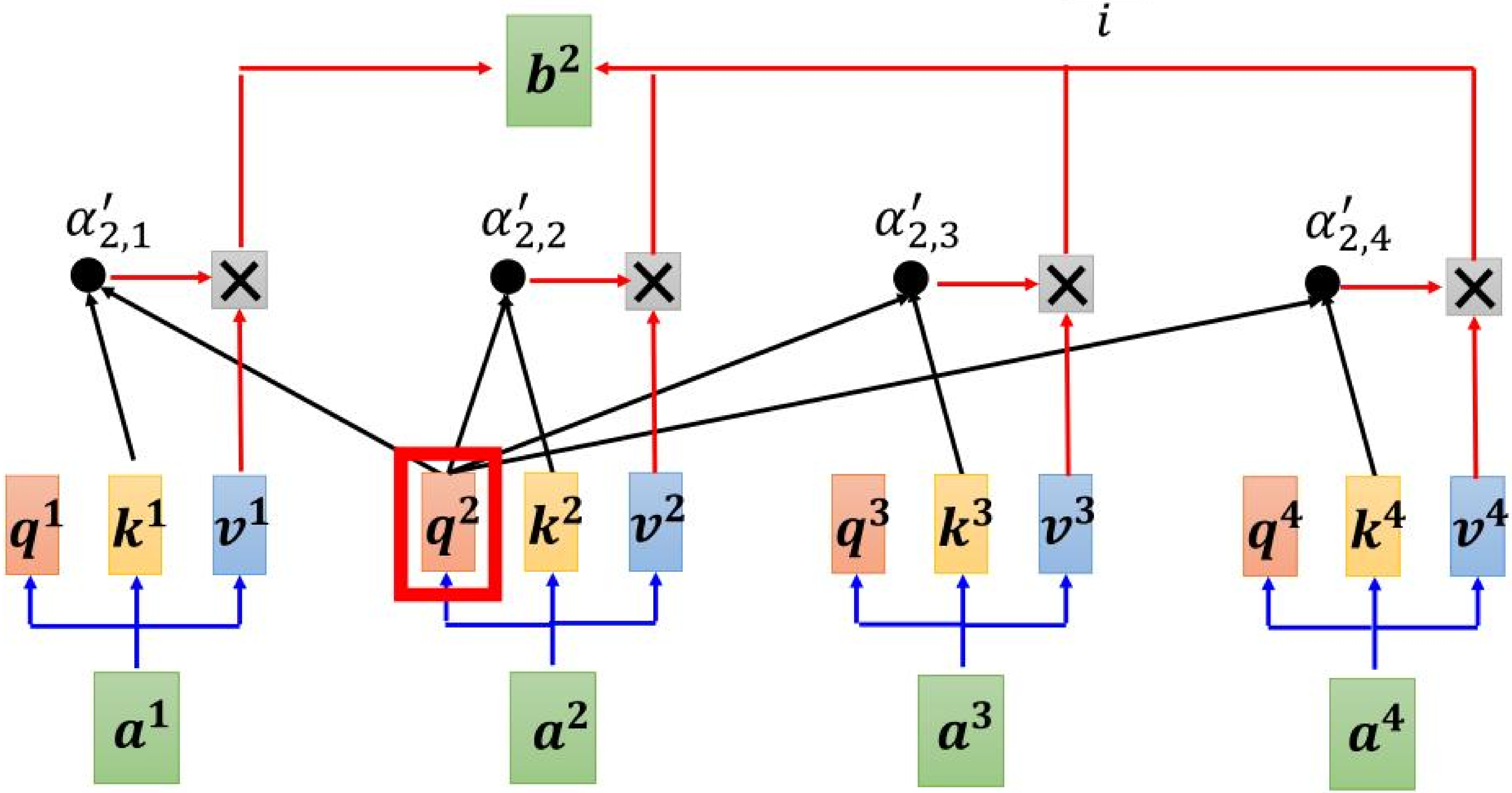
Self-attention



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

Self-attention

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

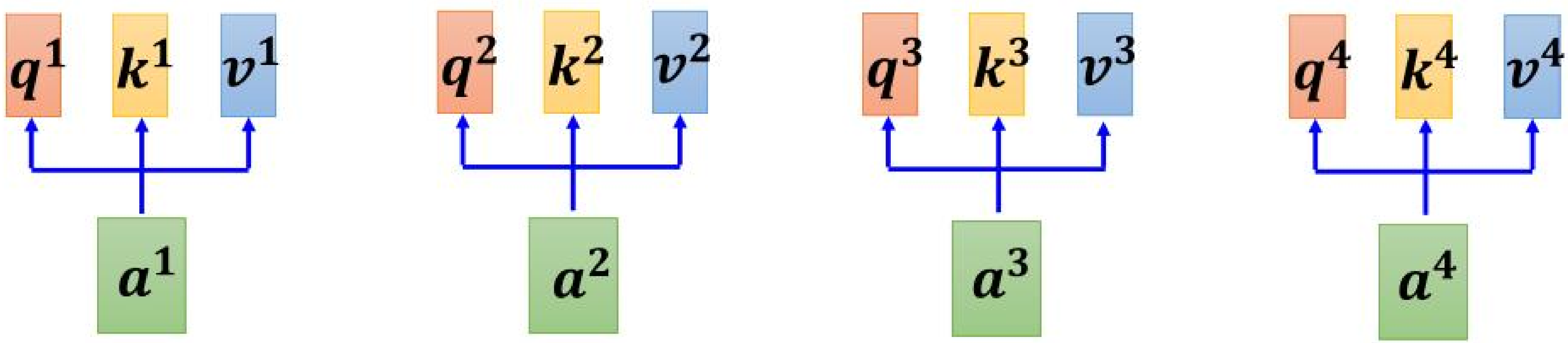


Self-attention

$$q^i = W^q a^i \qquad \underbrace{q^1\, q^2\, q^3\, q^4}_{Q} = W^q\, \underbrace{a^1\, a^2\, a^3\, a^4}_{I}$$

$$k^i = W^k a^i \qquad \underbrace{k^1\, k^2\, k^3\, k^4}_{K} = W^k\, \underbrace{a^1\, a^2\, a^3\, a^4}_{I}$$

$$v^i = W^v a^i \qquad \underbrace{v^1\, v^2\, v^3\, v^4}_{V} = W^v\, \underbrace{a^1\, a^2\, a^3\, a^4}_{I}$$

