Institute of Electronics
National Yang Ming Chiao Tung University
Hsinchu, Taiwan

AI Training Course Series

Introduction to Self-Attention & Transformer

Lecture 6

Distinguished Professor
Juinn-Dar Huang, Ph.D.
jdhuang@nycu.edu.tw

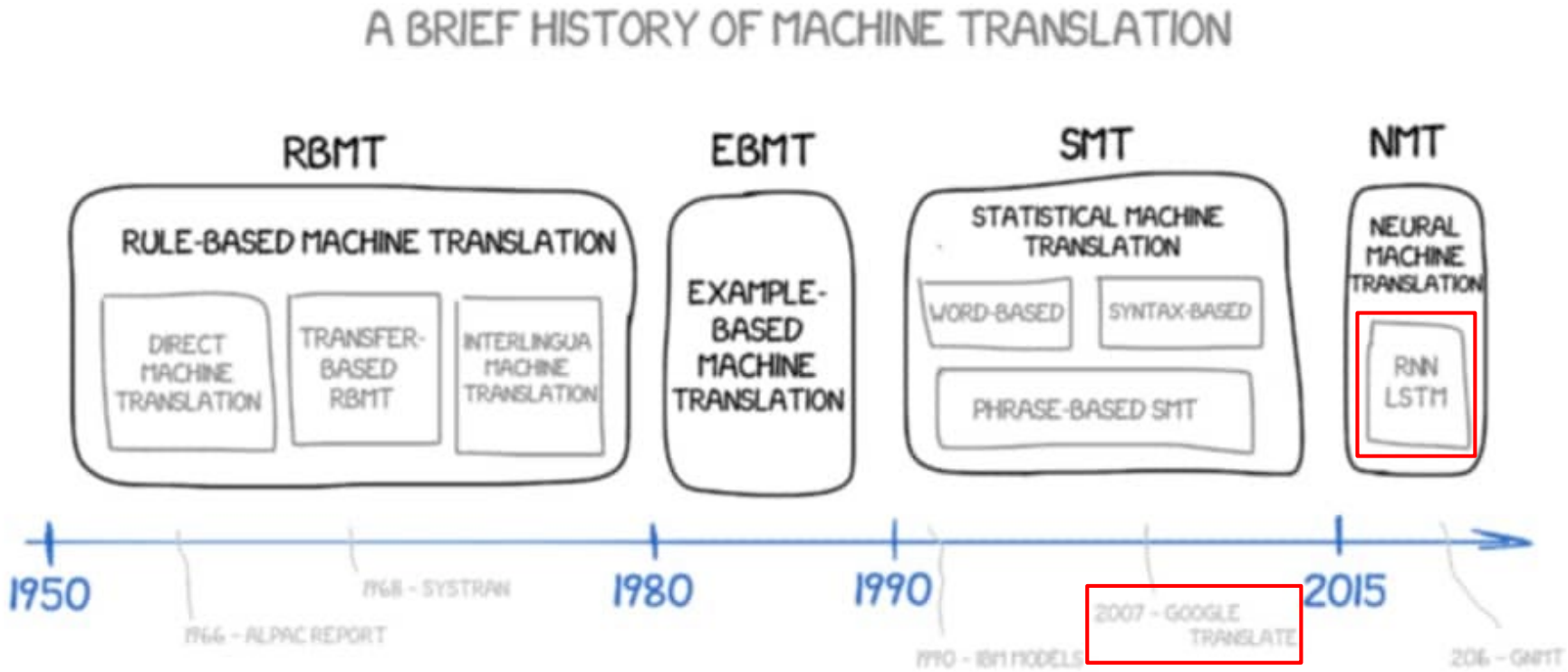
July 23, 2024

Outline

- Machine translation
- Self-attention
  - core ideas
  - parallel computation
  - multi-head self-attention (MSA)
  - positional encoding
- Transformer architecture
  - encoder
  - decoder
- Transformer for image classification
- Summary and references

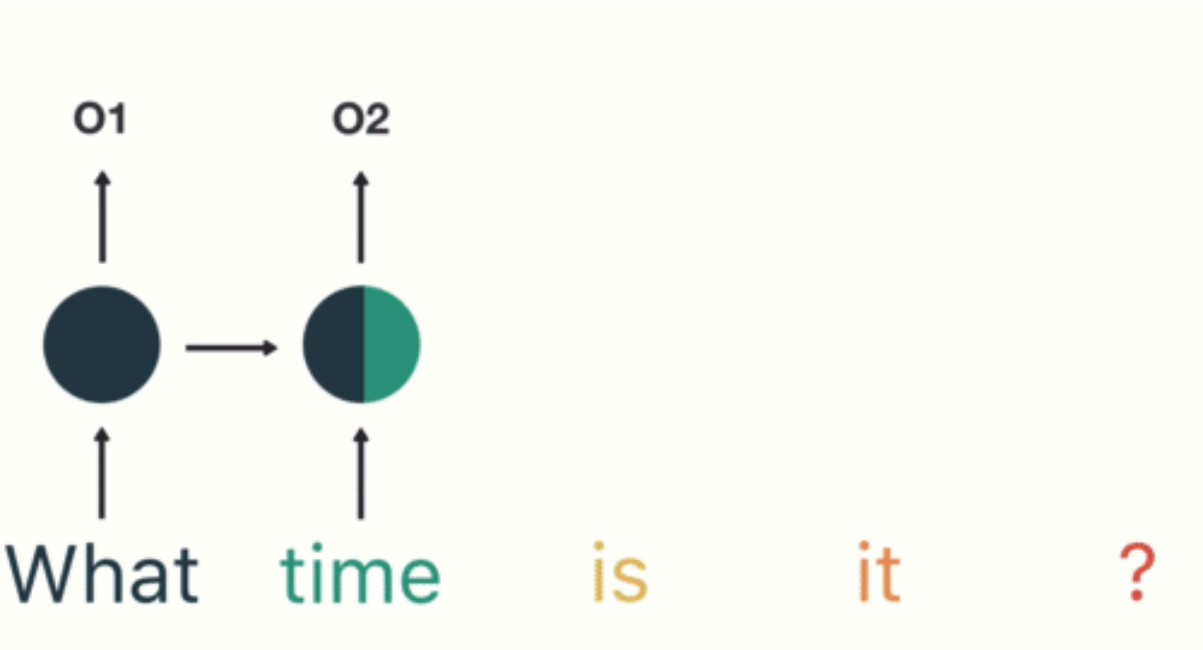
Machine Translation (MT)

History of Machine Translation

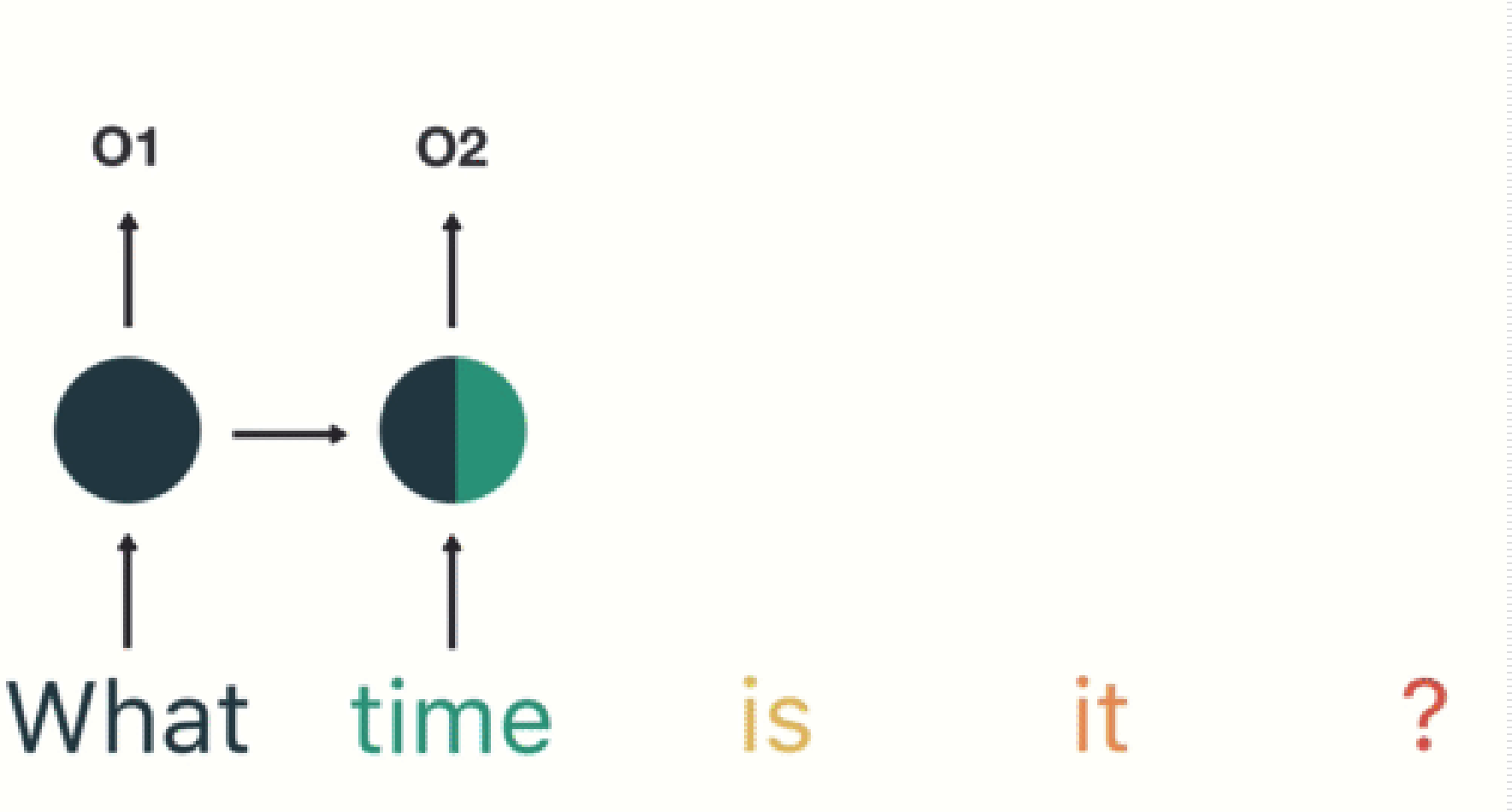


Sequence-to-Sequence (seq2seq)

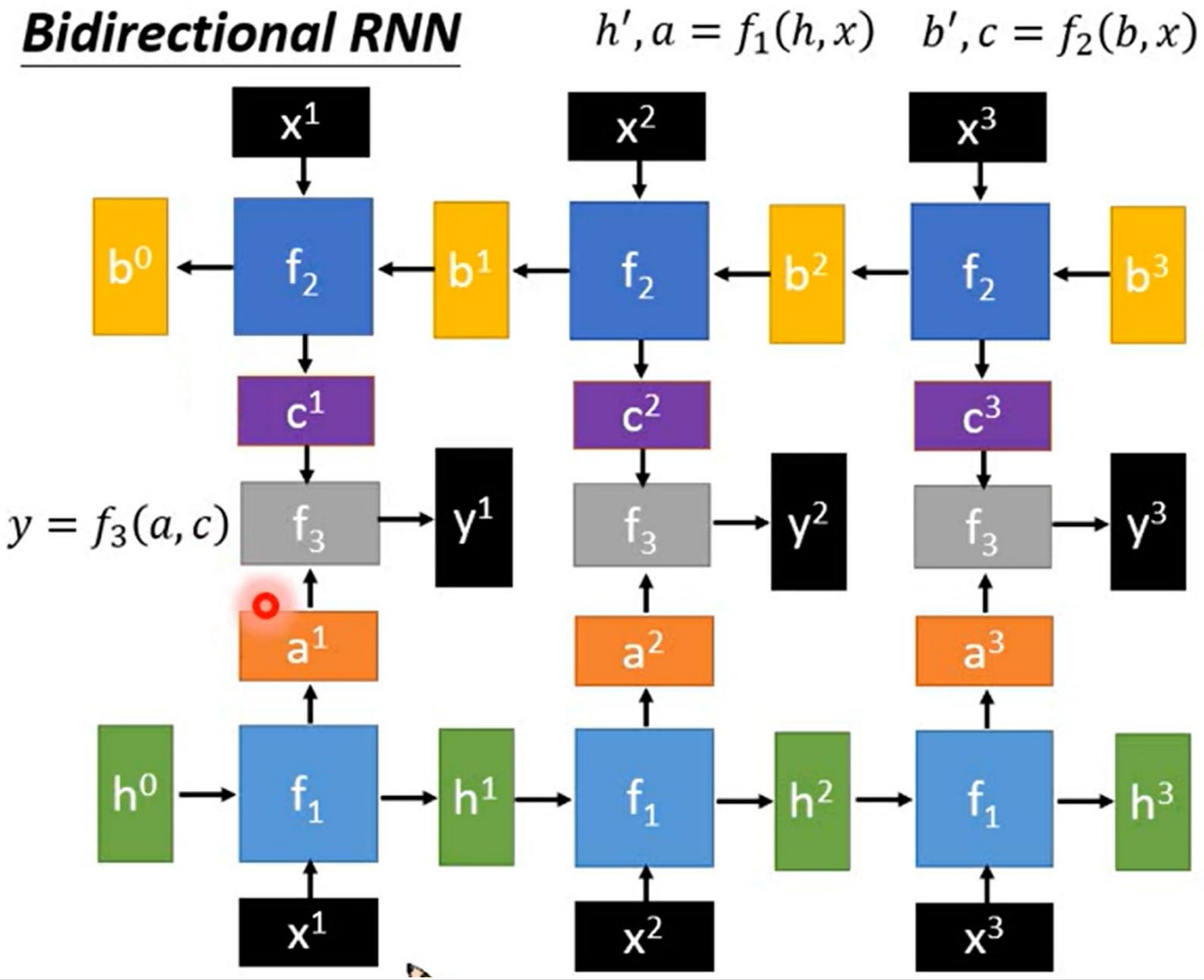
- A sentence: a sequence of words
  - 機器學習 ➔ machine learning
  - sequence ➔ sequence
- Machine translation: sequence to sequence
  - given a sequence, generate a new target sequence



MT by Unidirectional RNNs



MT by Bidirectional RNNs

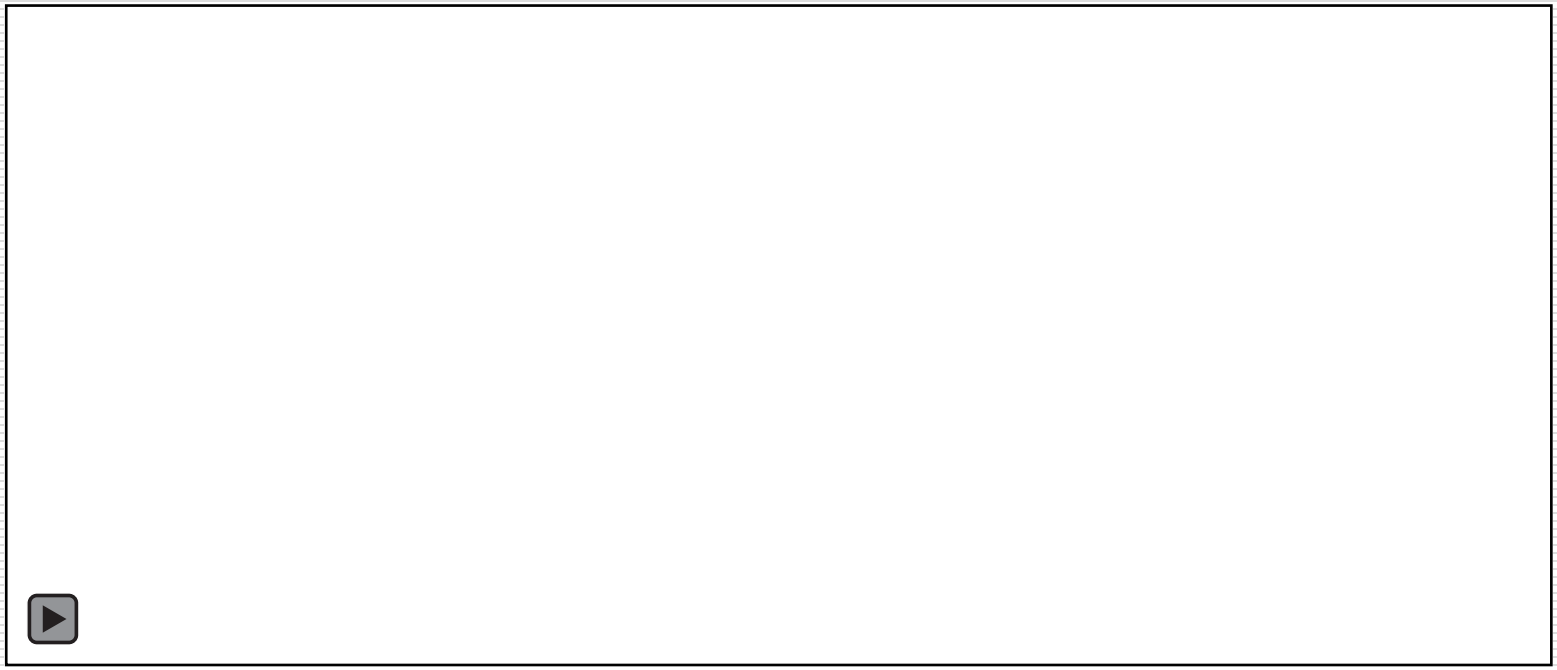


sequential computation ☹

long-range dependency ☹

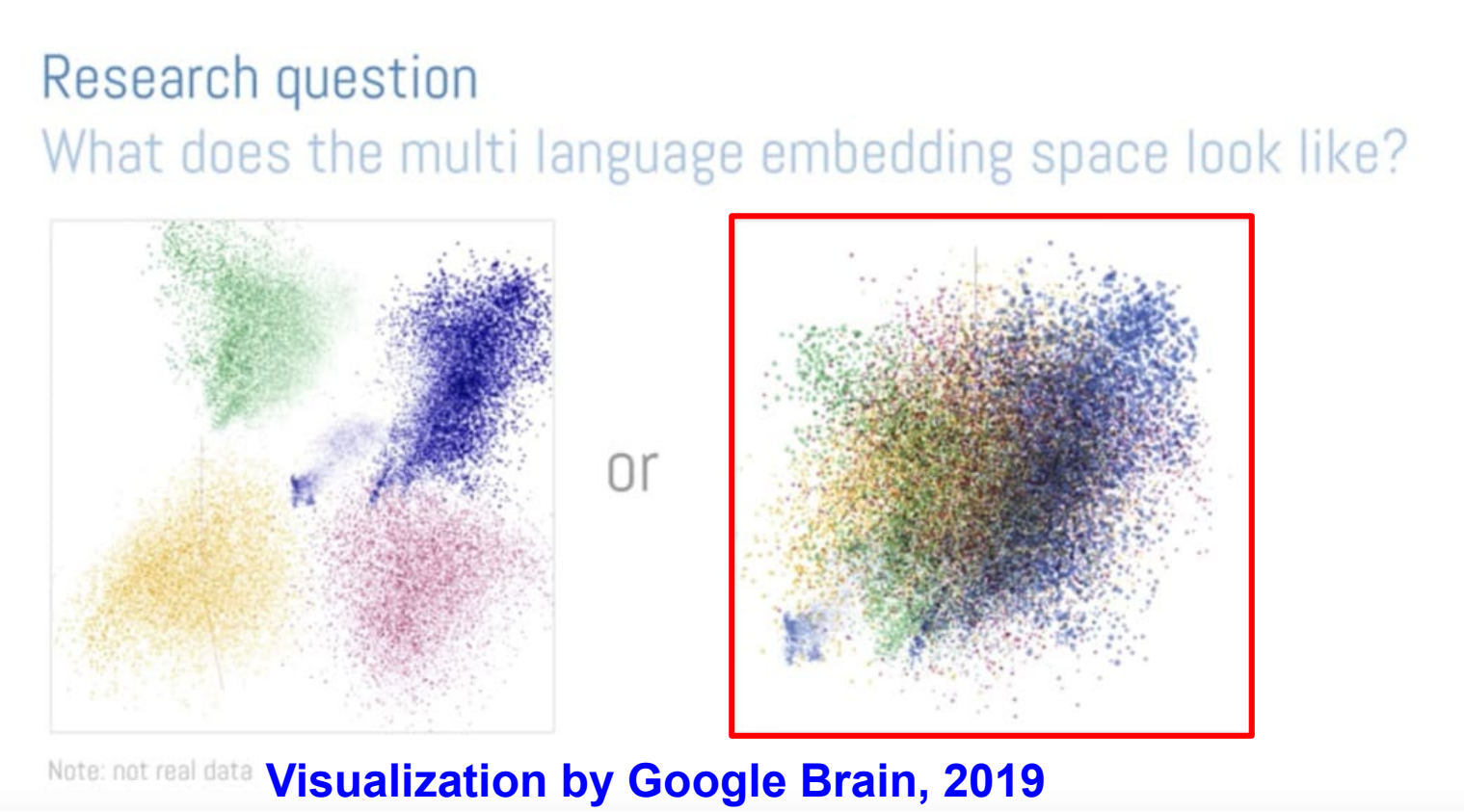
Encoder-Decoder Architecture for MT

- Encoder
  - understand the meaning of an input sequence
  - represented it as vectors in a space of high dimensionality
- Decoder
  - generate an output sequence with the same meaning



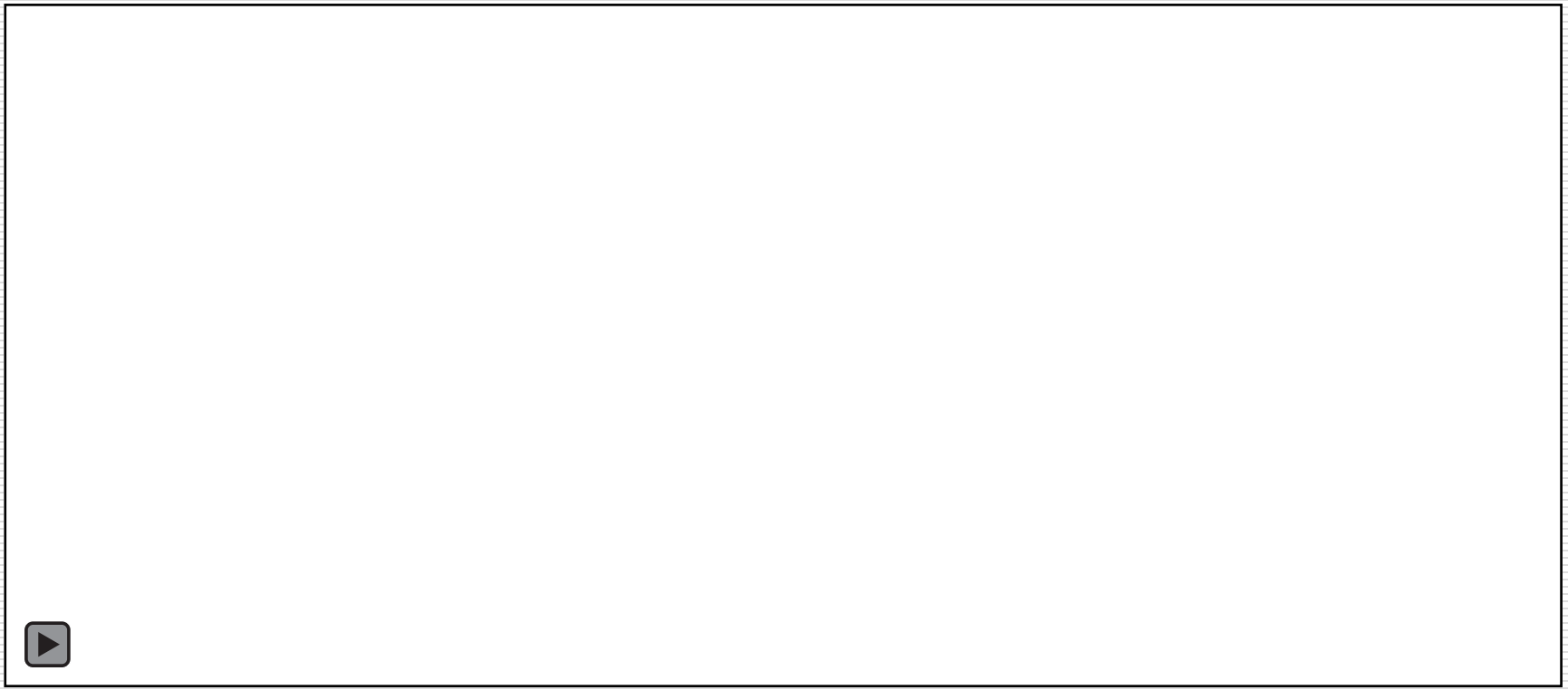
Does Encoder Really Work?

- 語言相異但語義相同的句子會被轉換成彼此距離相近的語義向量

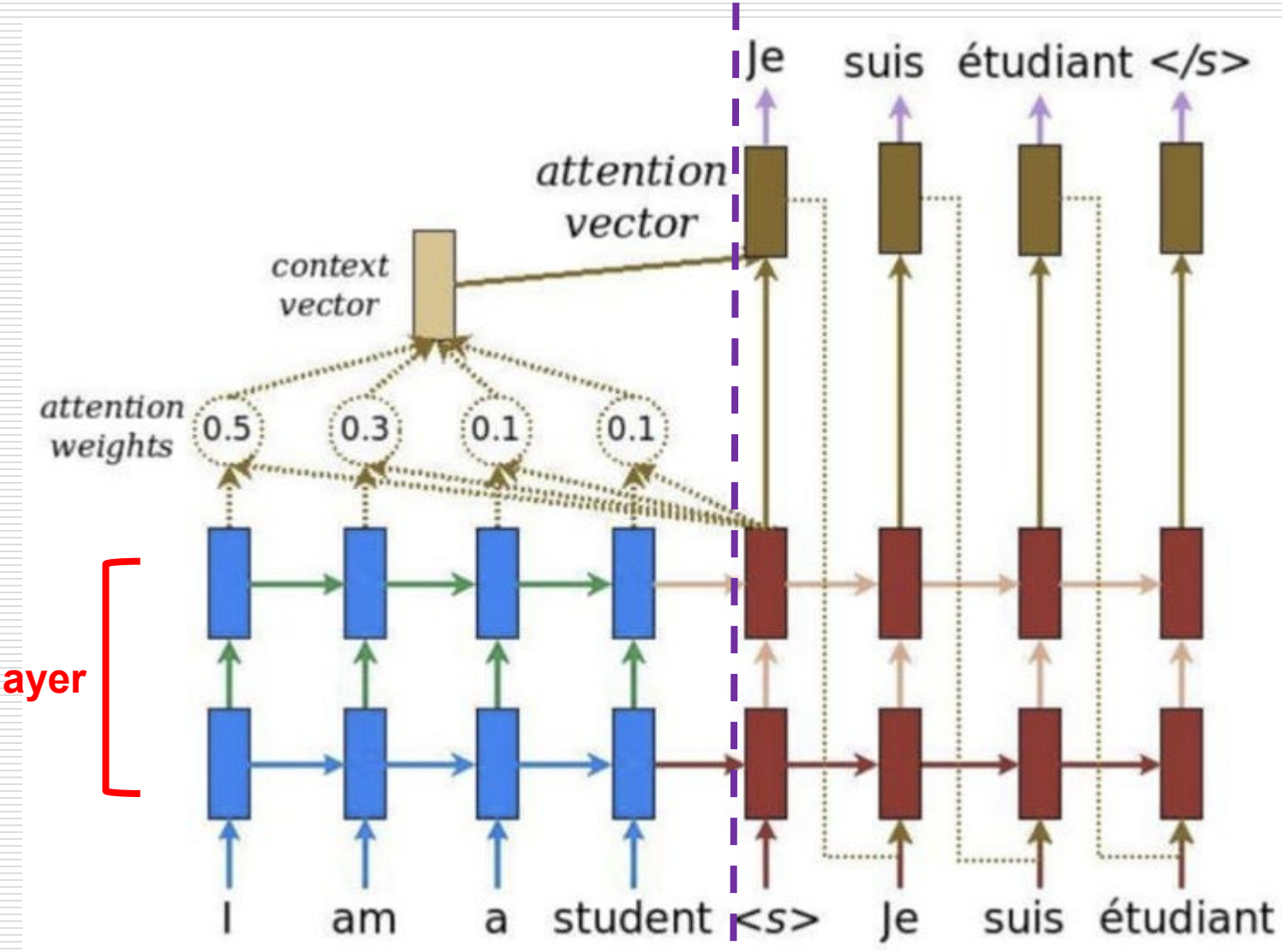


Attention Mechanism

- Provide MORE information to decoder

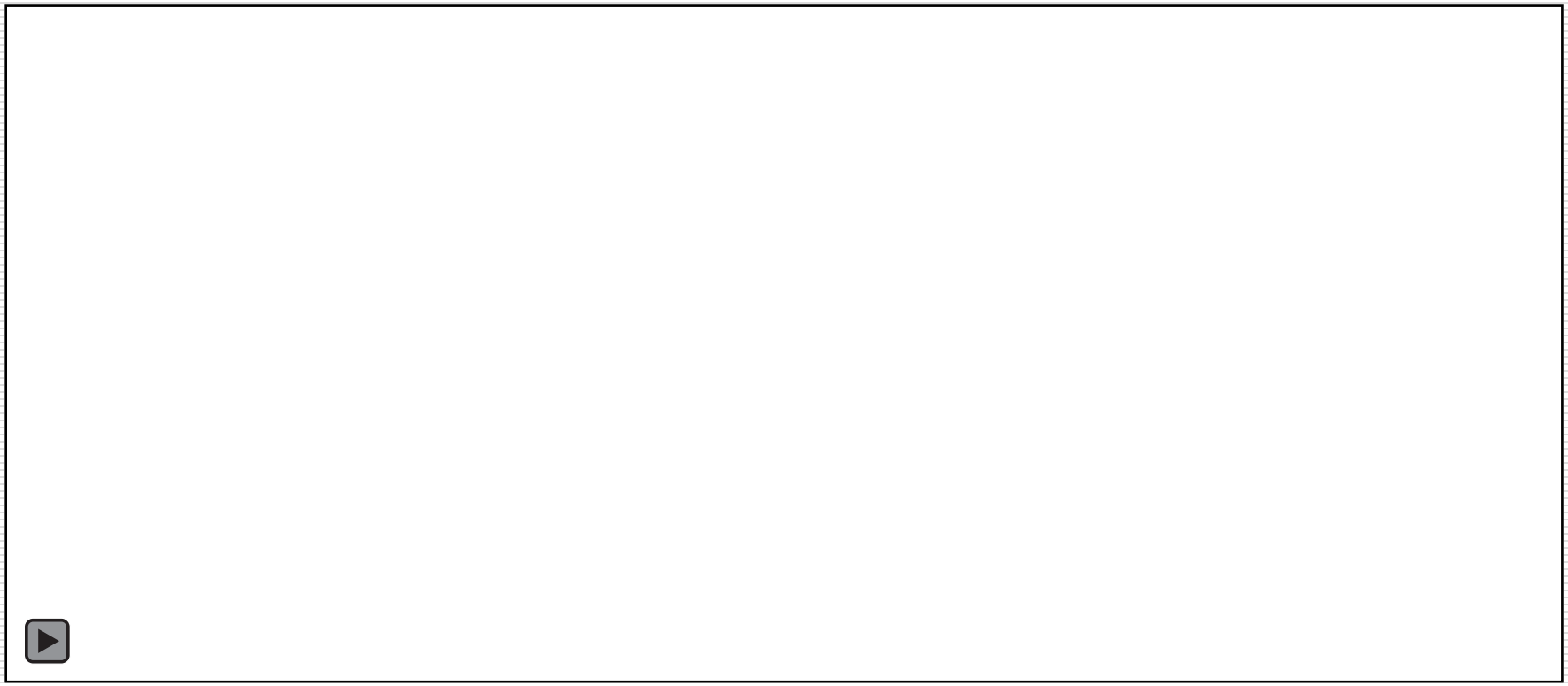


Decoder with Attention (1/3)