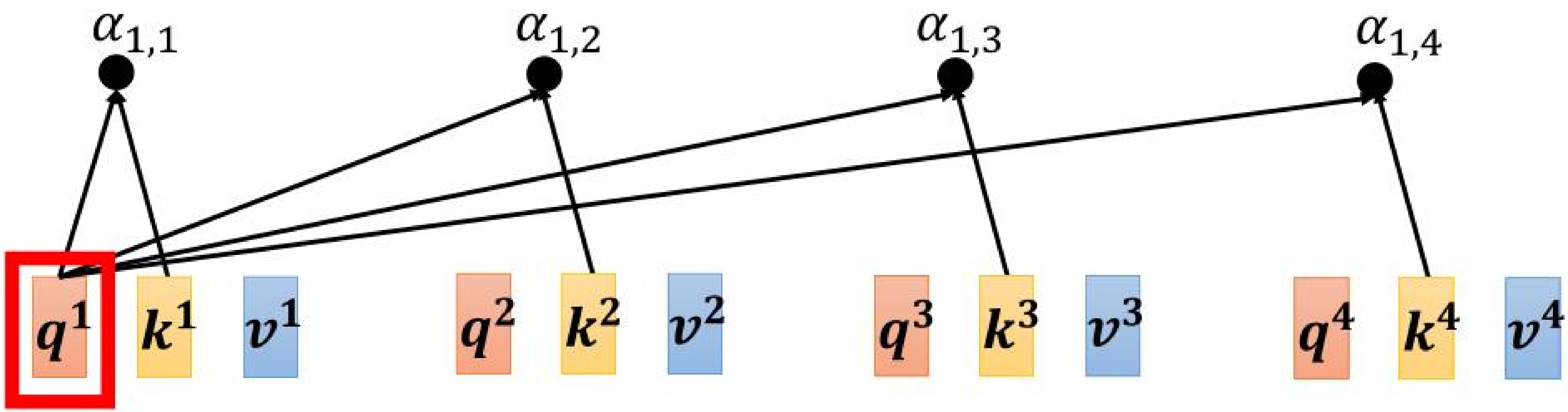


Self-attention

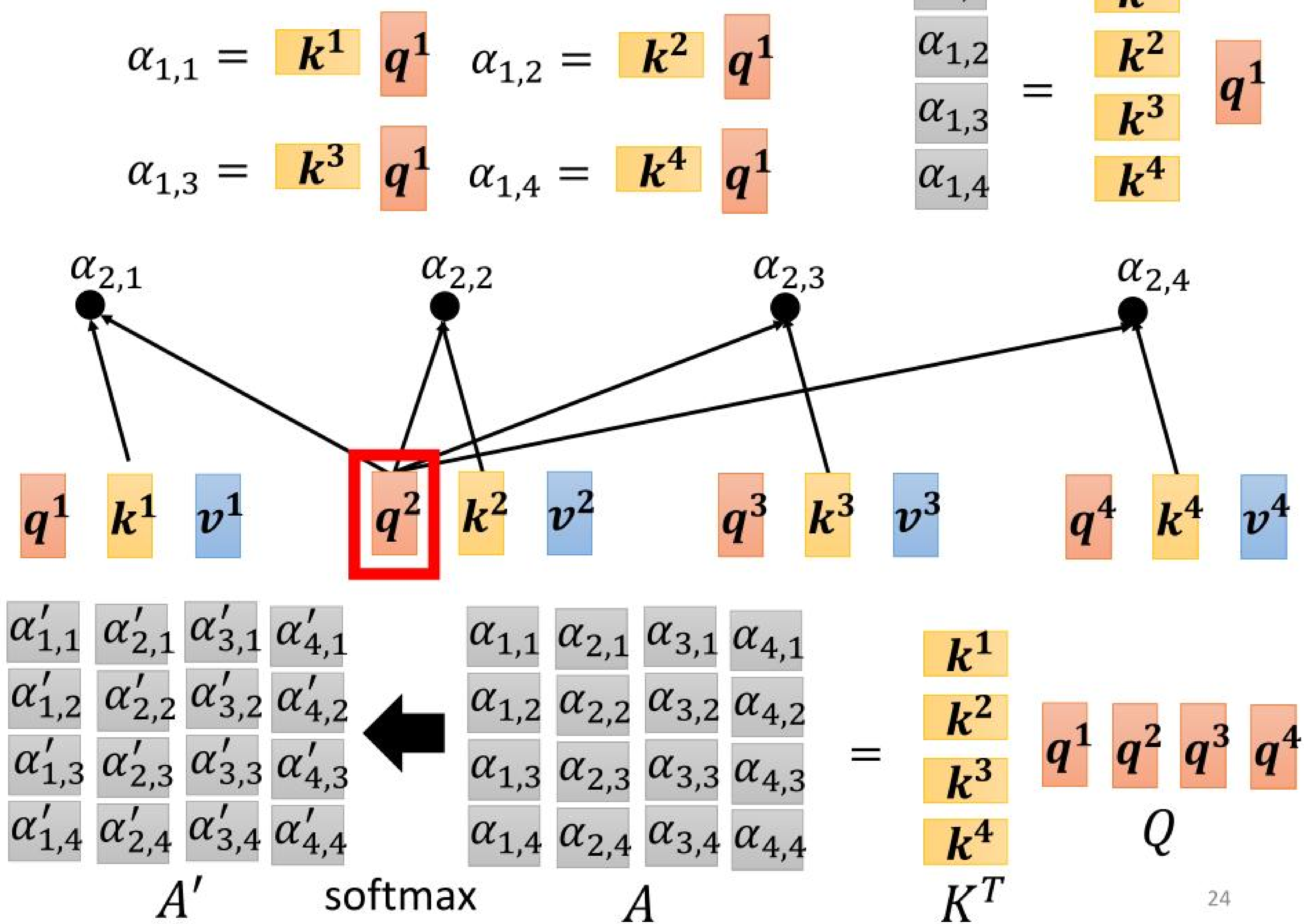
$$\alpha_{1,1} = \boldsymbol{k}^1 \boldsymbol{q}^1 \qquad \alpha_{1,2} = \boldsymbol{k}^2 \boldsymbol{q}^1$$

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

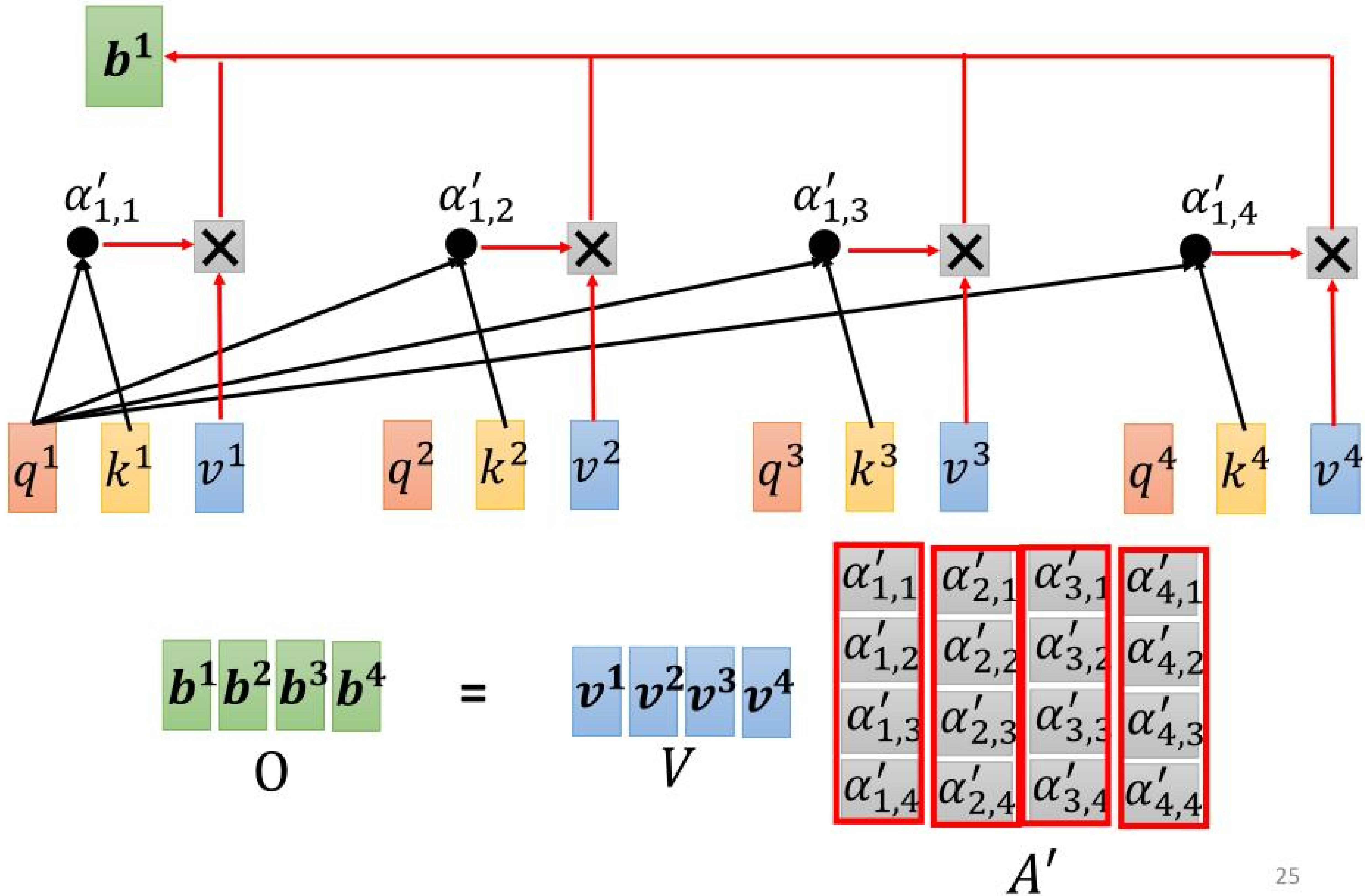
$$\begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}^1 \\ \boldsymbol{k}^2 \\ \boldsymbol{k}^3 \\ \boldsymbol{k}^4 \end{bmatrix} \boldsymbol{q}^1$$