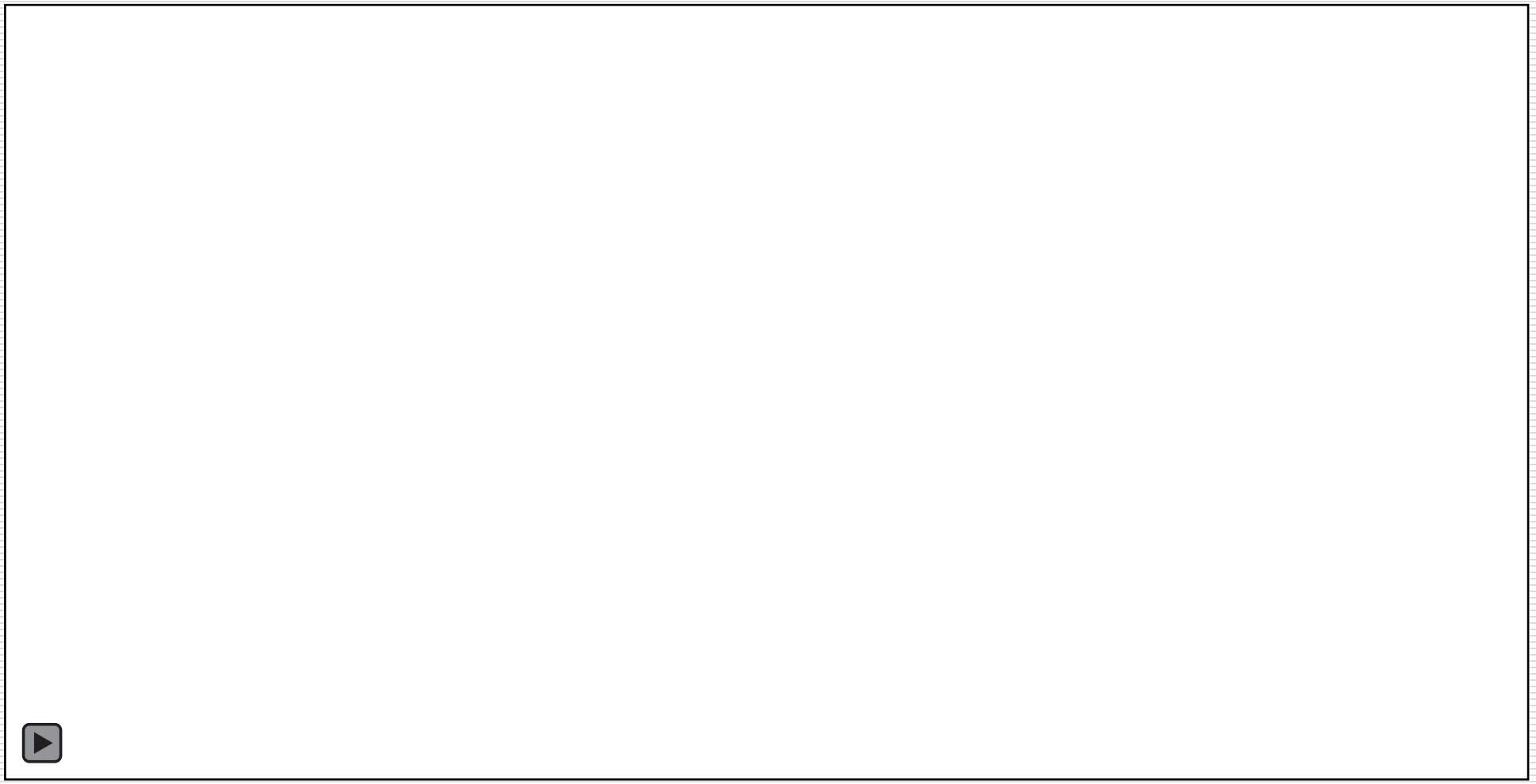


Decoder with Attention (2/3)

- Visualization



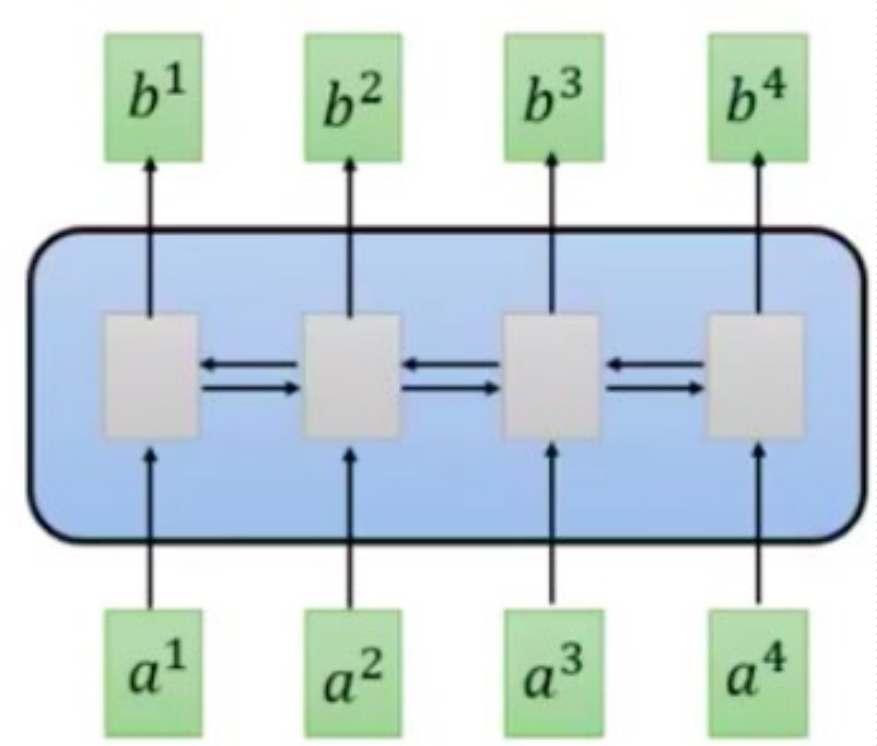
Decoder with Attention (3/3)



Self-Attention

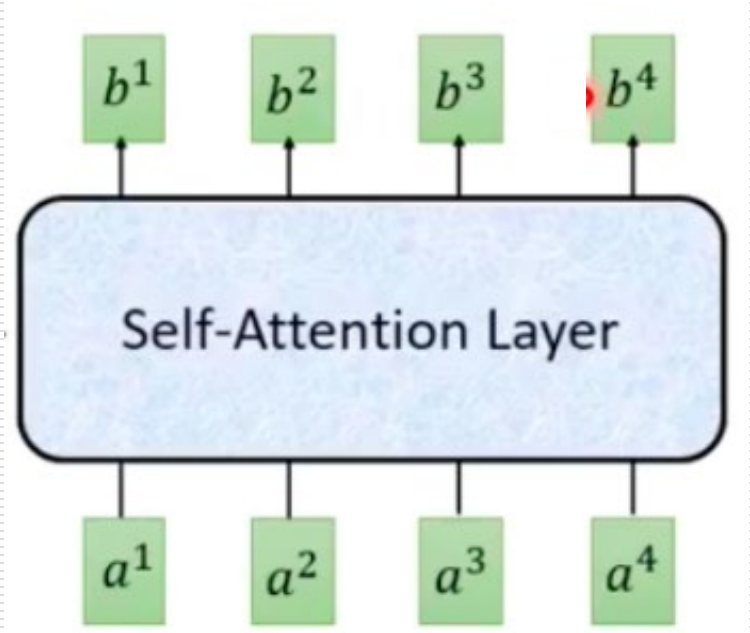
Issues of RNN-Based Implementations

- Problems of RNNs (GRU, LSTM, …)
  - the computation is intrinsically sequential
    ➔ **CANNOT be parallelized** (i.e., **accelerated**) ☹
  - virtually unable to capture relevance between **words far far away** ☹



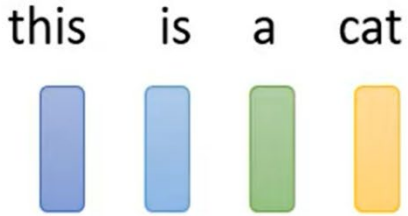
Self-Attention Mechanism

- Self-Attention Layer
  - examine all input **vectors** at the same time (~~near? far?~~)
  - b^i is generated based on the whole input **vectors**
  - perform computation **in parallel**
    ➔ can be effectively accelerated in GPUs



Word Embedding

Vector Set as Input



X ***One-hot Encoding***

apple = [1 0 0 0 0 .....]

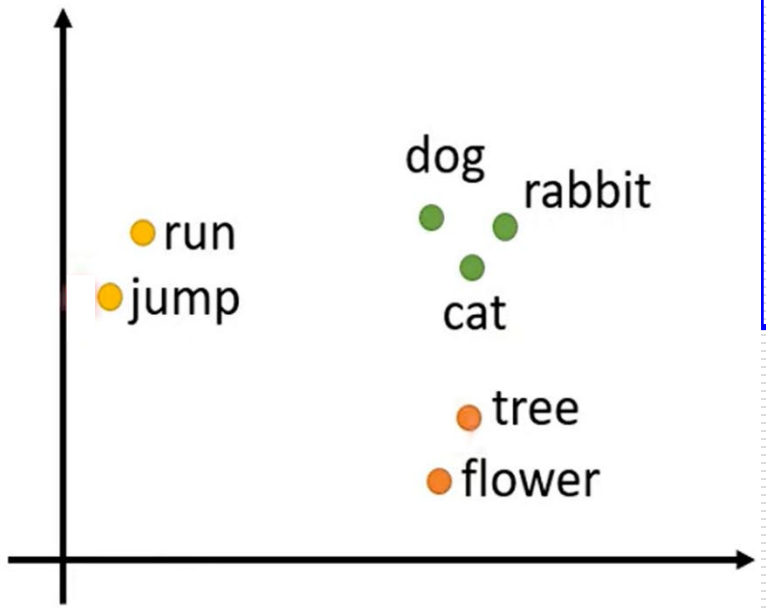
bag = [0 1 0 0 0 .....]

cat = [0 0 1 0 0 .....]

dog = [0 0 0 1 0 .....]

elephant = [0 0 0 0 1 .....]

O ***Word Embedding***



Dimension

Transformer-B: 512

Transformer-L: 1024

BERT-B: 768

BERT-L: 1024

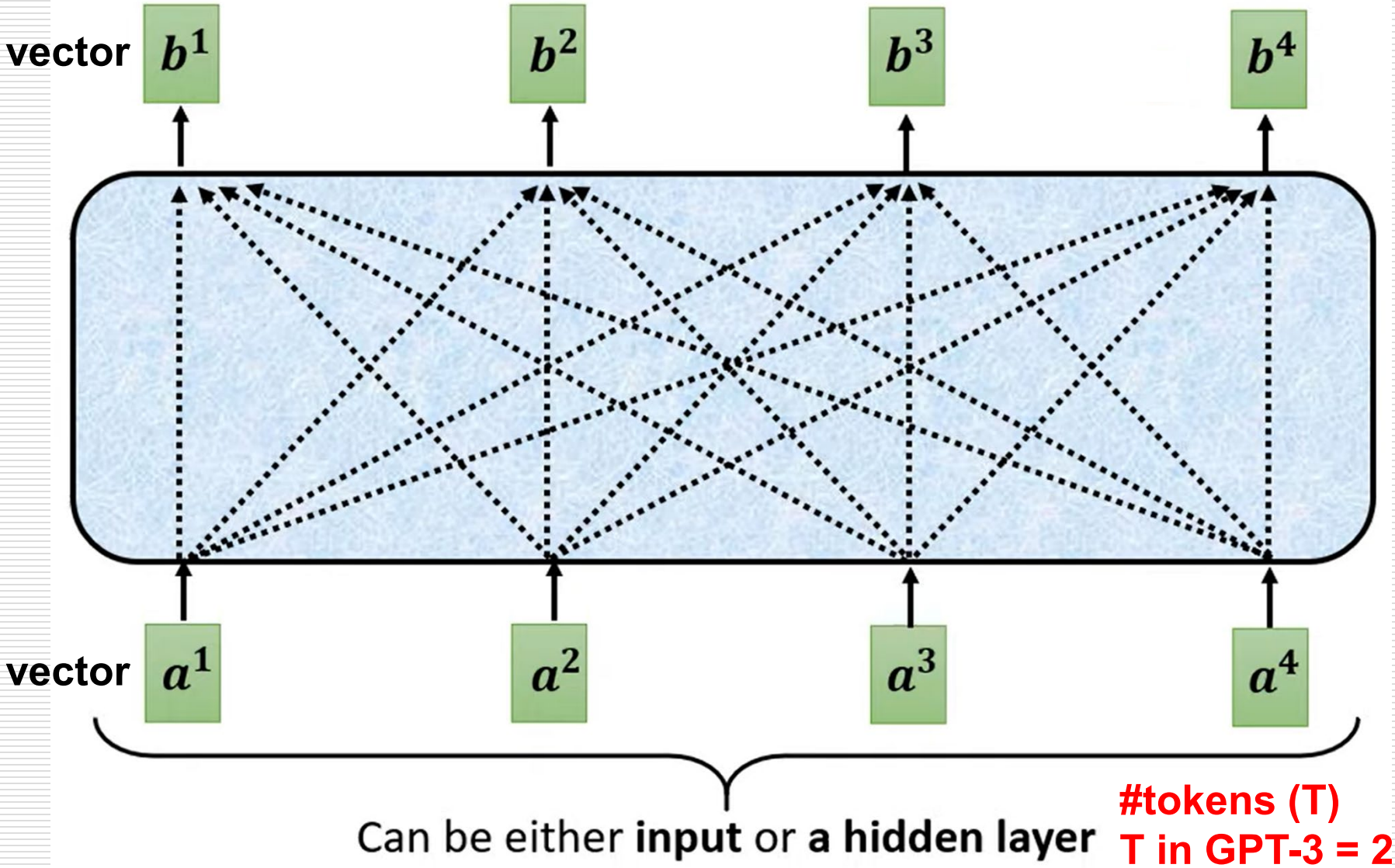
GPT-1: 768

GPT-2: 1600

GPT-3: **12288**

Self-Attention Module (1/2)

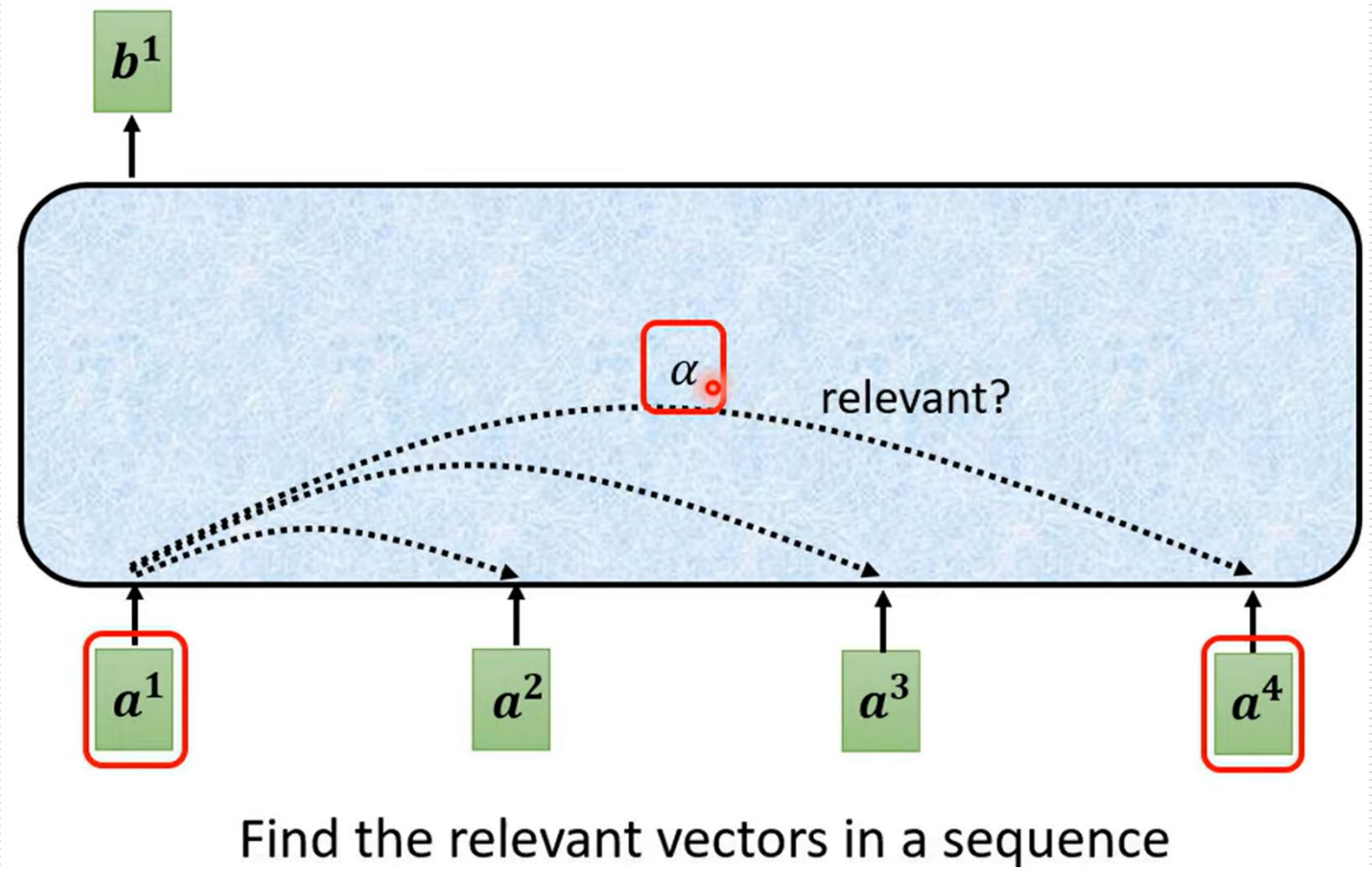
a^i, b^i: same dimension (d)