Self-attention



Self-attention



Self-attention

$$Q = W^q I$$

$$K = W^k I$$

$$V = W^v I$$

Parameters to be learned

$$A = K^T Q$$

$$A' \leftarrow A$$

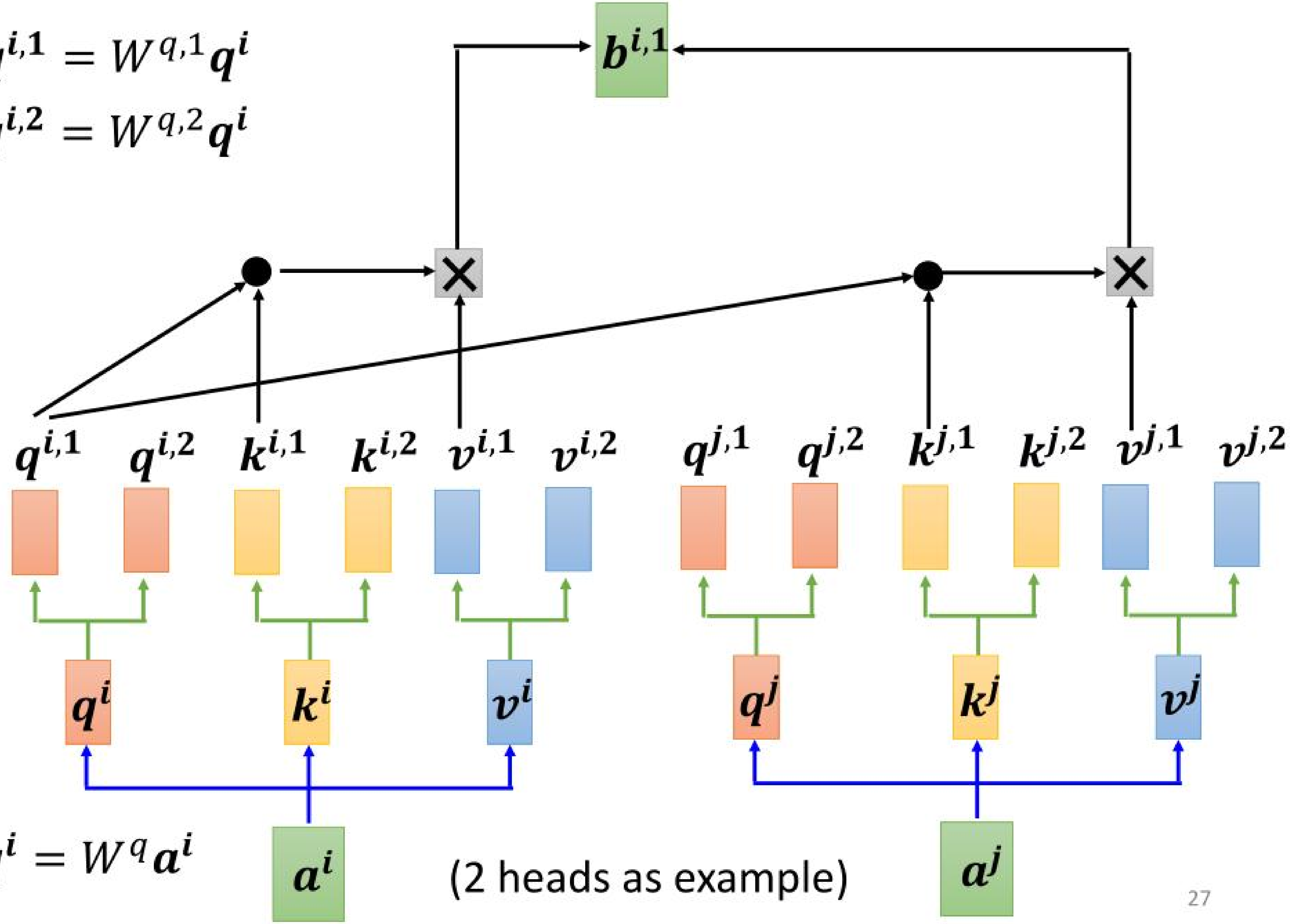
Attention Matrix

$$O = V A'$$

Multi-head Self-attention Different types of relevance

$q^{i,1} = W^{q,1} q^i$

$q^{i,2} = W^{q,2} q^i$

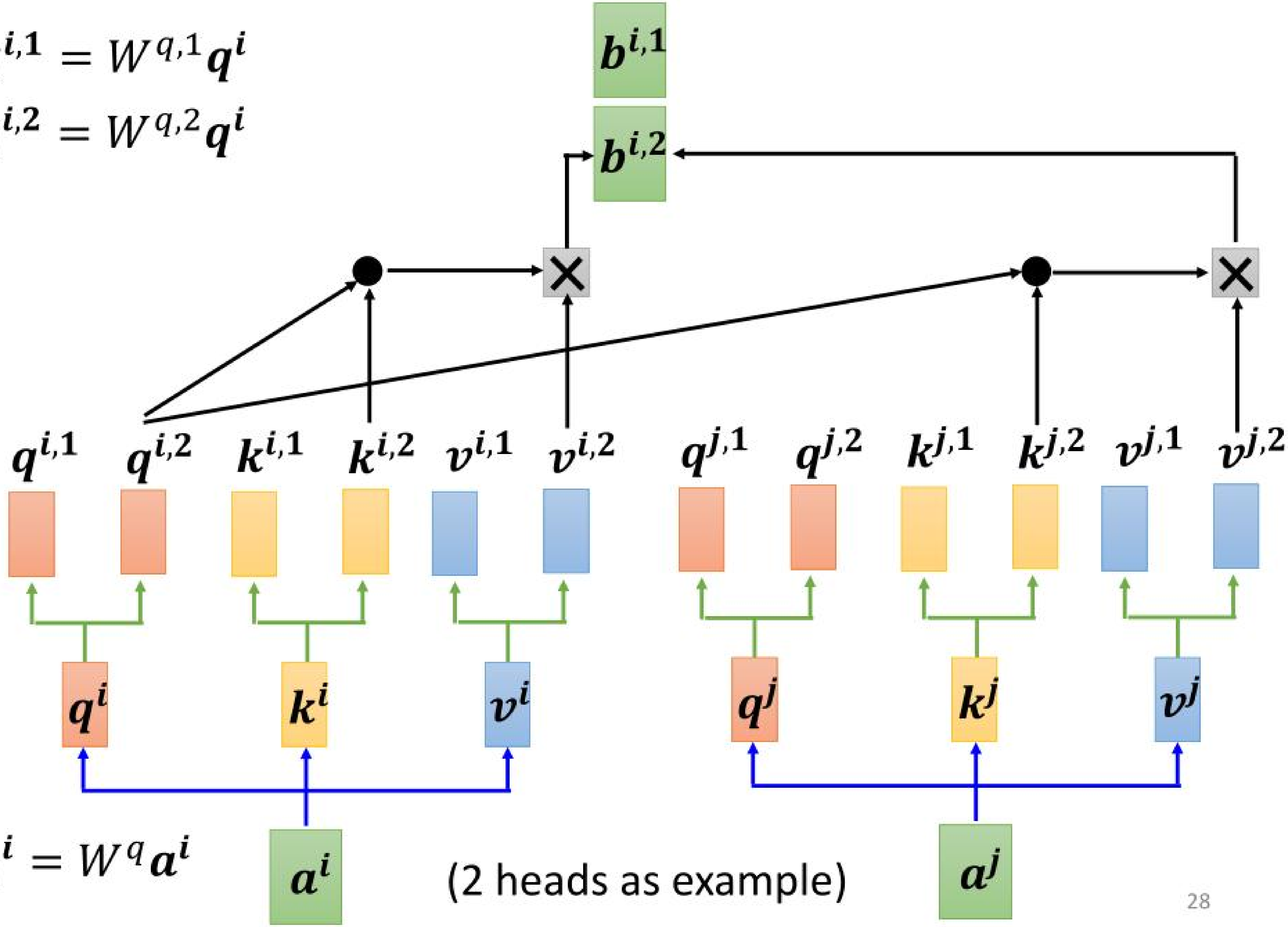


$q^i = W^q a^i$

(2 heads as example)

Multi-head Self-attention Different types of relevance

$$\boldsymbol{q}^{i,1} = W^{q,1}\boldsymbol{q}^{i}$$
$$\boldsymbol{q}^{i,2} = W^{q,2}\boldsymbol{q}^{i}$$



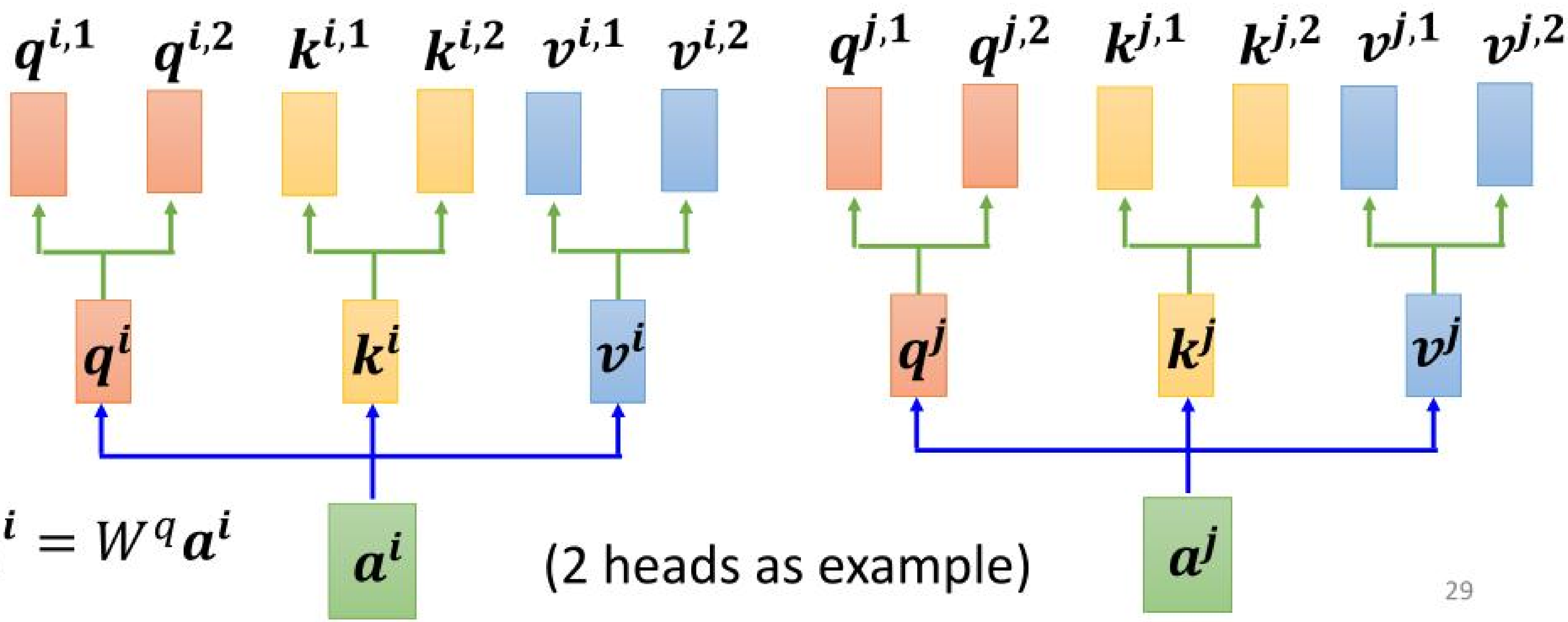
$$\boldsymbol{q}^{i} = W^{q}\boldsymbol{a}^{i}$$

(2 heads as example)

Multi-head Self-attention

Different types of relevance

$$\boldsymbol{b}^{i} = W^{O} \begin{bmatrix} \boldsymbol{b}^{i,1} \\ \boldsymbol{b}^{i,2} \end{bmatrix}$$

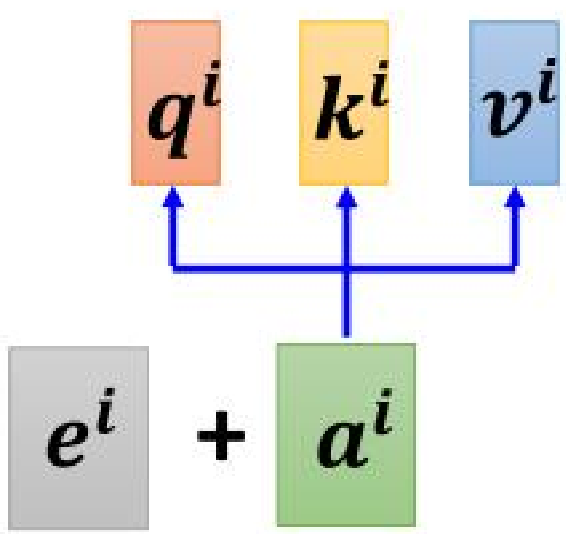


$$\boldsymbol{q}^{i} = W^{q} \boldsymbol{a}^{i}$$

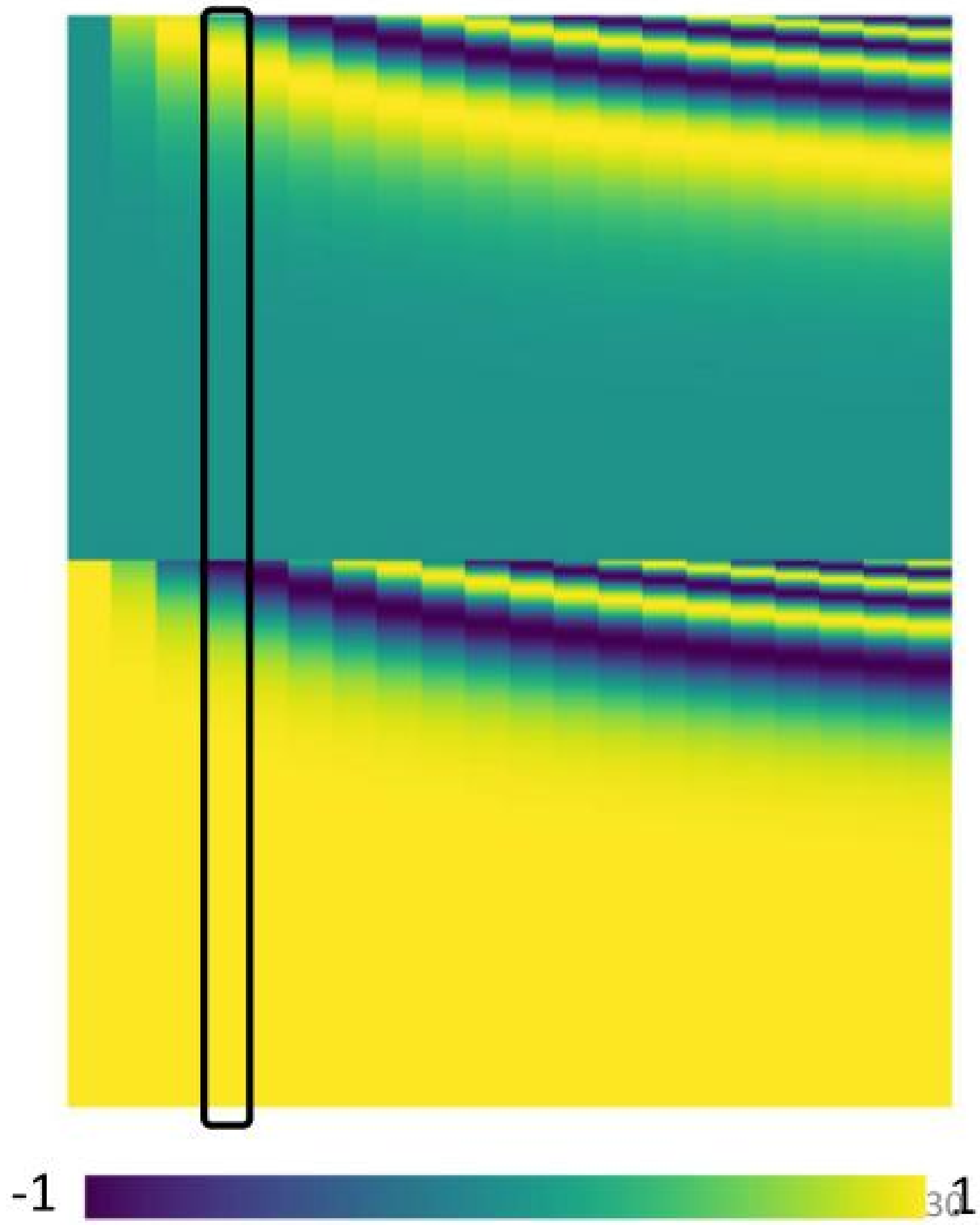
(2 heads as example)

Positional Encoding

- No position information in self-attention.
- Each position has a unique positional vector e^i
- **hand-crafted**
- **learned from data**



Each column represents a positional vector e^i



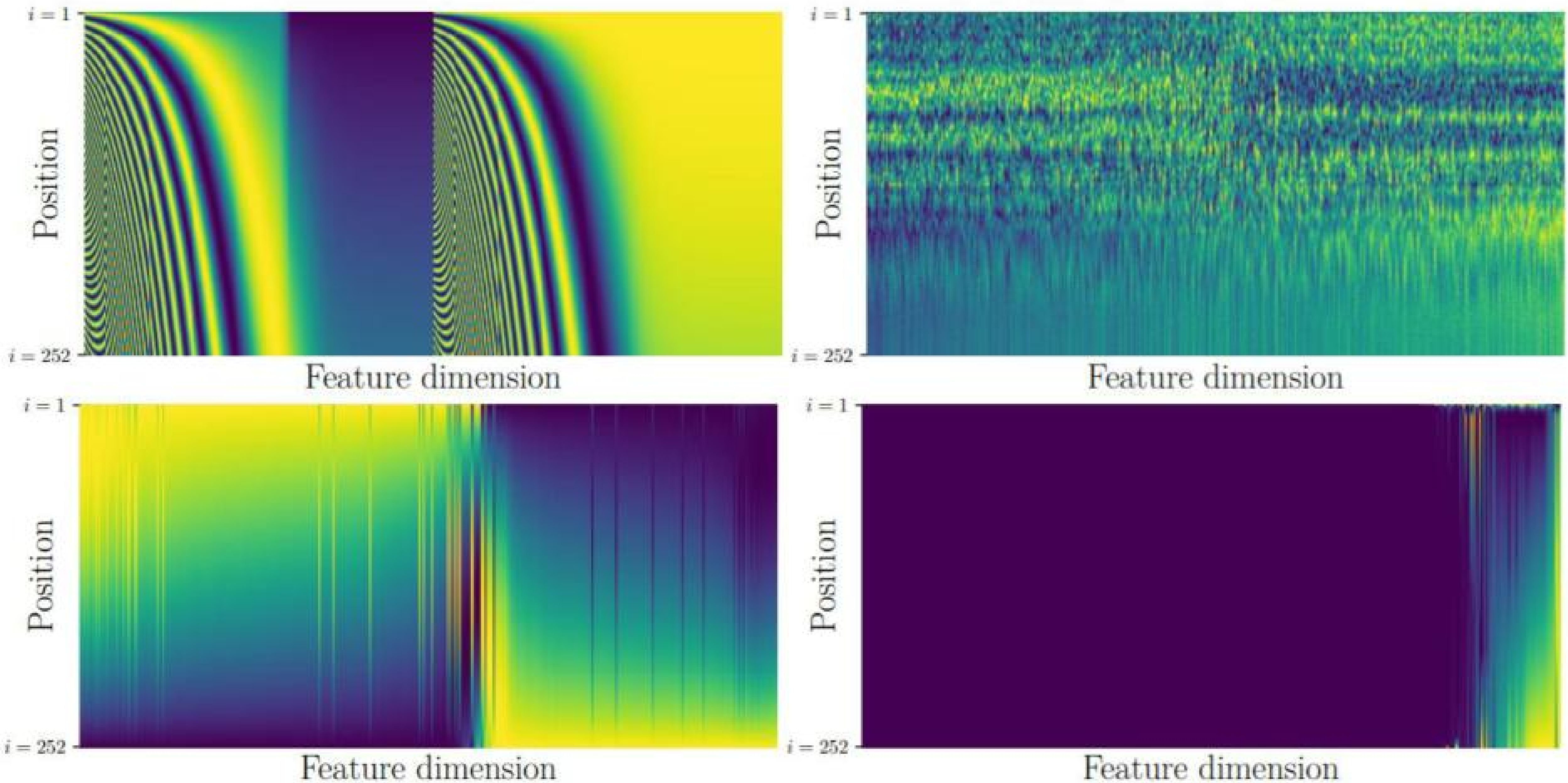
https://arxiv.org/abs/2003.09229

Table 1. Comparing position representation methods

Methods	Inductive	Data-Driven	Parameter Efficient
Sinusoidal (Vaswani et al., 2017)	✓	✗	✓
Embedding (Devlin et al., 2018)	✗	✓	✗
Relative (Shaw et al., 2018)	✗	✓	✓
This paper	✓	✓	✓

(a) Sinusoidal

(b) Position embedding



(c) FLOATER

(d) RNN

Many applications …

Transformer

https://arxiv.org/abs/1706.03762

BERT

https://arxiv.org/abs/1810.04805

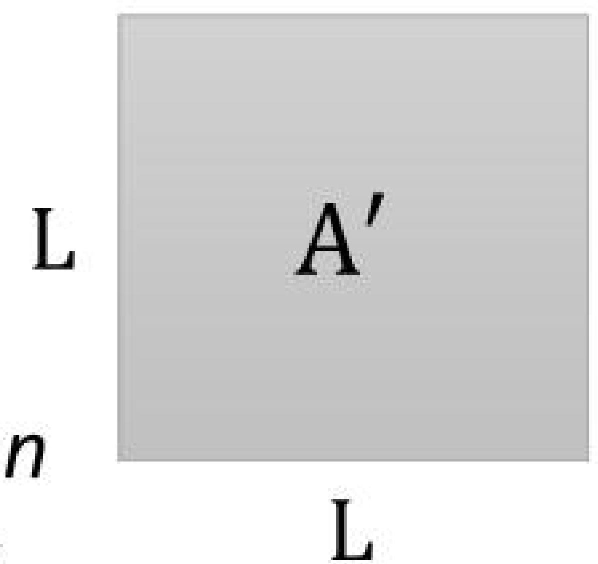
Widely used in Natural Langue Processing (NLP)!