#tokens (T)
T in GPT-3 = 2048

Self-Attention Module (2/2)

- Find the pairwise relevance
- Calculate b^i from a^1~a^4



Triplet: <Q, K, V>

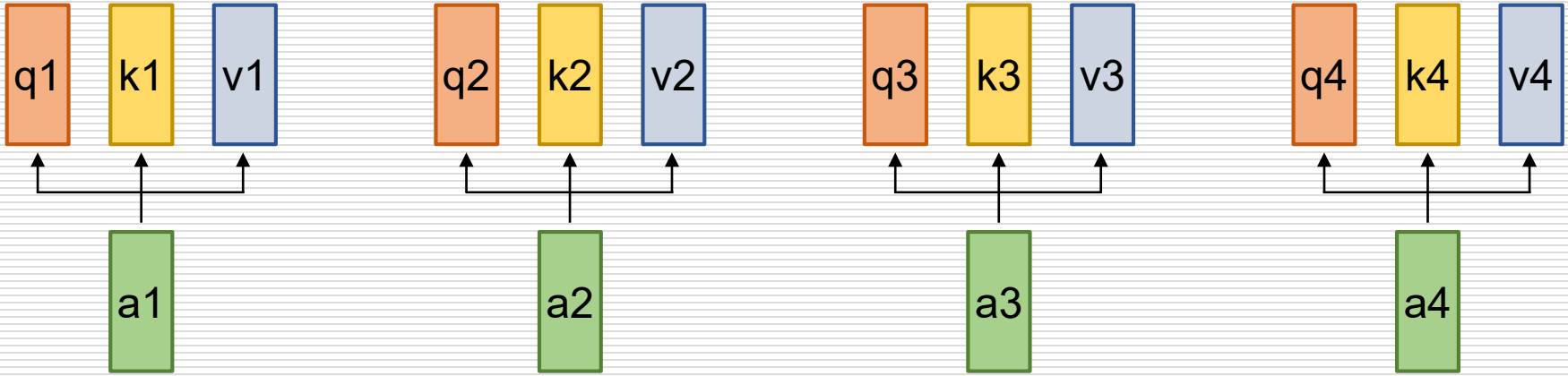
- Q: query
- K: key
- V: value

$$\begin{bmatrix} q^i \\ k^i \\ v^i \end{bmatrix} = \begin{bmatrix} W^q \\ W^k \\ W^v \end{bmatrix} \times \begin{bmatrix} a^i \end{bmatrix}$$

shared

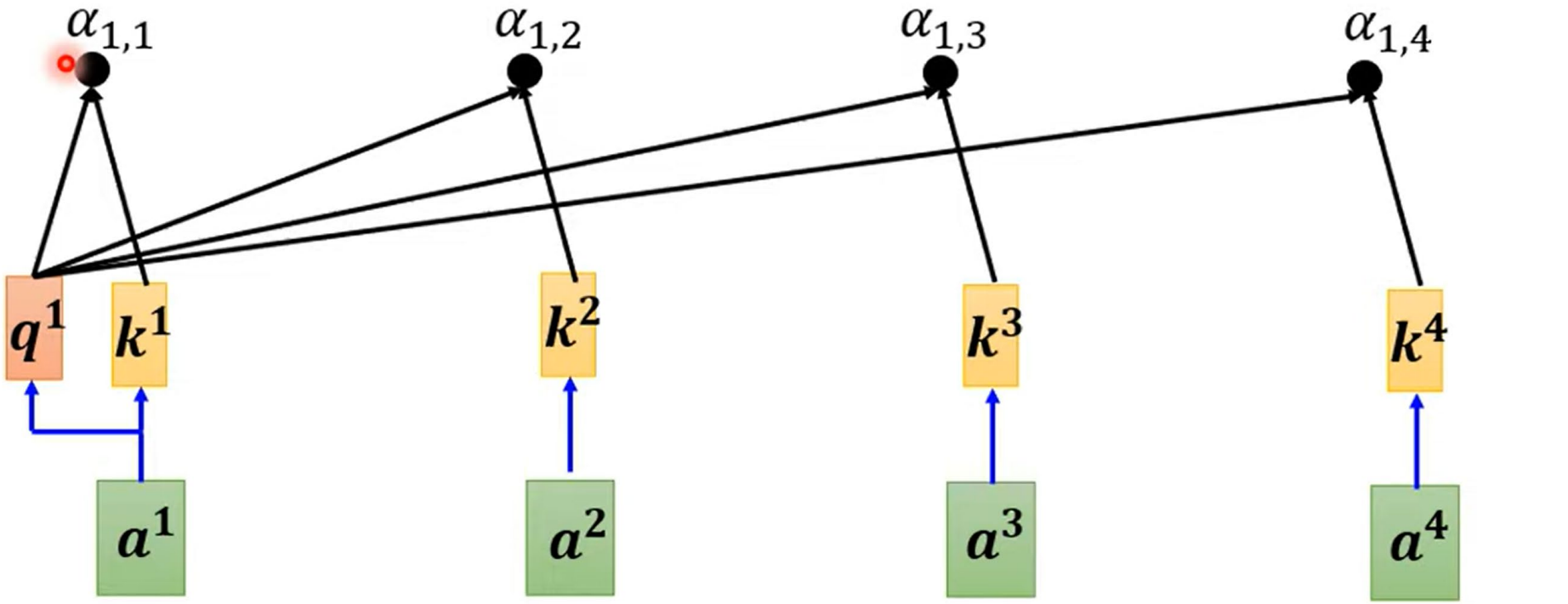
W: d x d

a, q, k, v: d x 1



Relevance: Attention Scores (1/2)

$\alpha_{i,j} = q^i \cdot k^j$ (dot product of two vectors, α is a scalar)



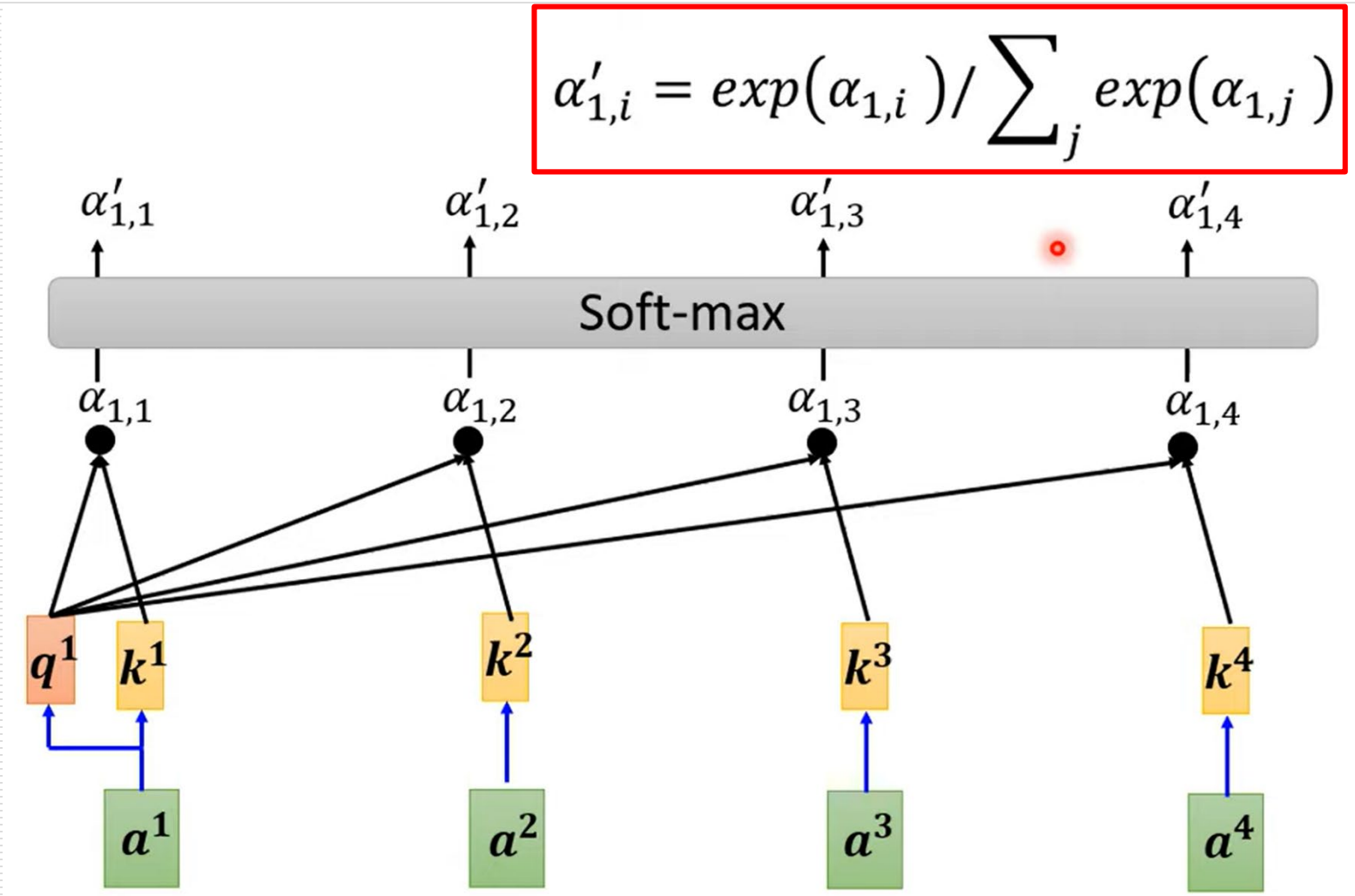
$q^1 = W^q a^1$ $k^2 = W^k a^2$ $k^3 = W^k a^3$ $k^4 = W^k a^4$

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

Relevance: Attention Scores (2/2)

- Normalization via **Softmax**

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

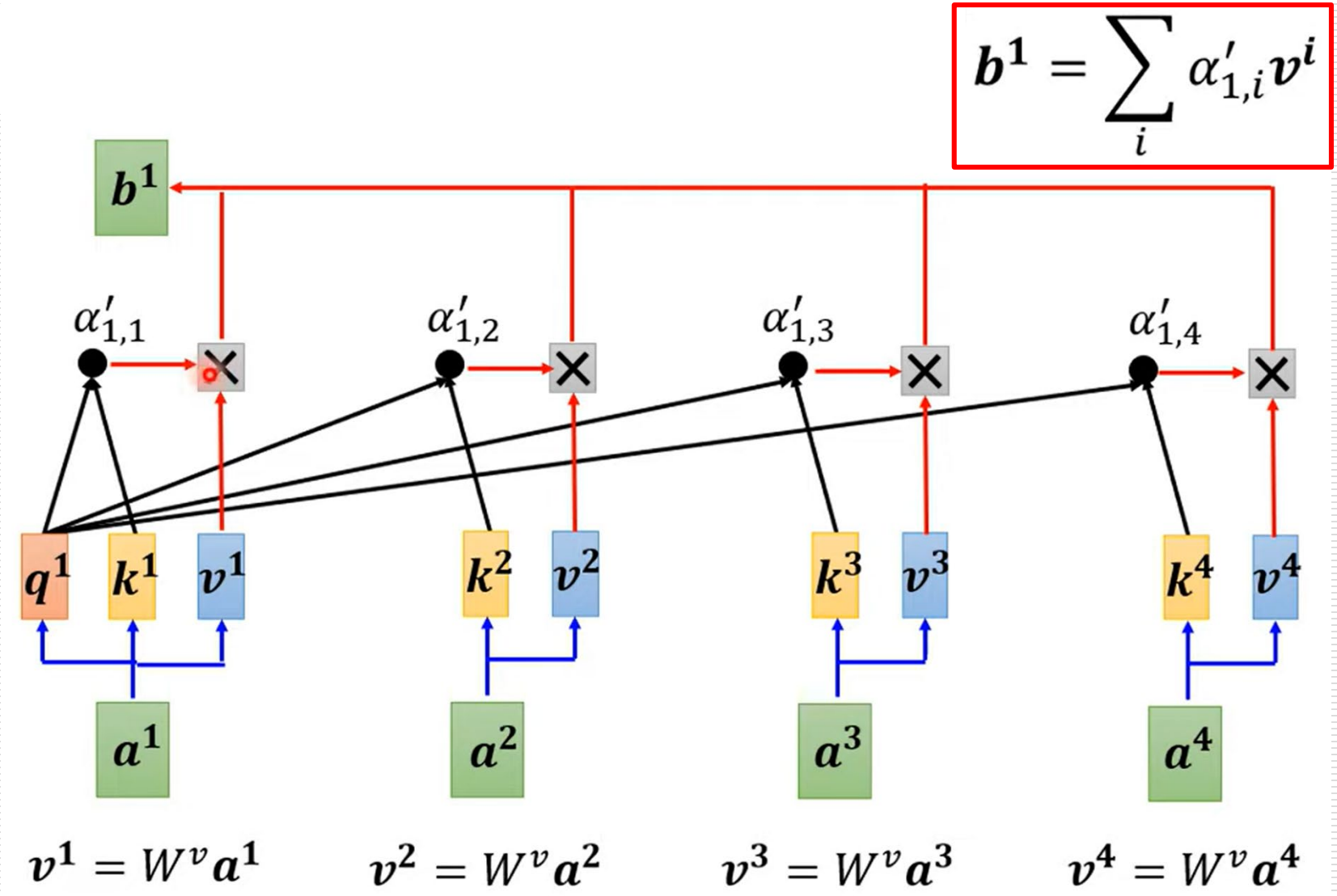


Exponential function…

real division

Output Generation (1/2)

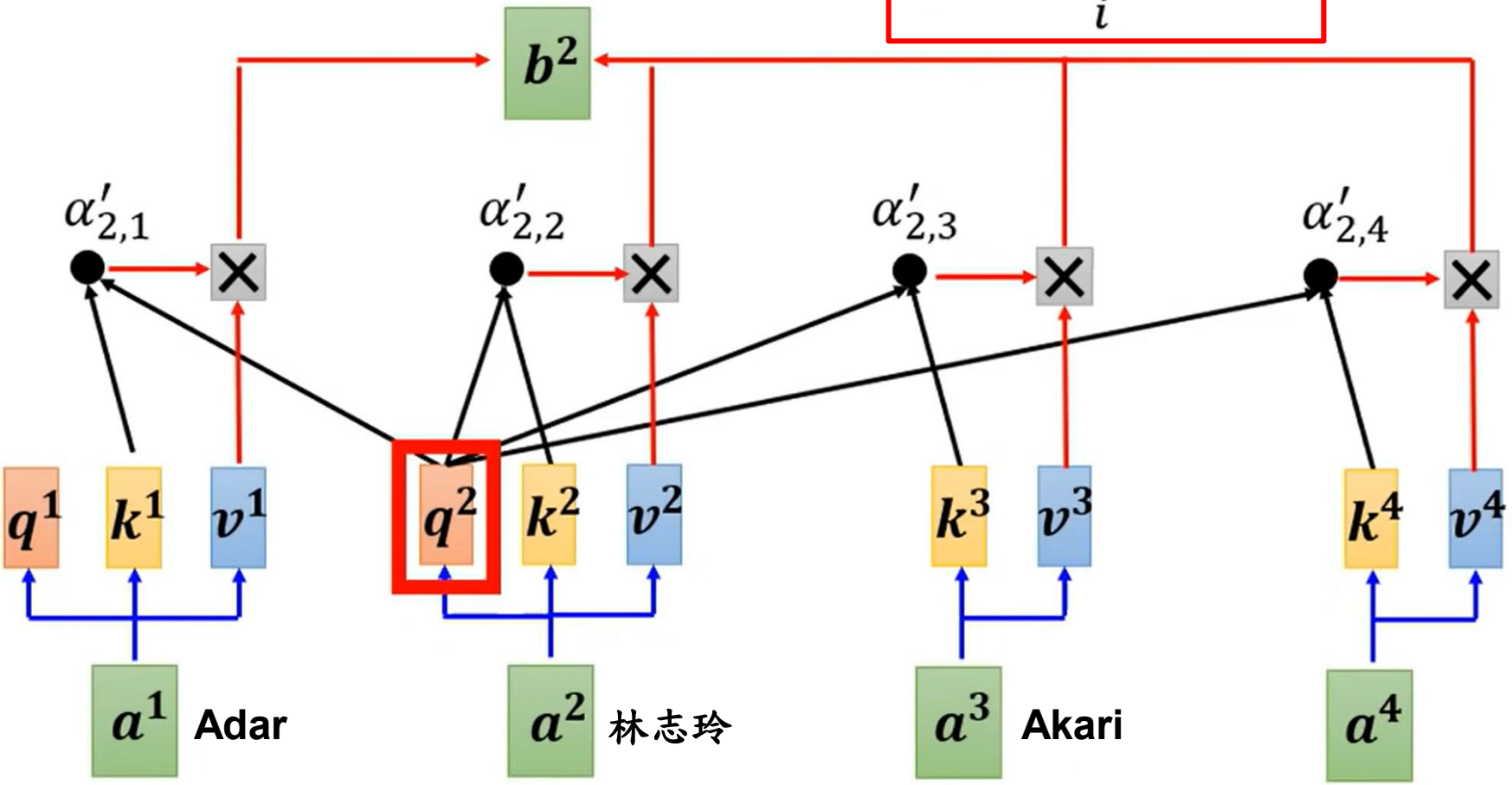
$$b^1 = \sum_i \alpha'_{1,i} v^i$$



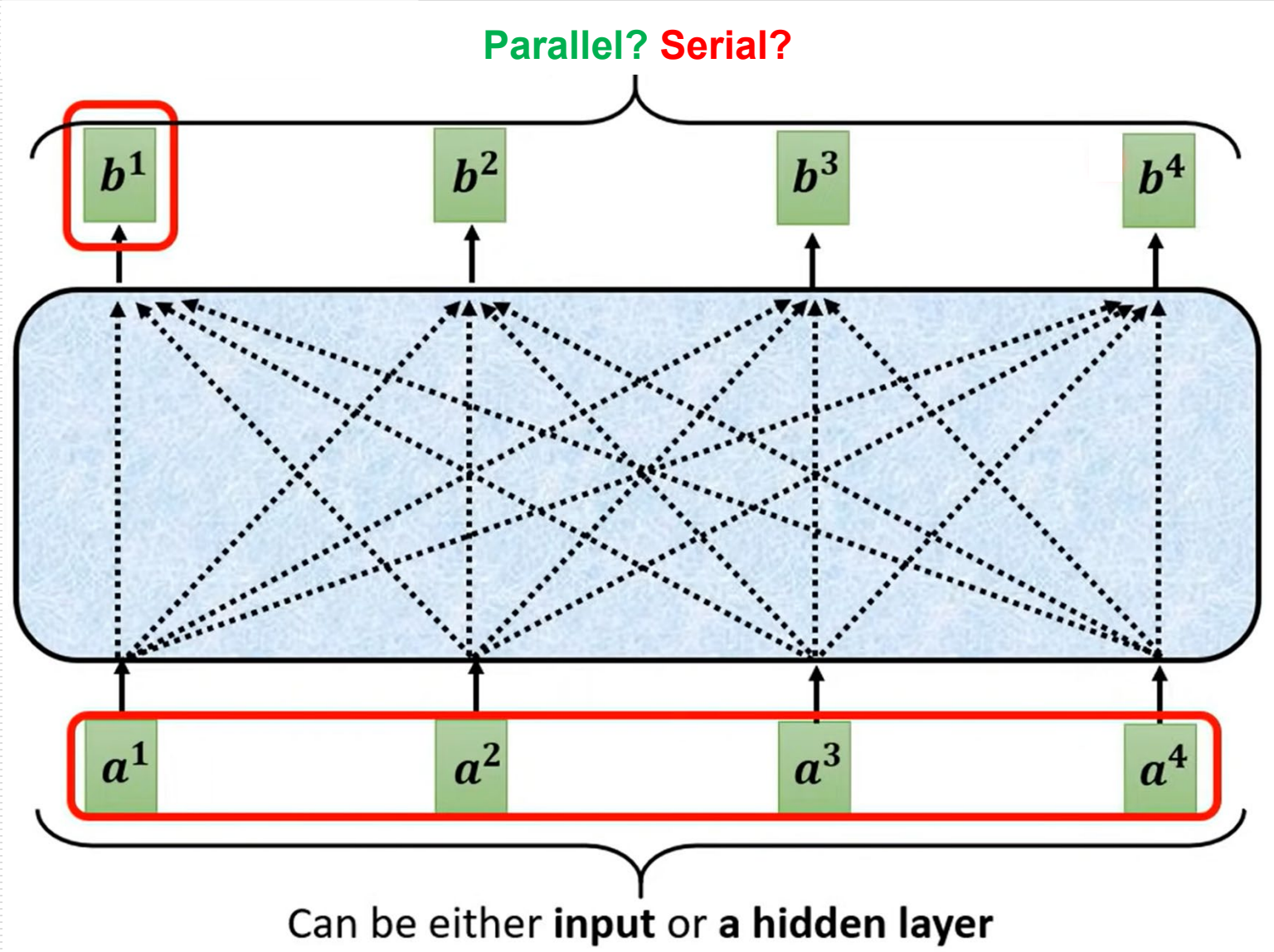
Output Generation (2/2)

$$\alpha_{i,j} = q^i \cdot k^j \neq q^j \cdot k^i = \alpha_{j,i}$$

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



Computation in Serial or Parallel?



<Q, K, V> in Parallel

$$q^i = W^q a^i \qquad q^1\,q^2\,q^3\,q^4 = W^q\,a^1\,a^2\,a^3\,a^4$$

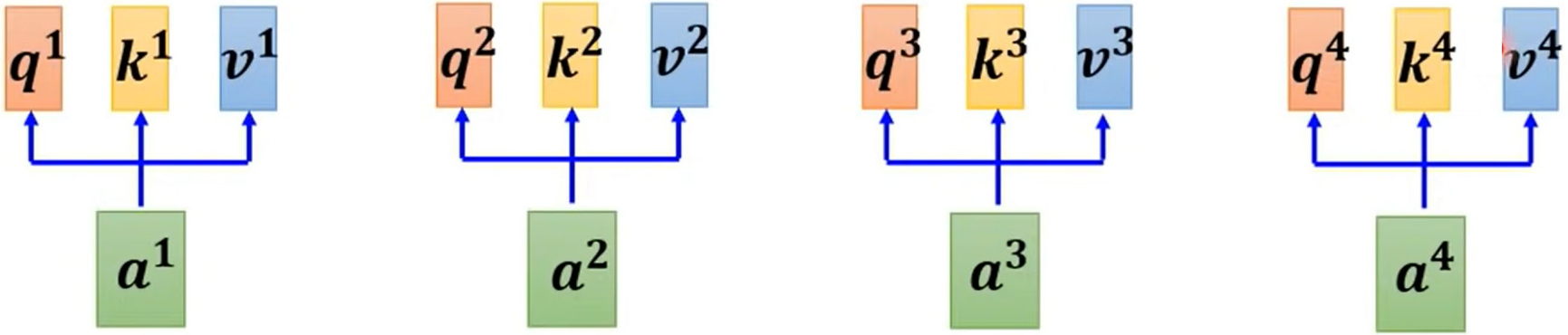
Q $\qquad$ I

$$k^i = W^k a^i \qquad k^1\,k^2\,k^3\,k^4 = W^k\,a^1\,a^2\,a^3\,a^4$$

K $\qquad$ I

$$v^i = W^v a^i \qquad v^1\,v^2\,v^3\,v^4 = W^v\,a^1\,a^2\,a^3\,a^4$$

V $\qquad$ I

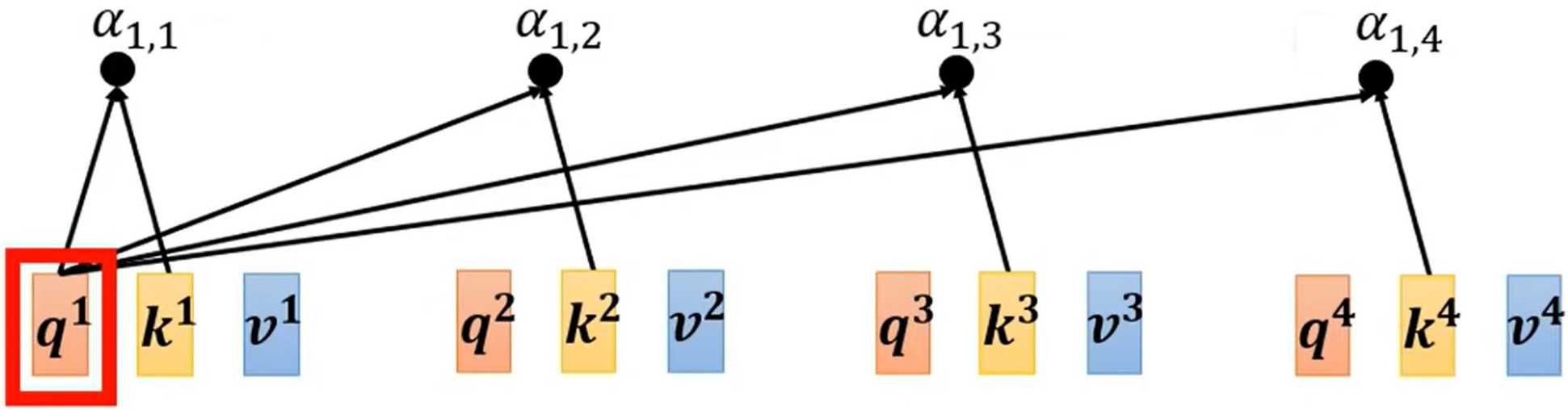


Attention Scores (α) in Parallel (1/2)

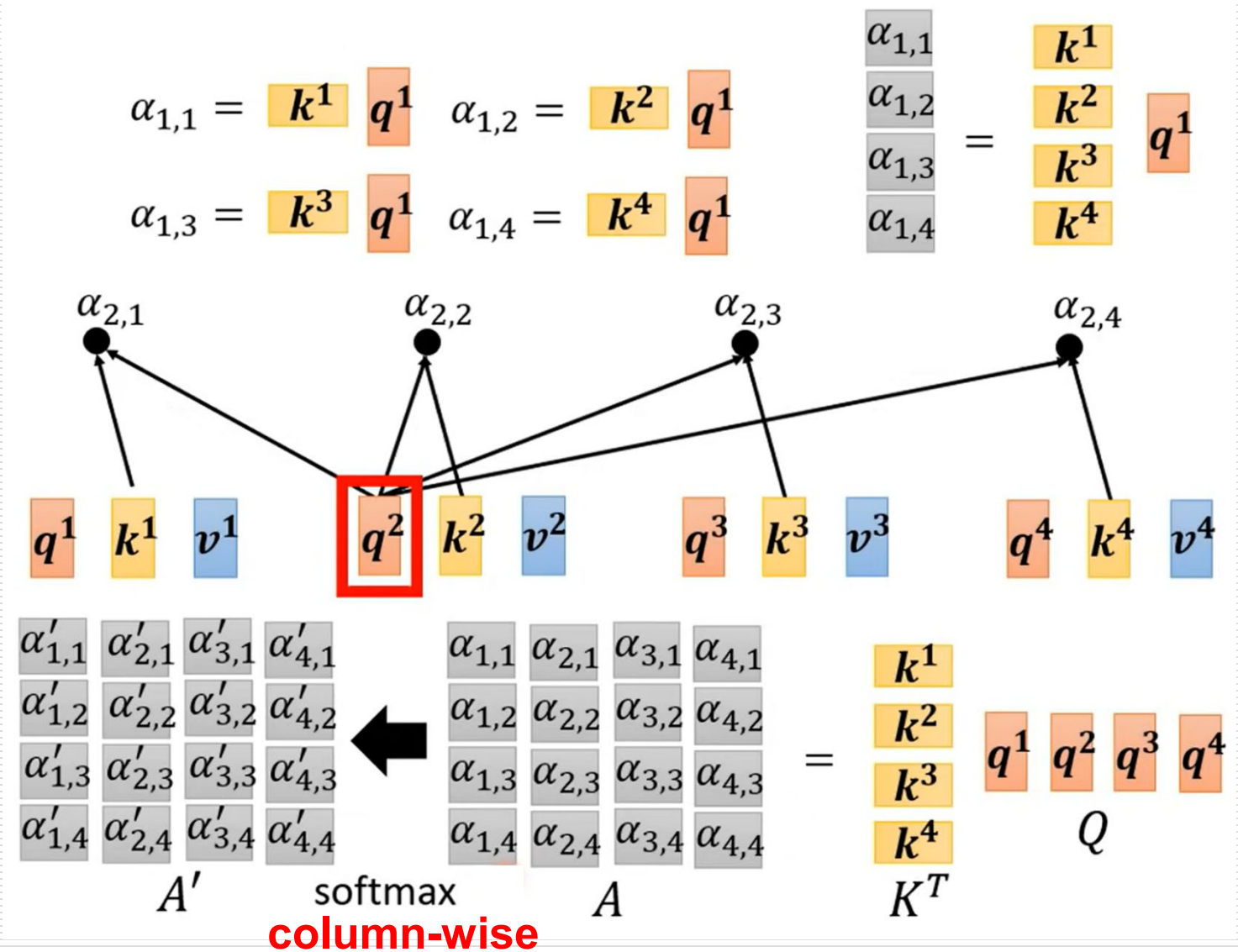
$$\alpha_{1,1} = k^1 q^1 \quad \alpha_{1,2} = k^2 q^1$$

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

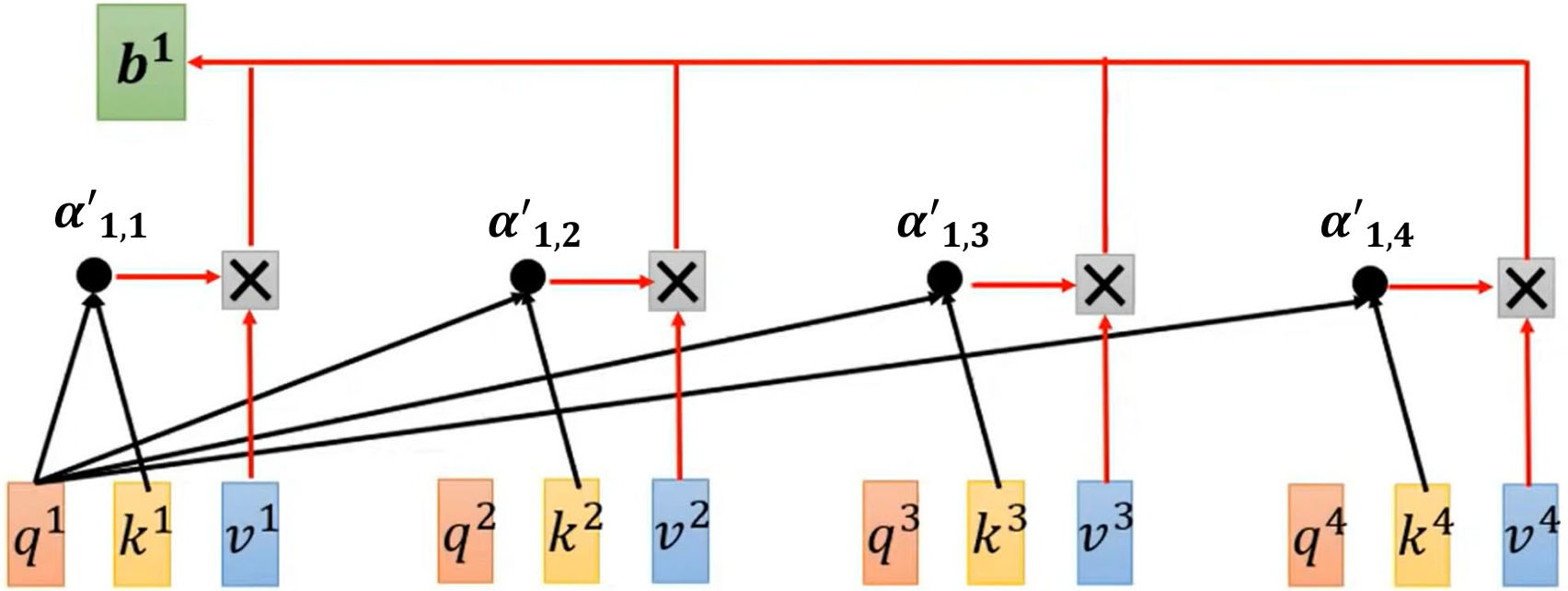
$$\begin{matrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{matrix} = \begin{matrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{matrix} q^1$$