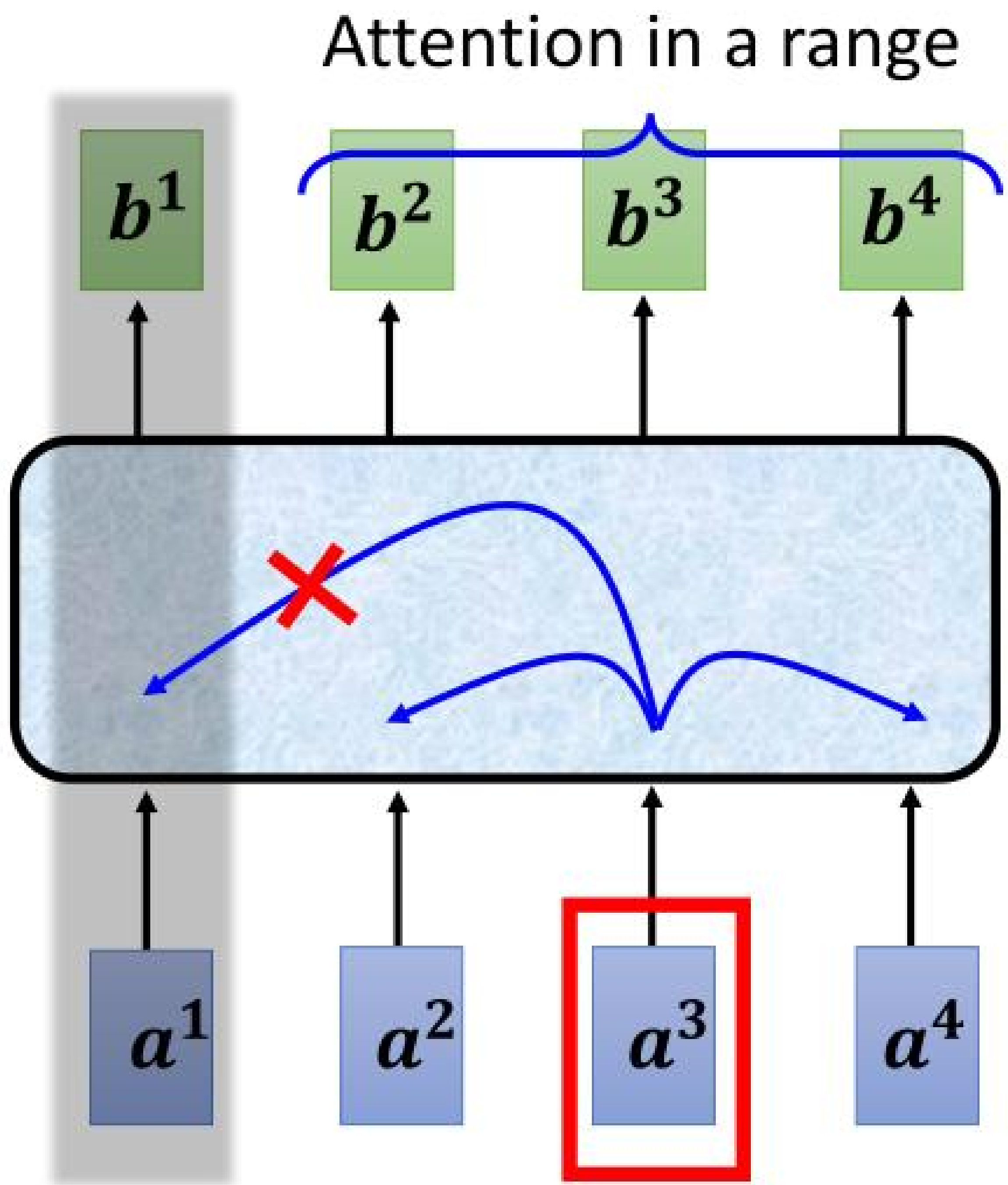
https://arxiv.org/abs/1910.12977

Self-attention for Speech

Speech is a very long vector sequence.

If input sequence is length L

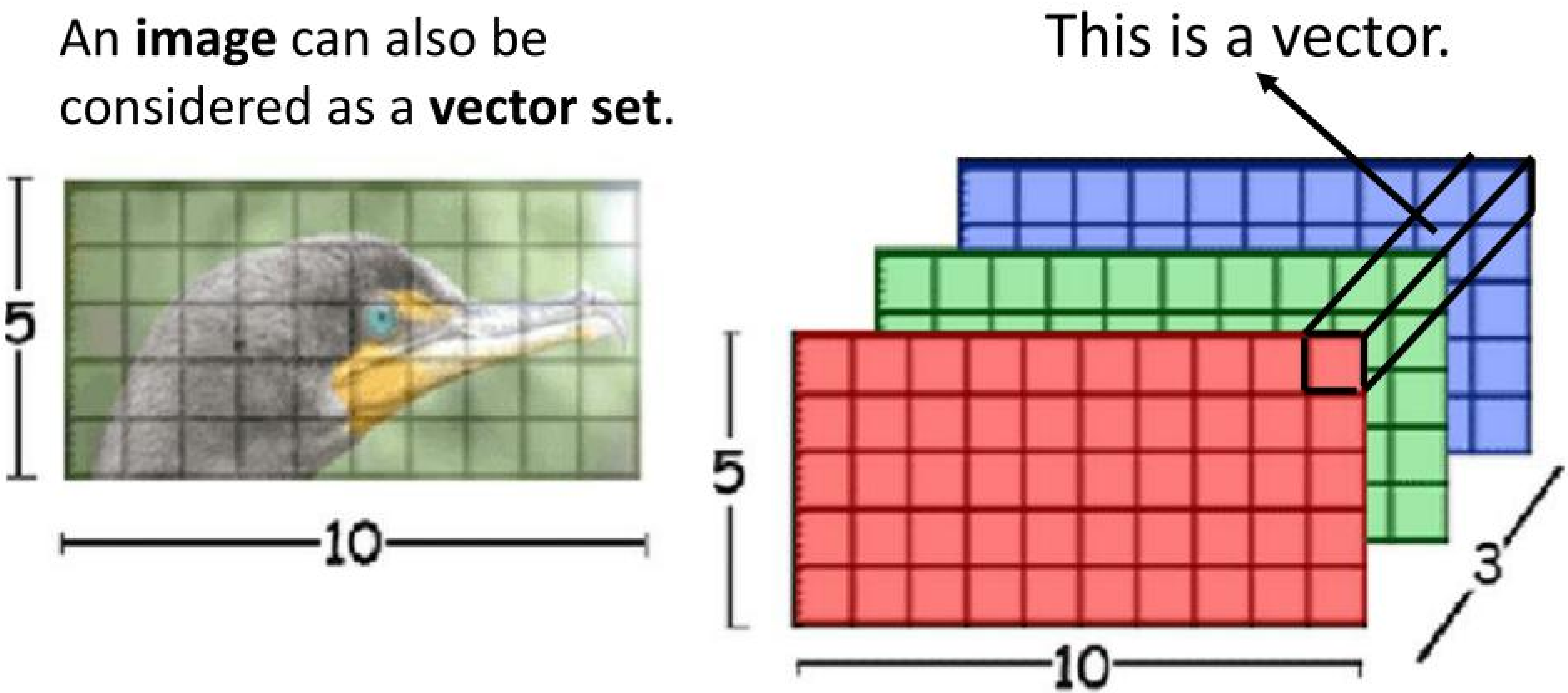




Truncated Self-attention

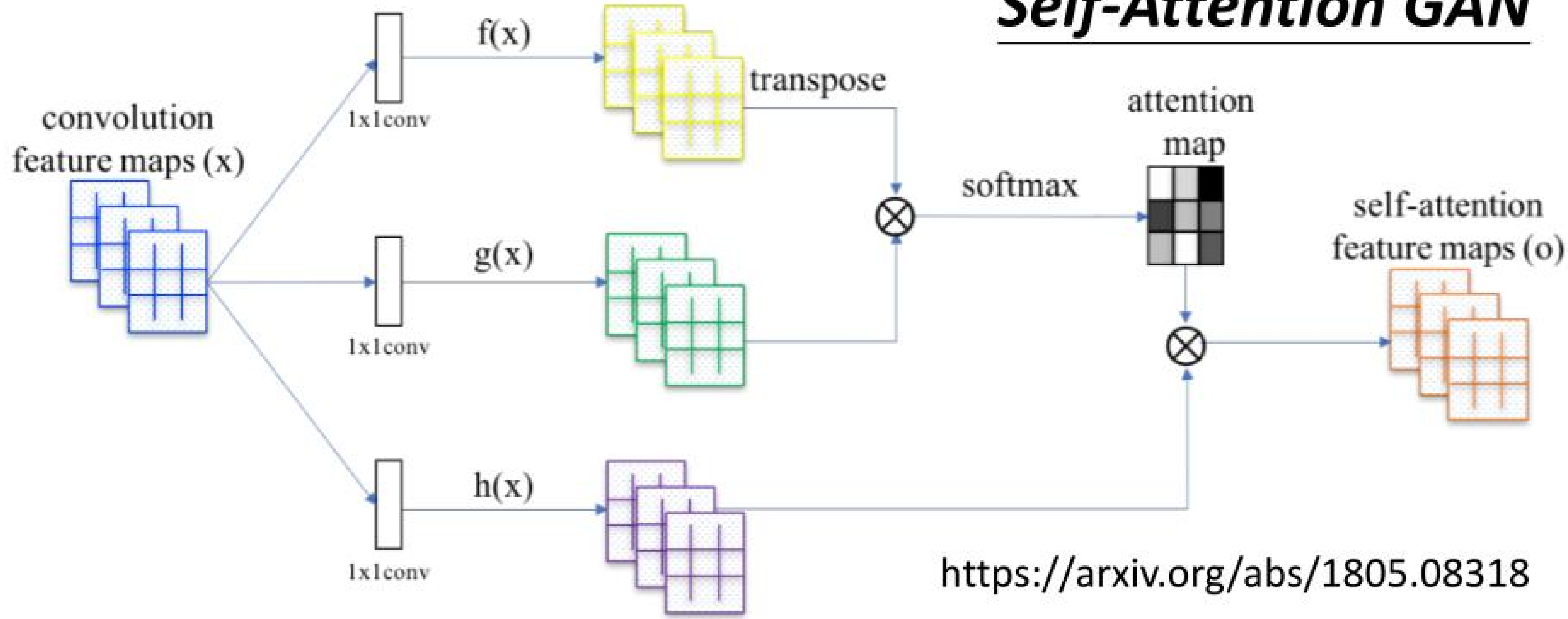
Self-attention for Image

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



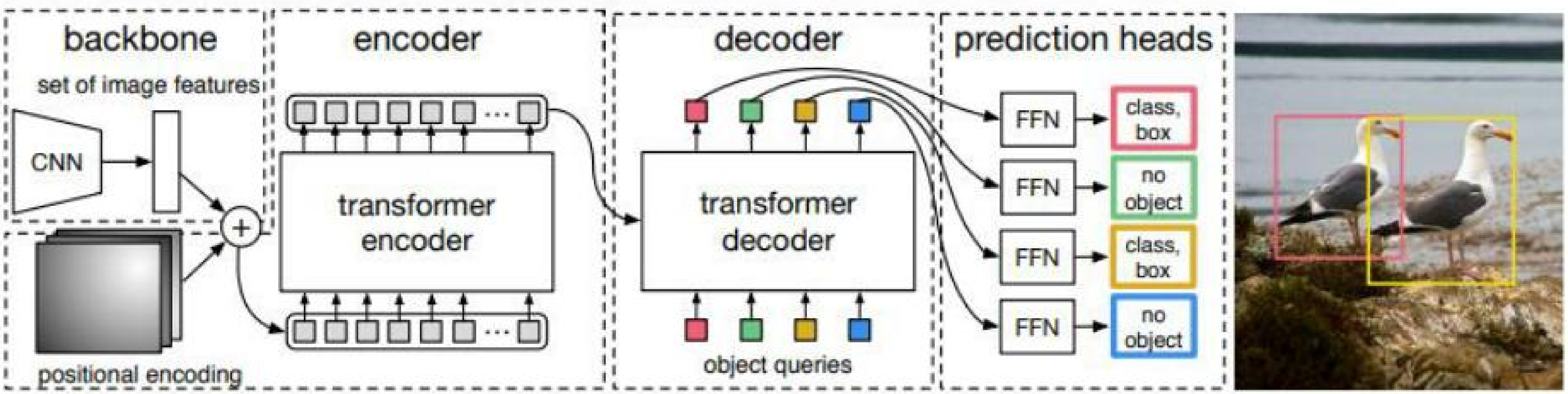
Source of image: https://www.researchgate.net/figure/Color-image-representation-and-RGB-matrix_fig15_282798184