Attention Scores (α) in Parallel (2/2)



Output Generation in Parallel



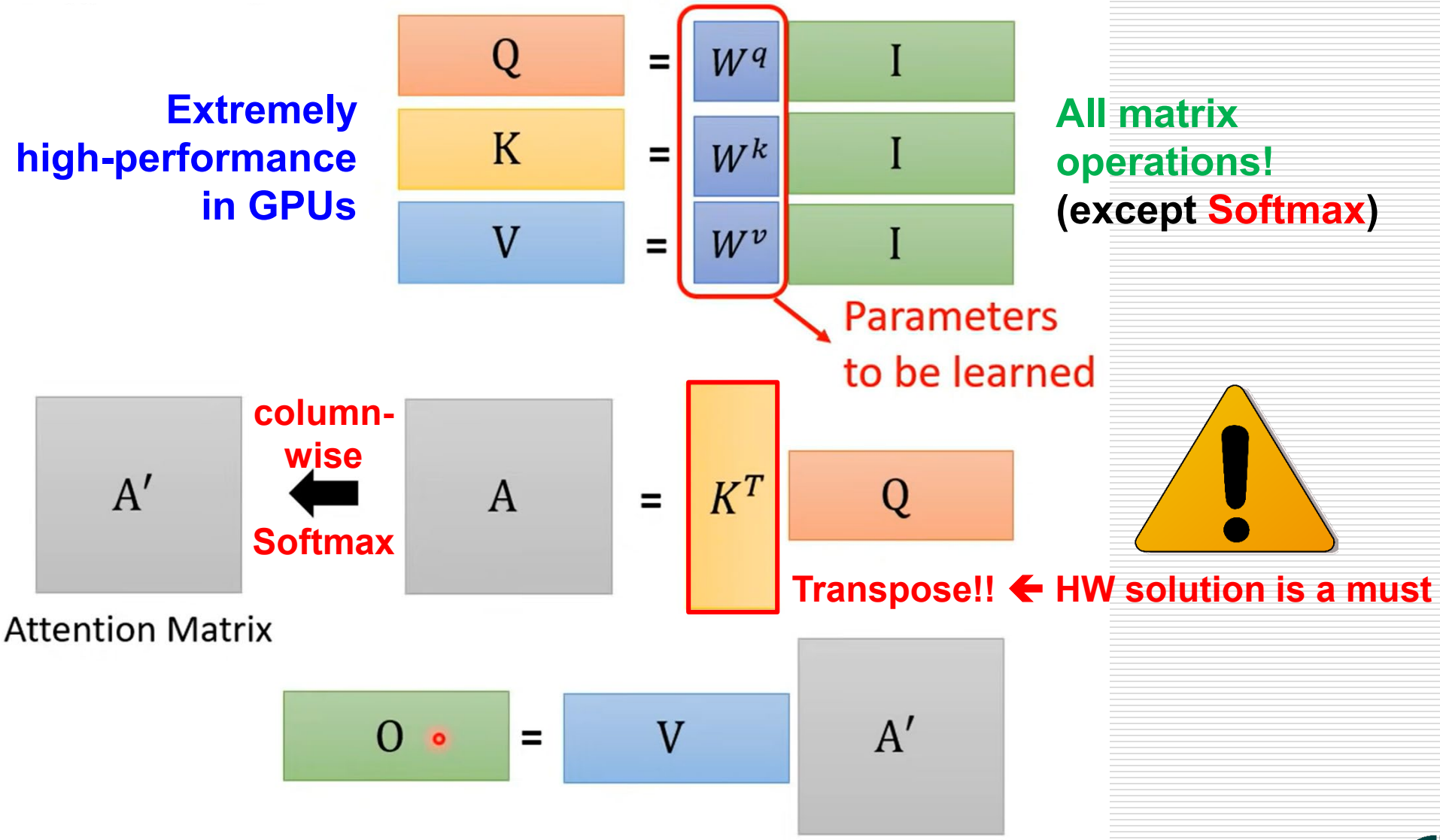
$$\begin{bmatrix} b^1 & b^2 & b^3 & b^4 \end{bmatrix} = \begin{bmatrix} v^1 & v^2 & v^3 & v^4 \end{bmatrix}$$

O V

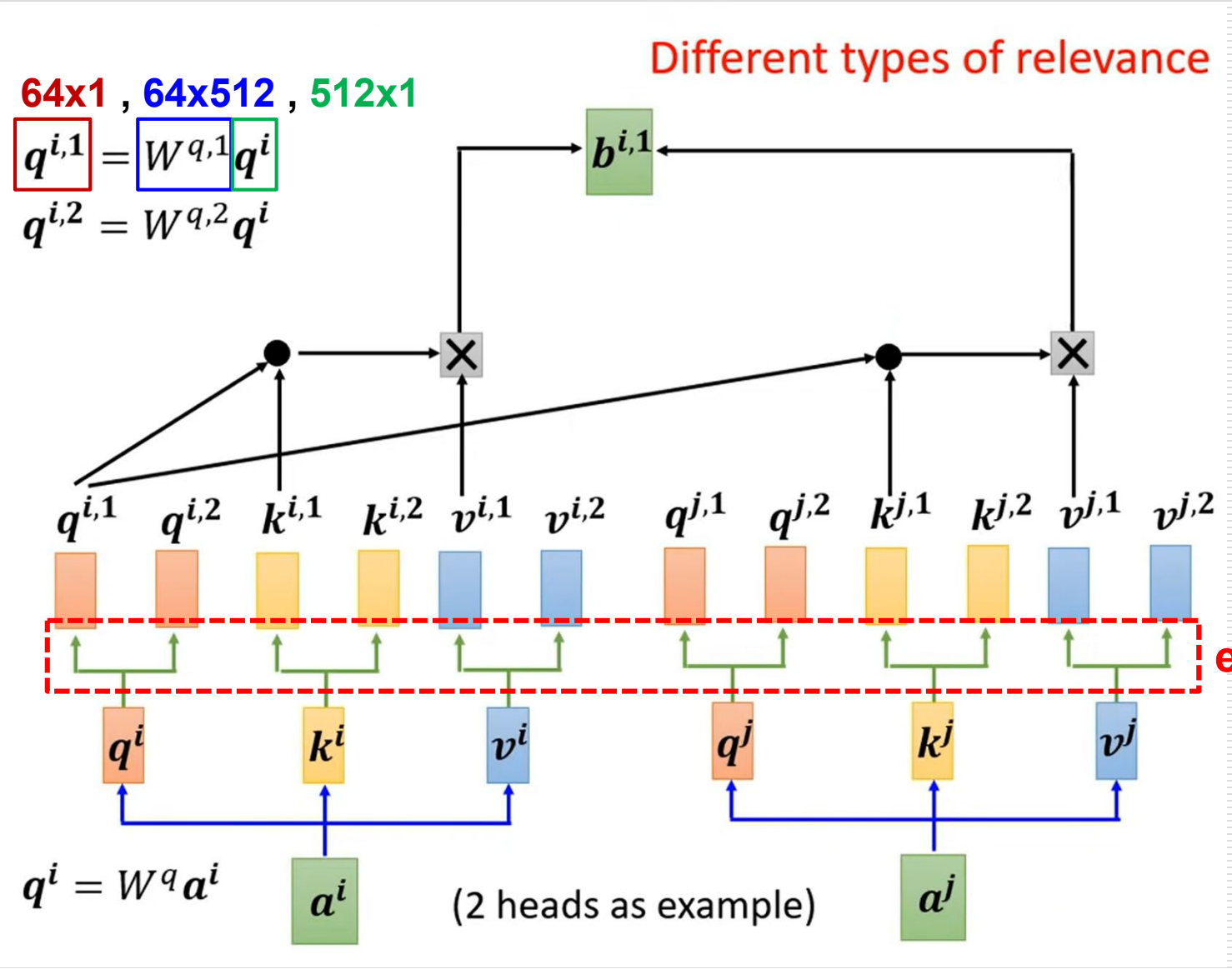
$$\begin{bmatrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{bmatrix}$$

A'

Put Them Together



Multi-Head Self-Attention (MSA) (1/2)



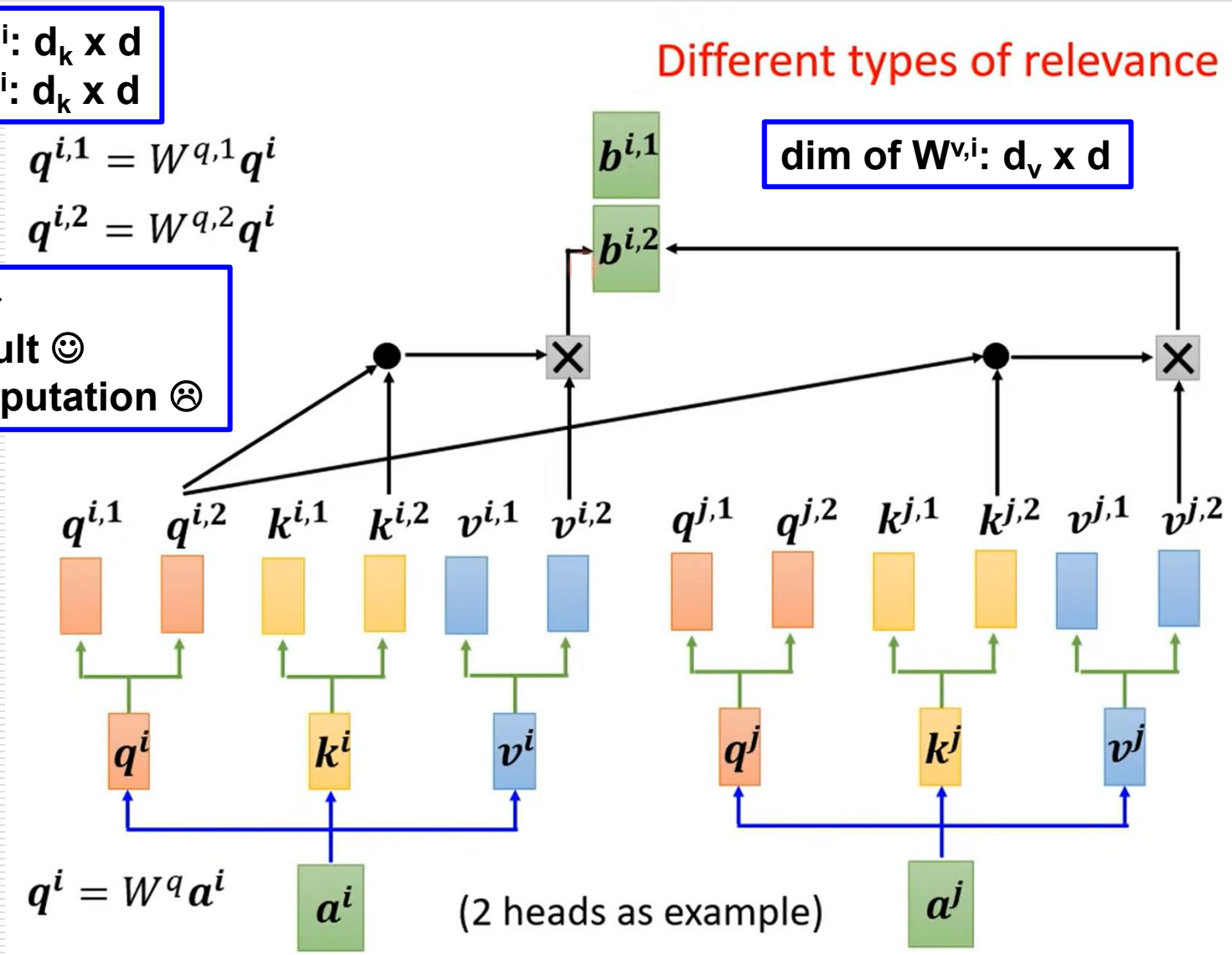
Dimension
$q^{i,j}$, $k^{i,j}$: d_k
$v^{i,j}$: d_v

Multi-Head Self-Attention (MSA) (2/2)

dim of $W^{q,i}$: d_k x d
dim of $W^{k,i}$: d_k x d

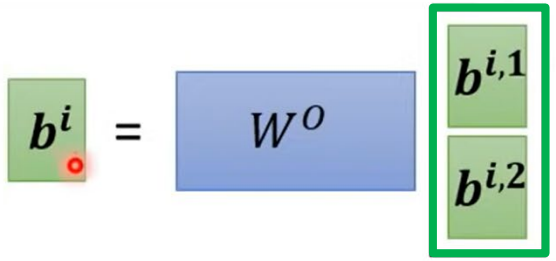
$$q^{i,1} = W^{q,1} q^i$$
$$q^{i,2} = W^{q,2} q^i$$

large d_k ➔
better result ☺
more computation ☹



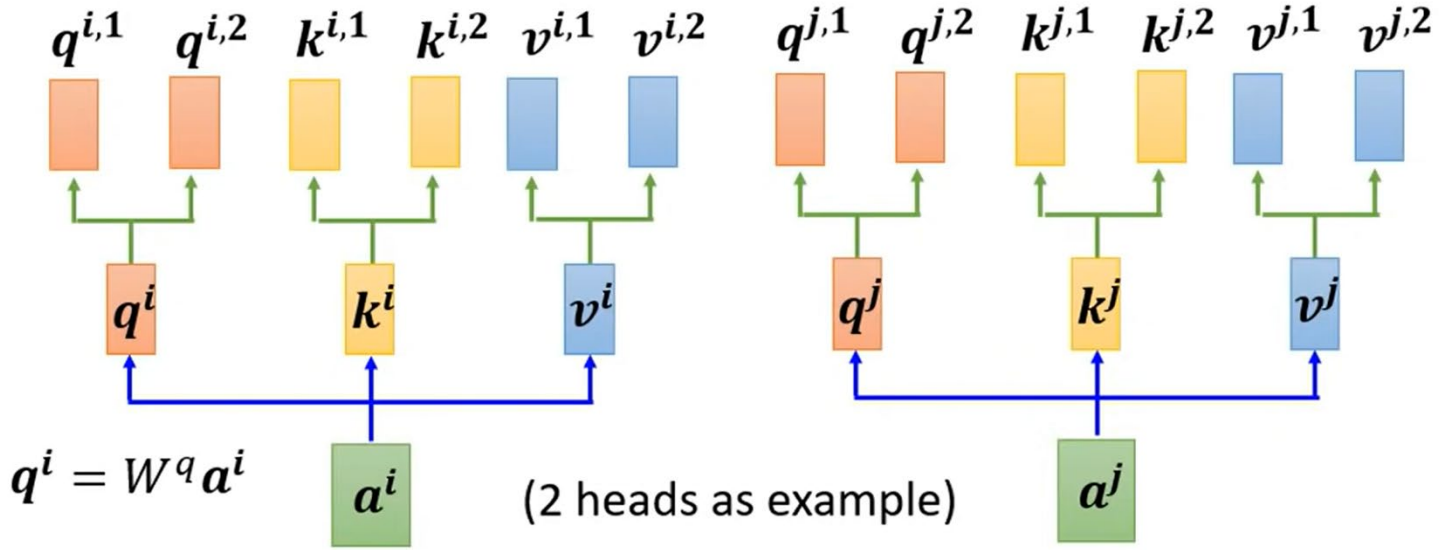
Output Generation in MSA

Different types of relevance

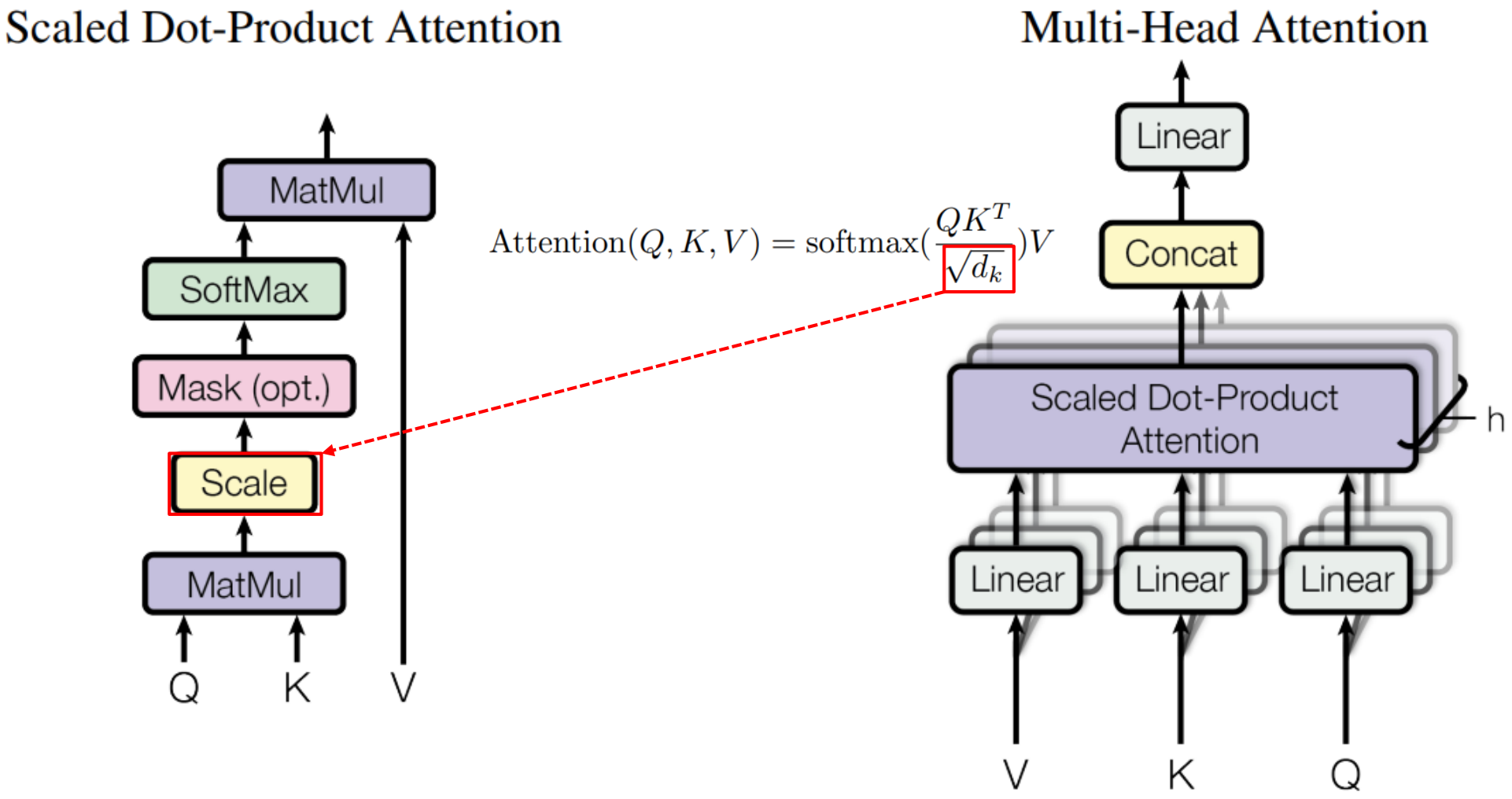


h: #heads
d_v: dim of $b^{i,j}$
$d_o = h \times d_v$
W^O: $d \times d_o$

in Transformer:
h = 8 or 16, and
$d_O = d$, typically

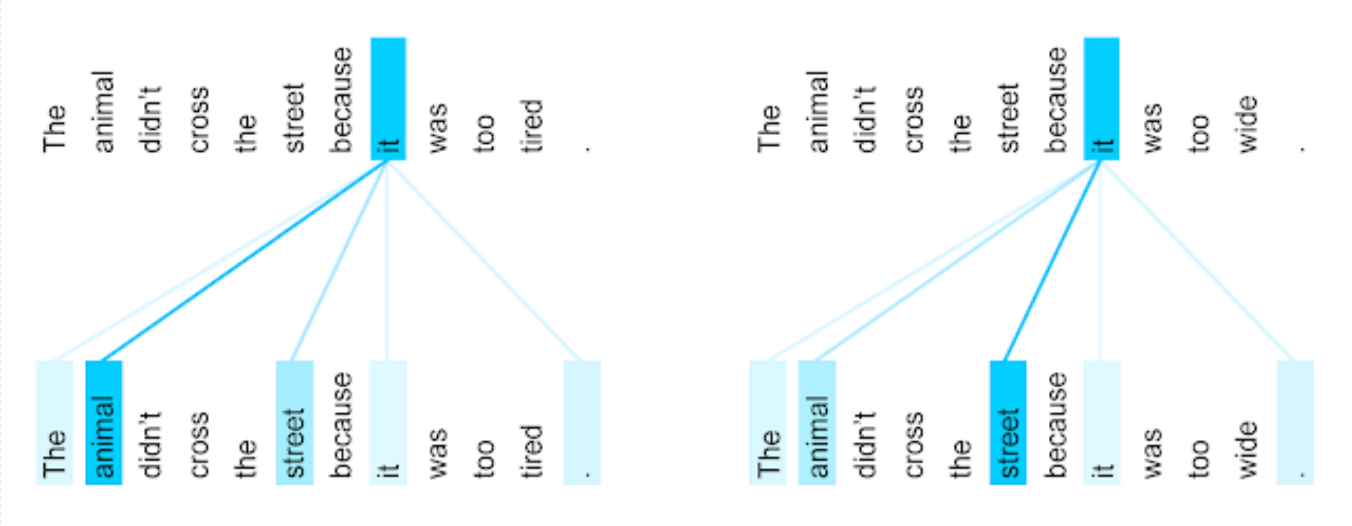


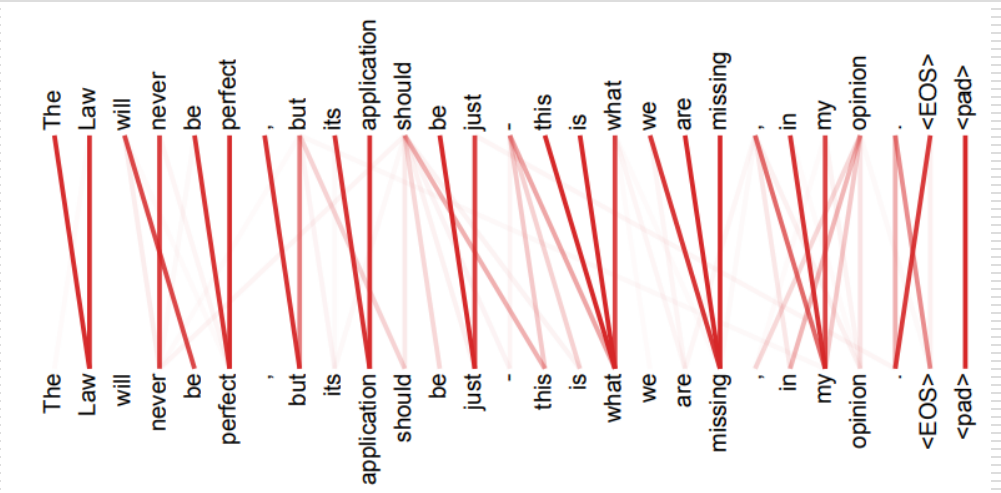
Single-Head vs. Multi-Head

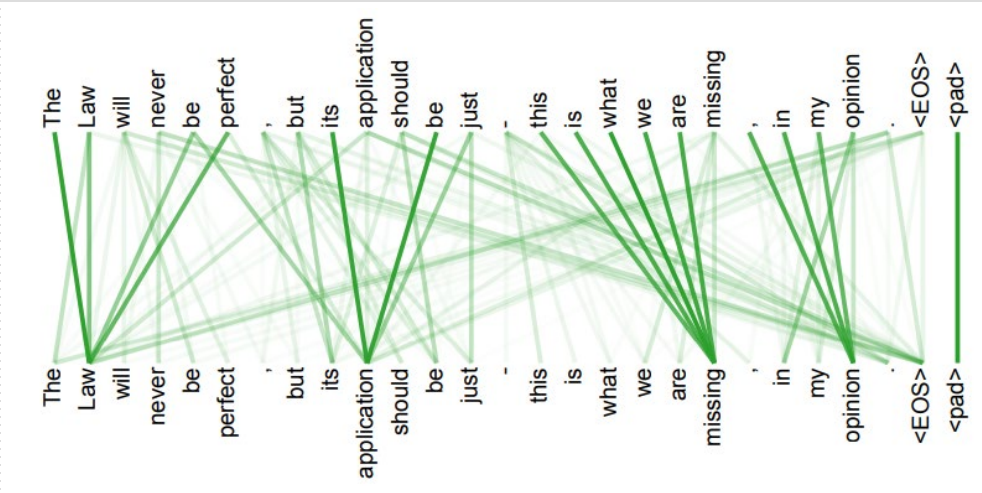


Visualization of MSA

- Different heads learn different kinds of relevance







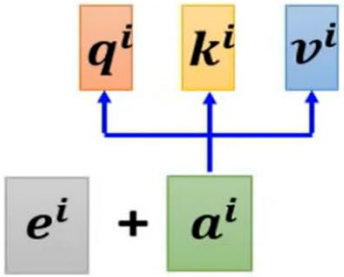
Positional Encoding (1/2)

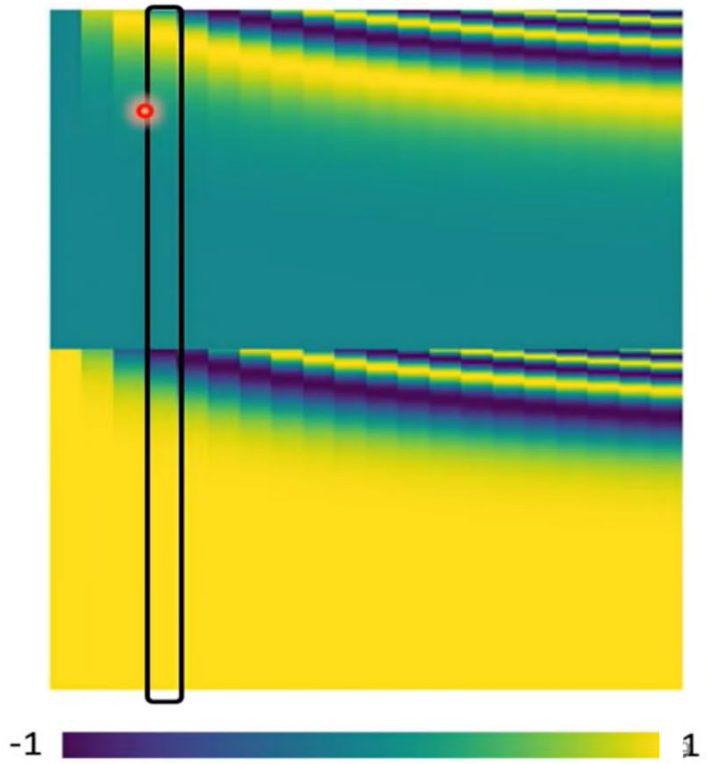
- 小龍女：我也想過過過兒過過的生活。
- 差點没上上上上海的車。
- 你等一會會會會會會計的人嗎?
- 校長：校服上除了校徽別別別的，讓你別別別的別別別的你非得別別的！

- I saw a saw saw a saw.
- Two to two to two two.
- Don’t trouble trouble until trouble troubles you.
- Is this a ship-shipping ship, shipping shipping ships?
- Can you can a can like a canner cans a can?

Positional Encoding (2/2)

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

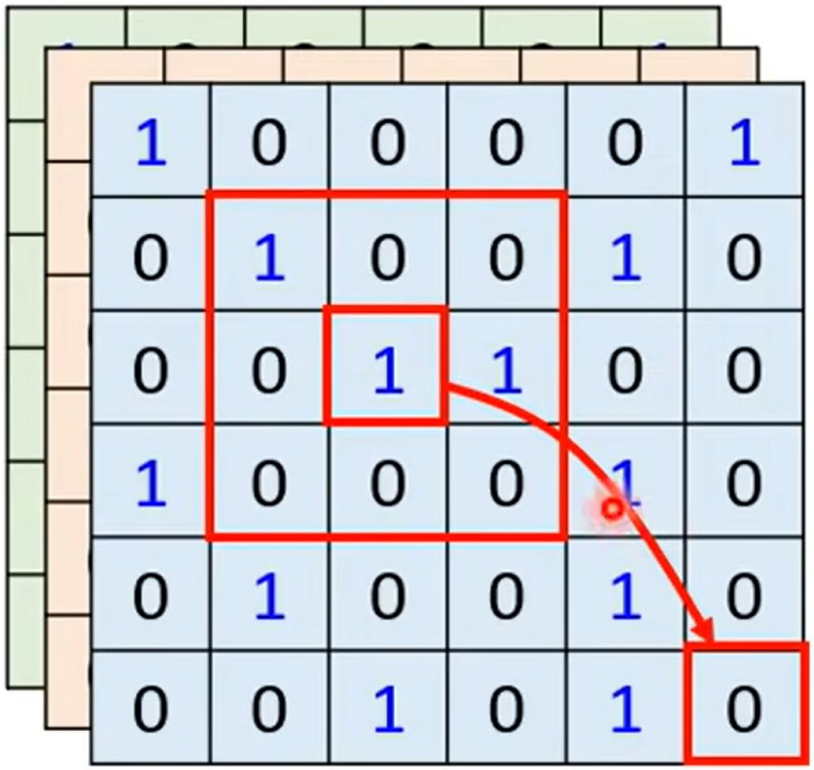




$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Self-Attention vs. CNN



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

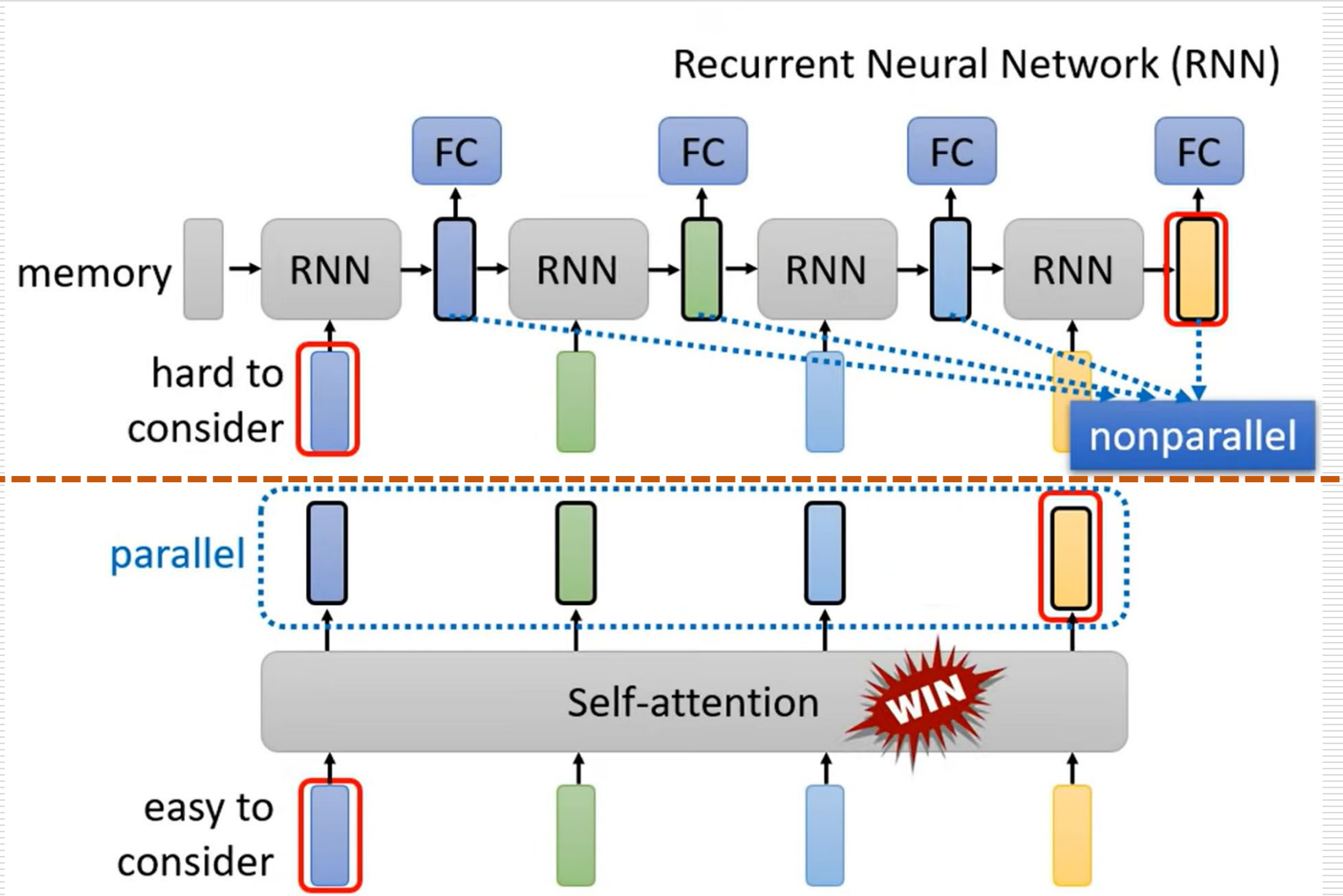
- CNN is simplified self-attention.