Self-Attention GAN



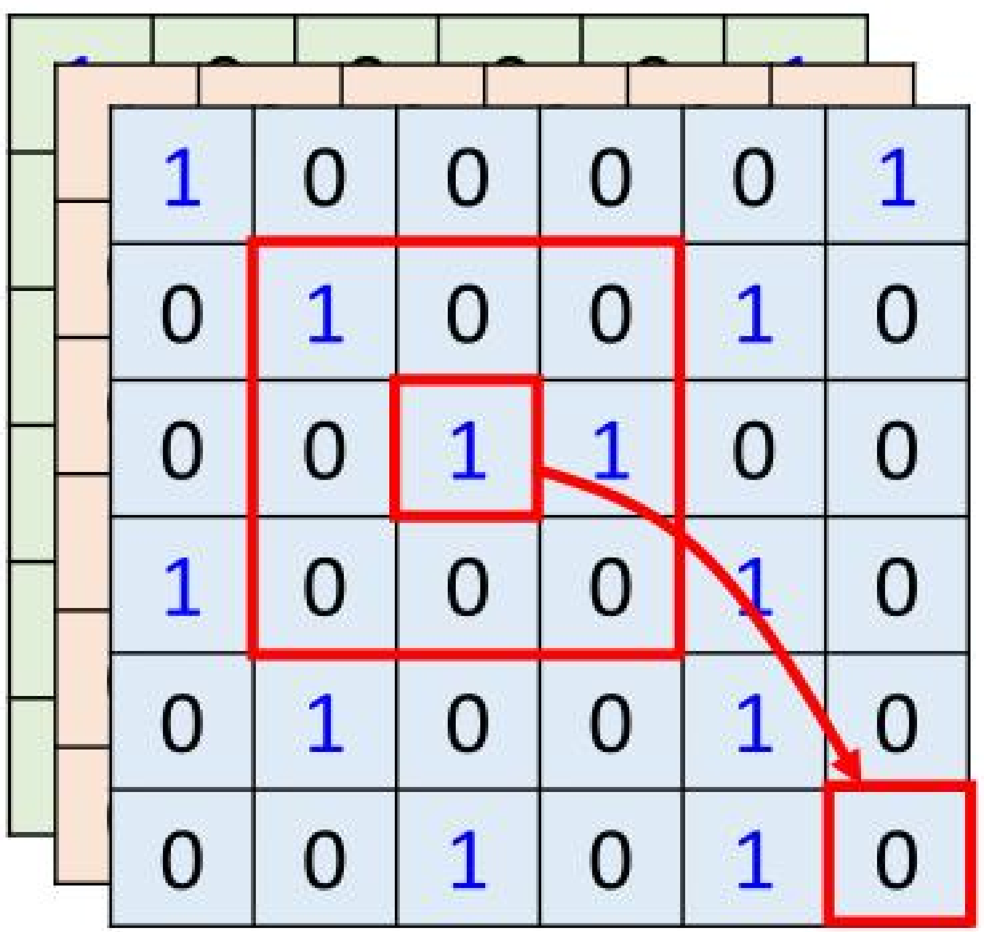
https://arxiv.org/abs/1805.08318

DEtection Transformer (DETR)



https://arxiv.org/abs/2005.12872

Self-attention v.s. CNN



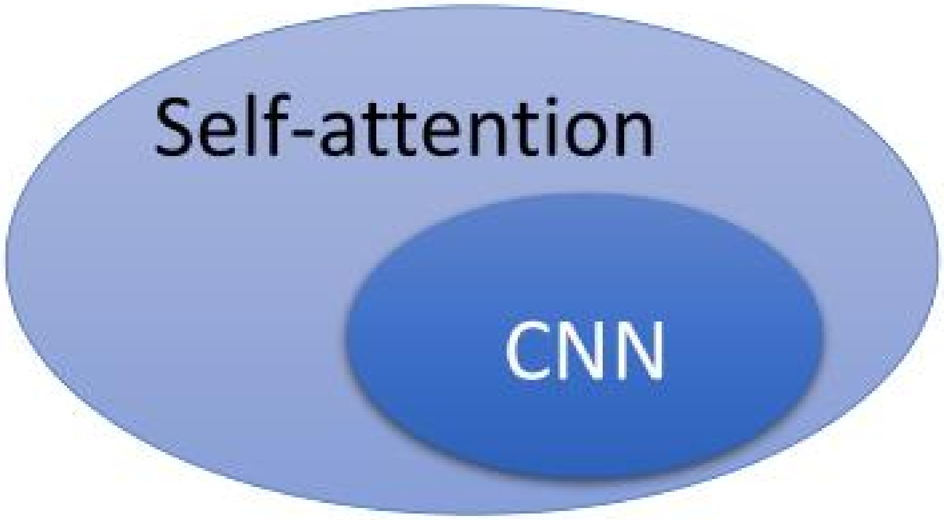
CNN: self-attention that can only attends in a receptive field

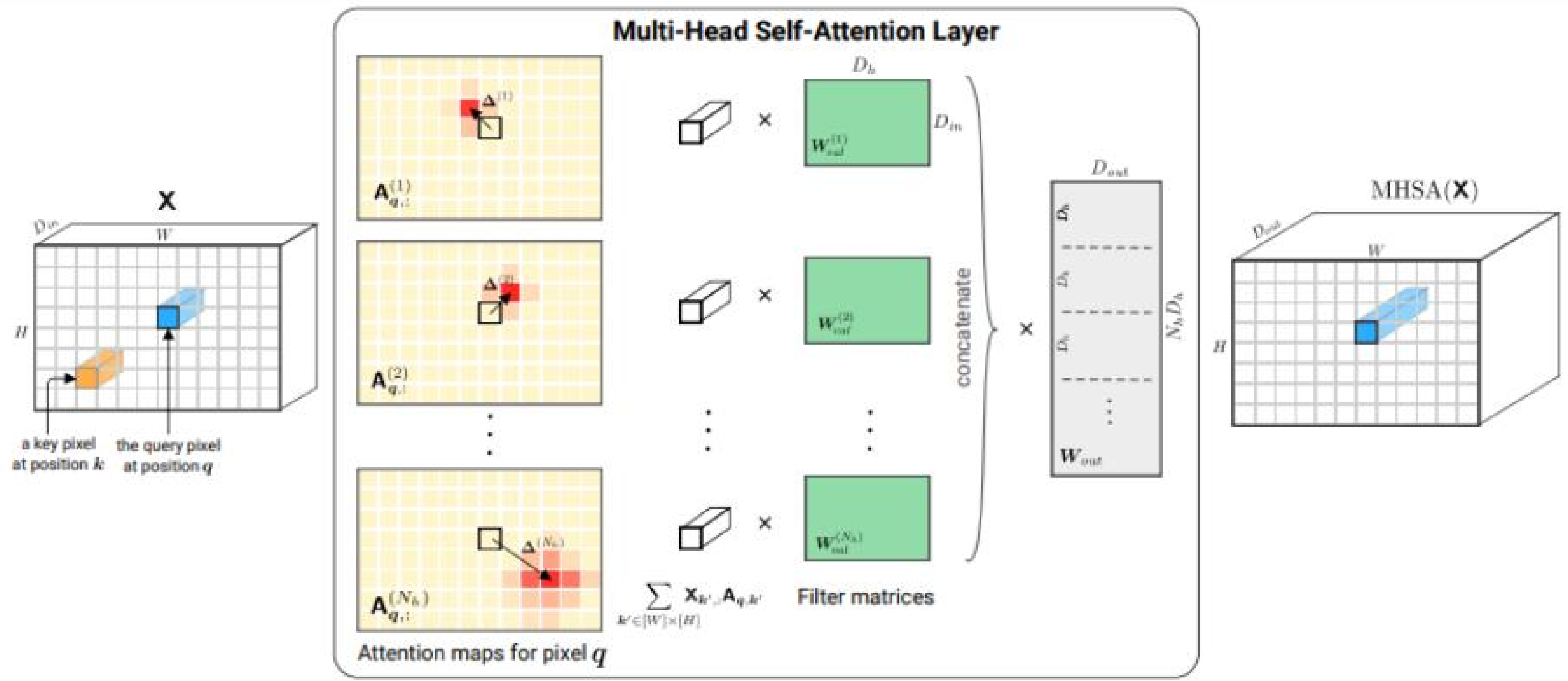
- ➢ CNN is simplified self-attention.

Self-attention: CNN with learnable receptive field

- ➢ Self-attention is the complex version of CNN.

Self-attention v.s. CNN

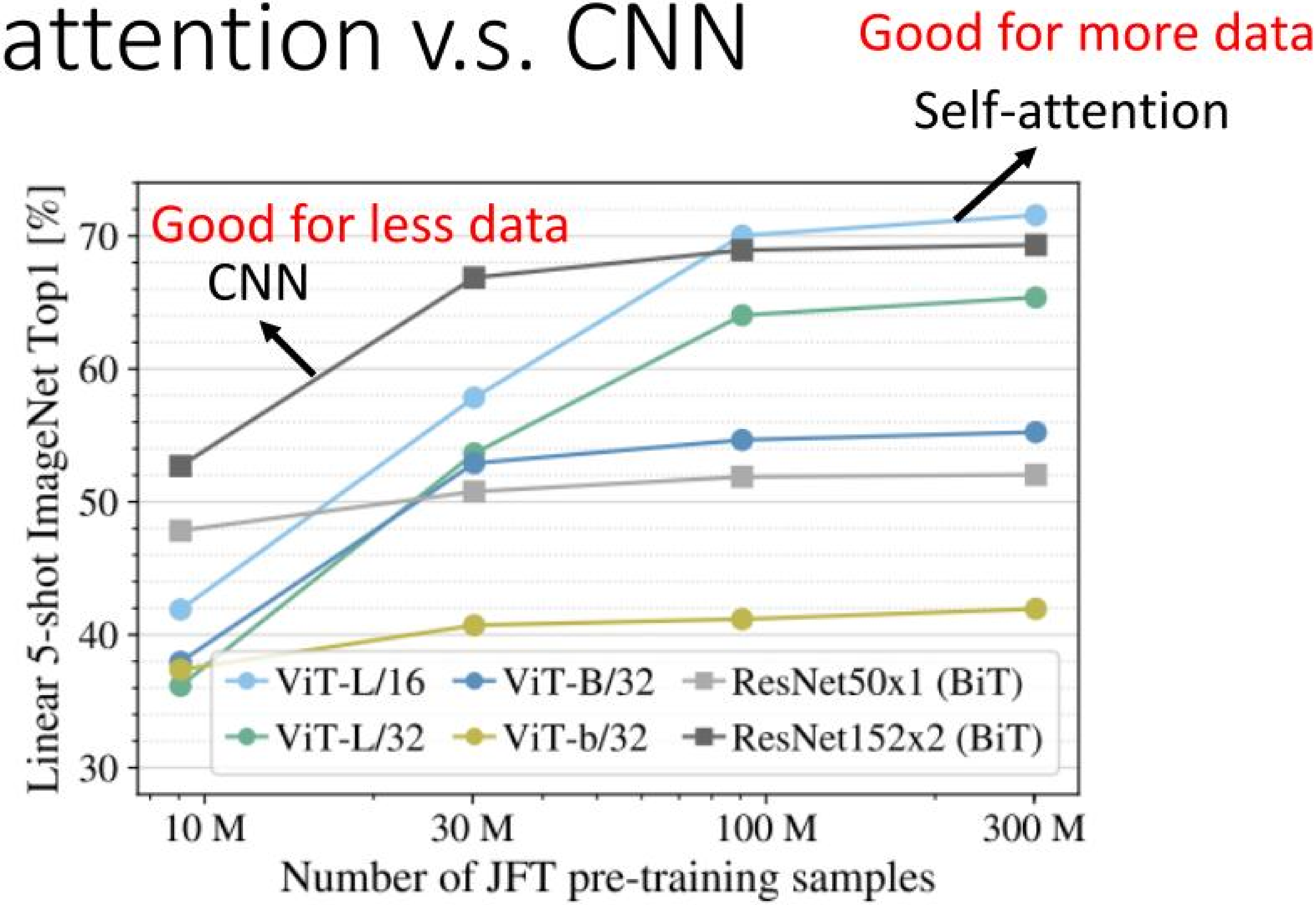




On the Relationship between Self-Attention and Convolutional Layers

https://arxiv.org/abs/1911.03584

Self-attention v.s. CNN

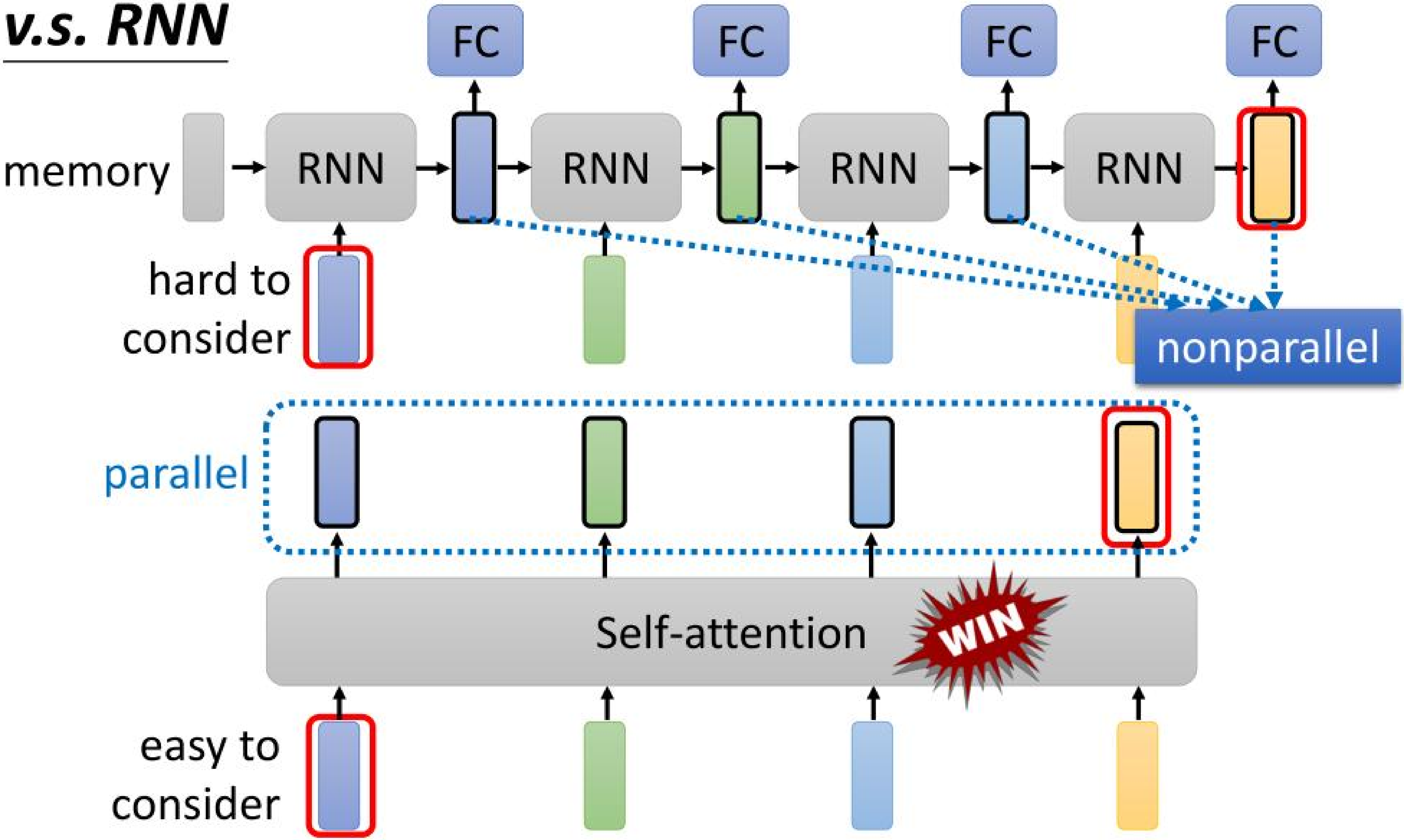


An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

https://arxiv.org/pdf/2010.11929.pdf

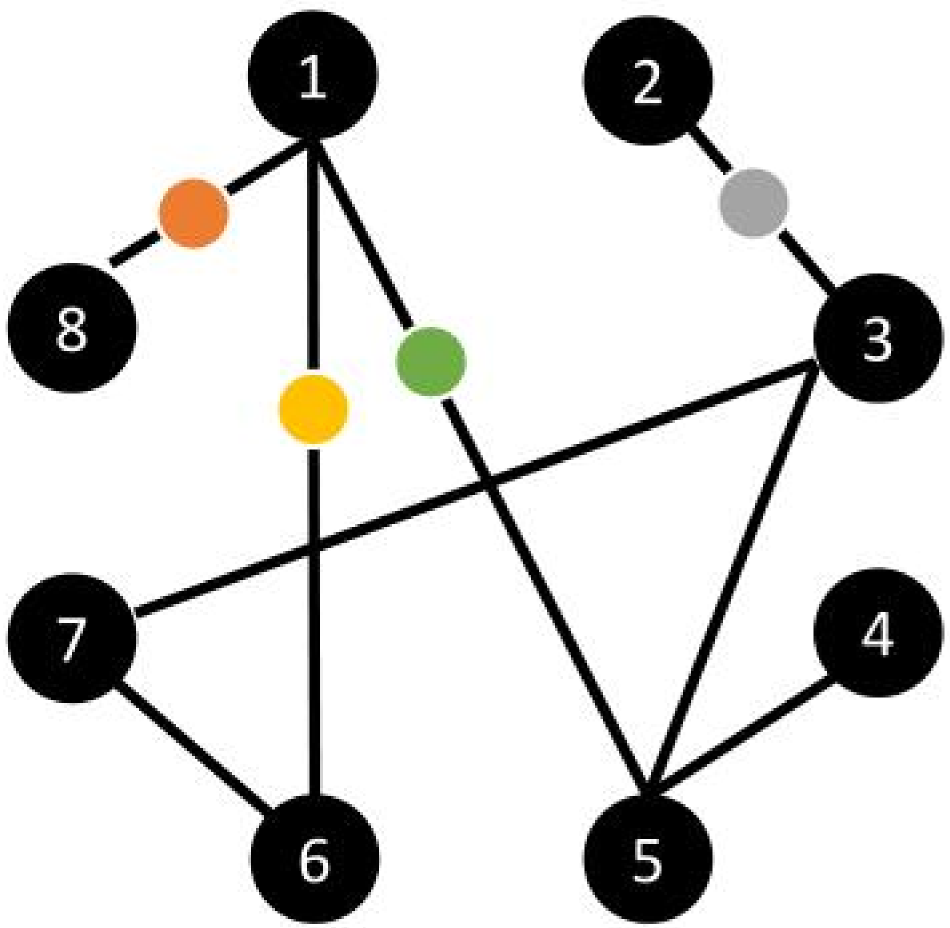
Self-attention v.s. RNN

Recurrent Neural Network (RNN)



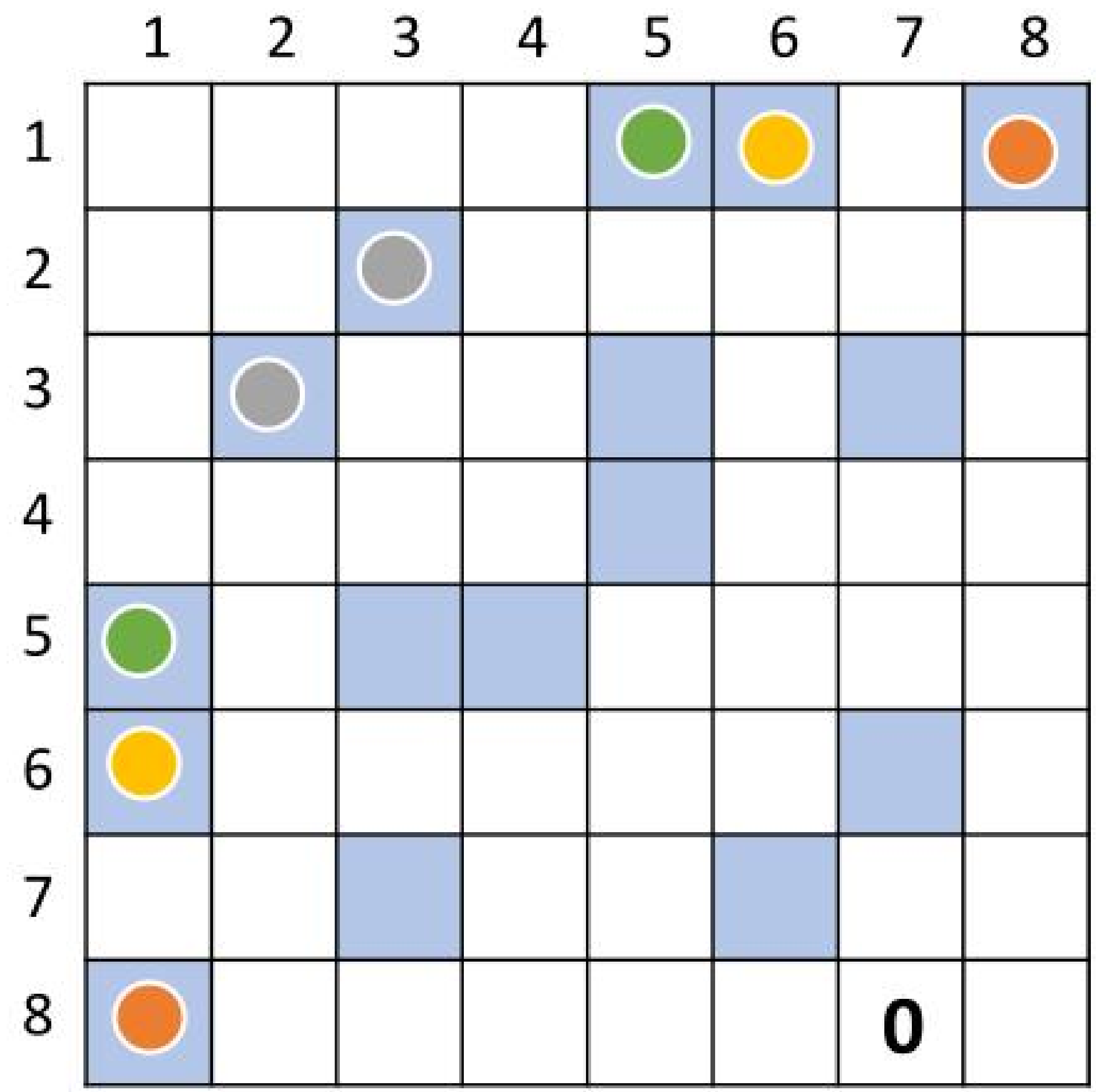
Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention https://arxiv.org/abs/2006.16236

Self-attention for Graph



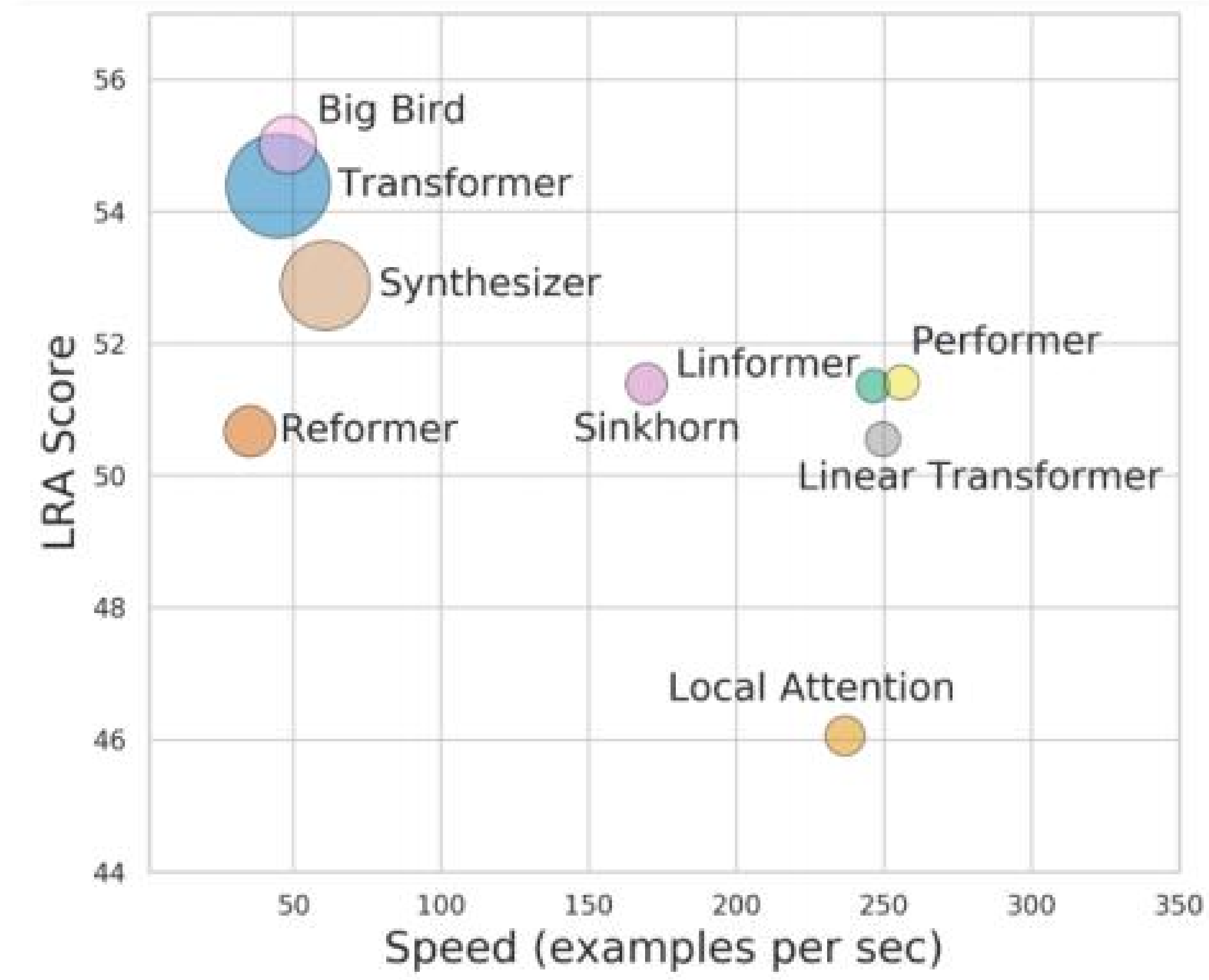
Consider **edge**: only attention to connected nodes

Attention Matrix



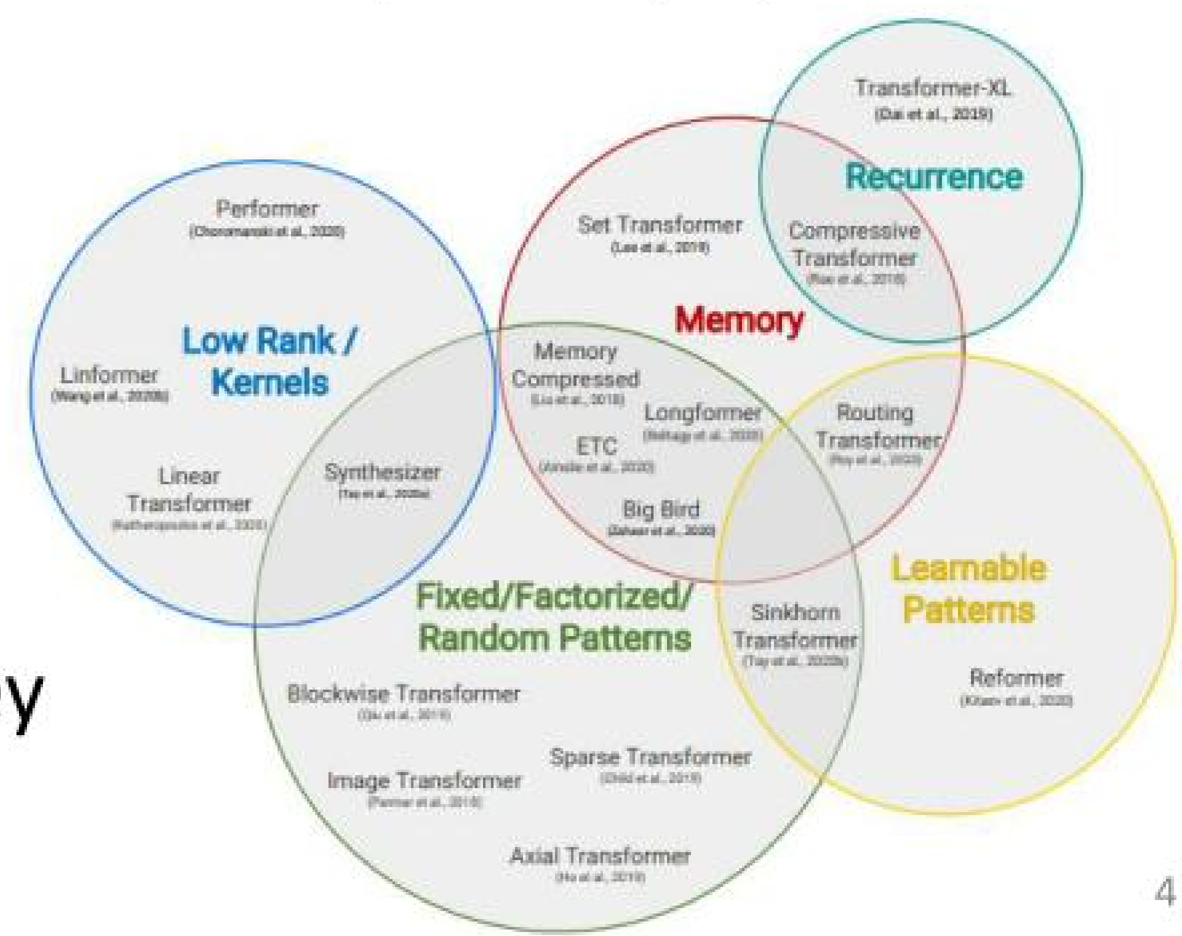
This is one type of **Graph Neural Network (GNN)**.

To Learn More …



Long Range Arena: A Benchmark for Efficient Transformers

https://arxiv.org/abs/2011.04006



Efficient Transformers: A Survey

https://arxiv.org/abs/2009.06732