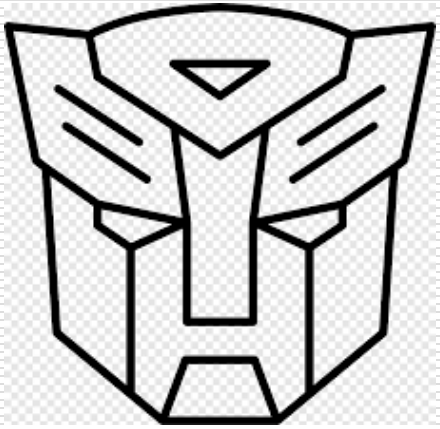
Self-attention: CNN with learnable receptive field

- Self-attention is the complex version of CNN.

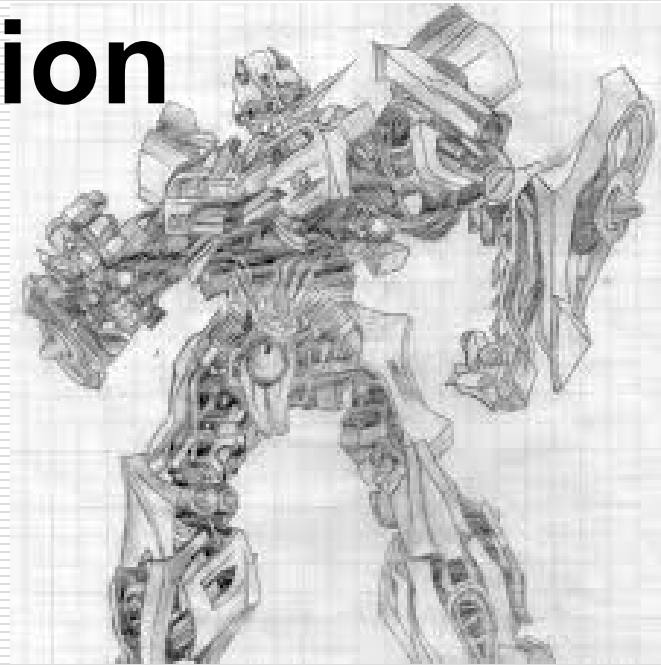
However, self-attention requires notably more computations

Self-Attention vs. RNN





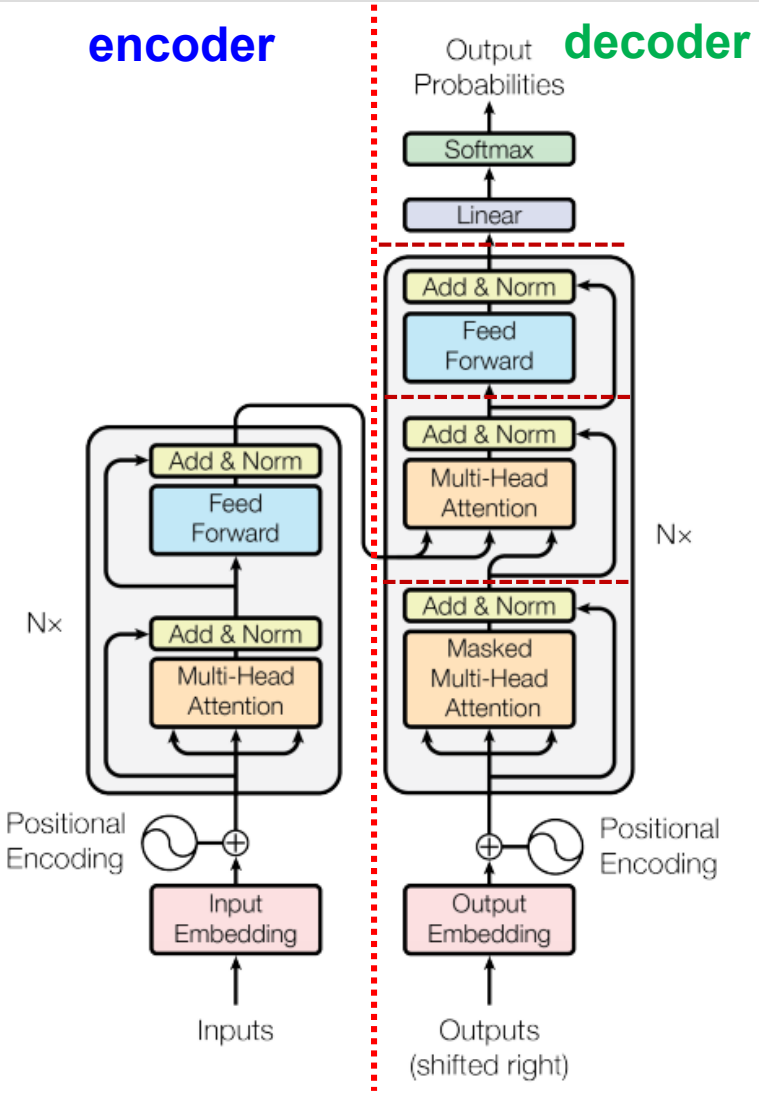
Transformer for Machine Translation



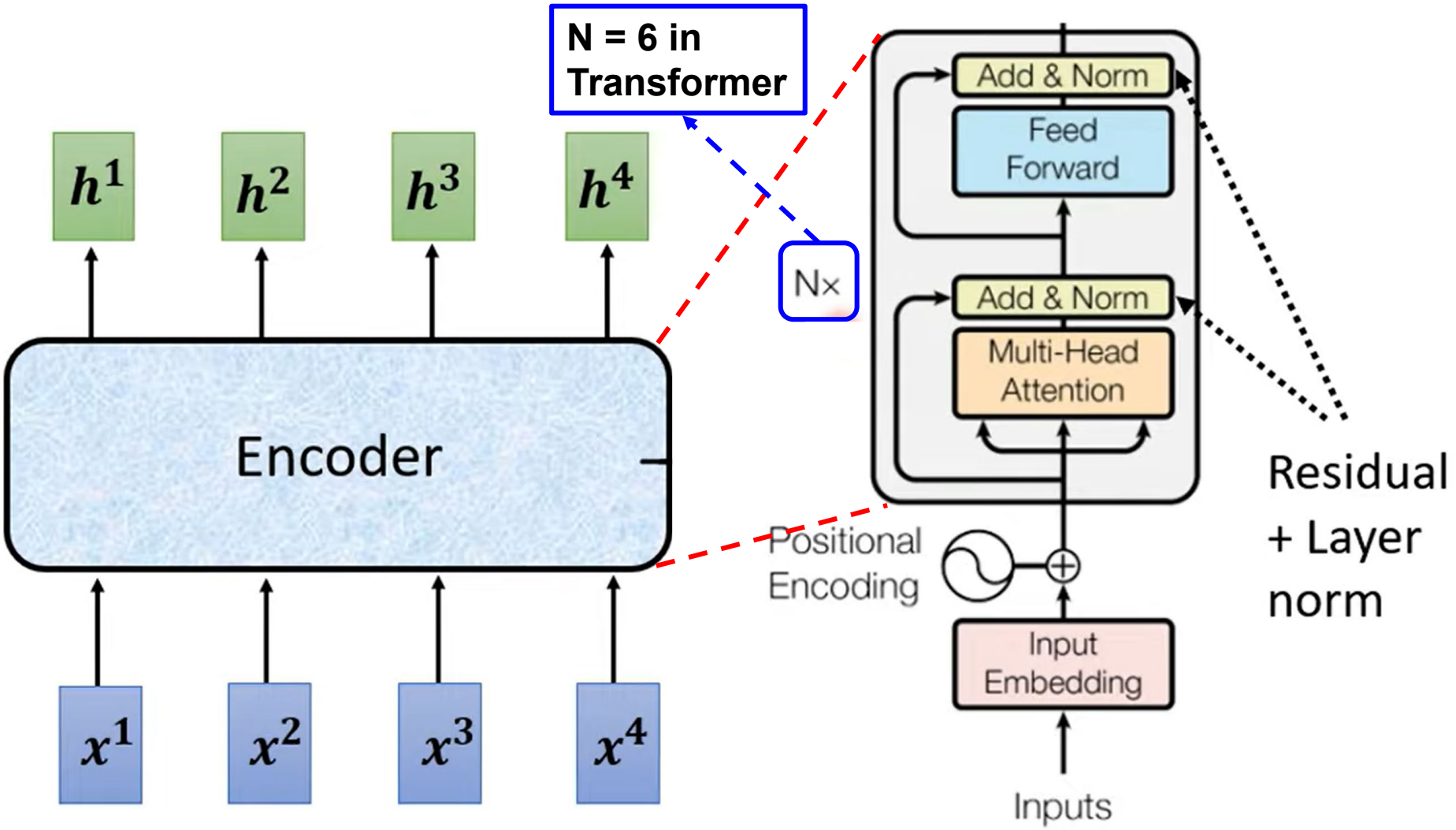
* 在本節中，沒有任何一隻變形金剛受到傷害。
* No Transformers were harmed in this section.

Transformer

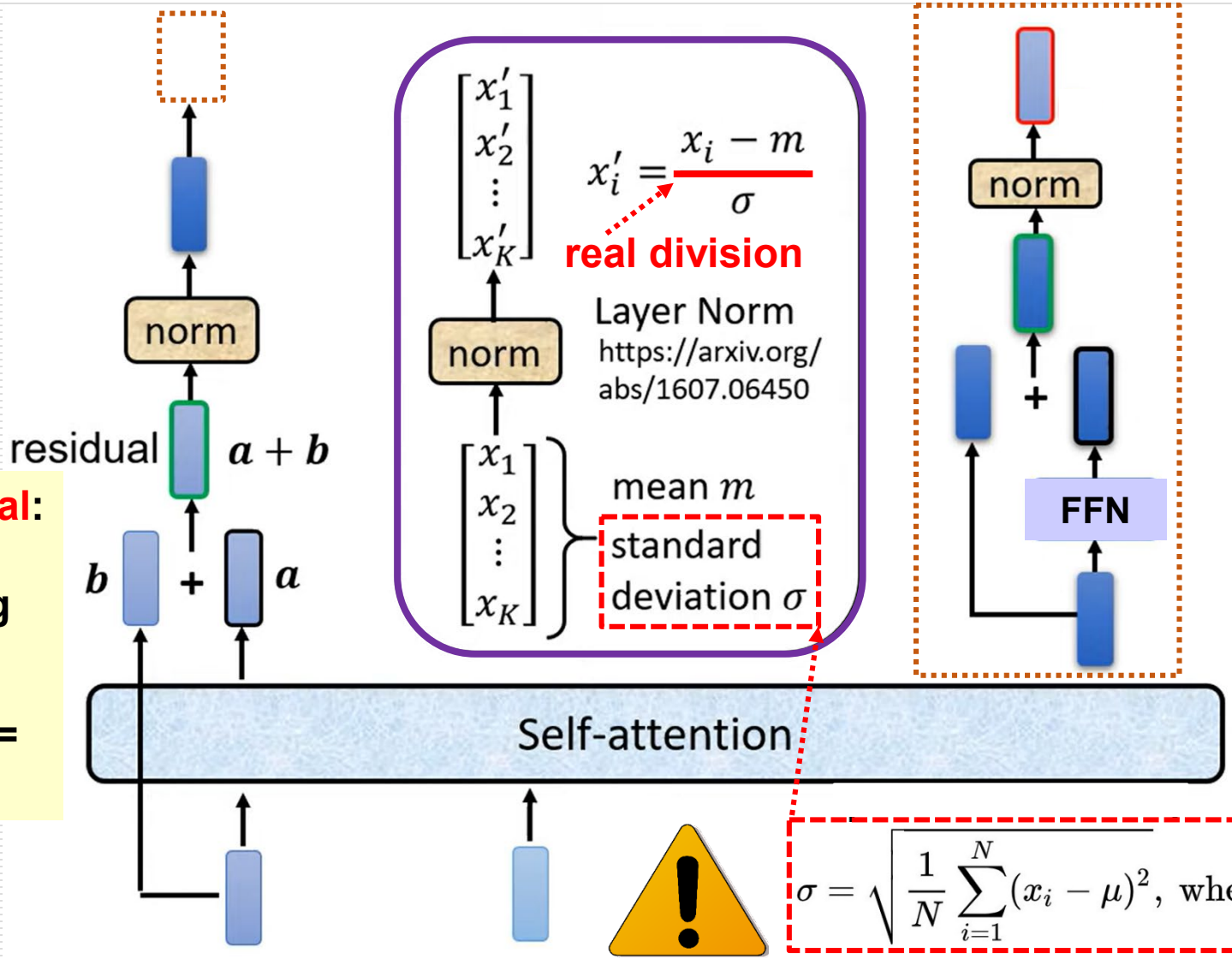
- Proposed by Google in NIPS 2017 (arXiv @ June 2017)
- Elements
  - encoder-decoder architecture
  - word embedding + positional encoding
  - (masked) MSA
  - layer normalization + residual connection
  - feed-forward network (FFN)



Encoder



Residual Link and Layer Normalization



Residual: make training easier

dim(a) = dim(b)

LayerNorm: limit the magnitude ➔ make training easier

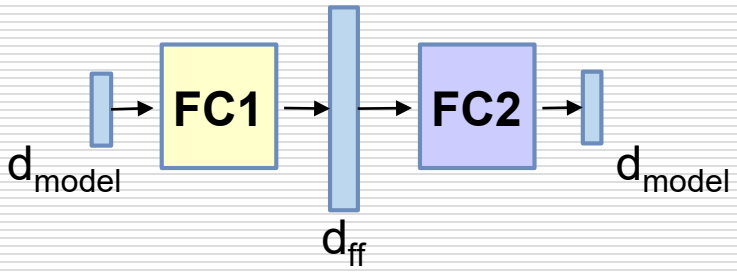
How?
by extra HW?
by a local CPU?

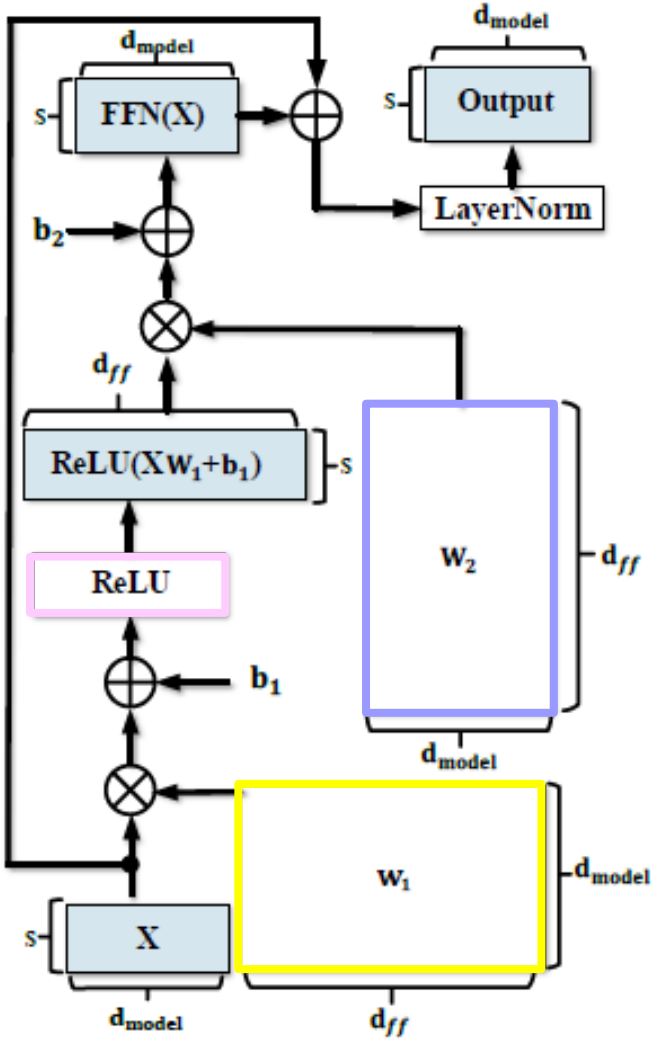
Feed-Forward Network (FFN)

- It consists of two fully-connected (FC) layers and one activation function ReLU in-between

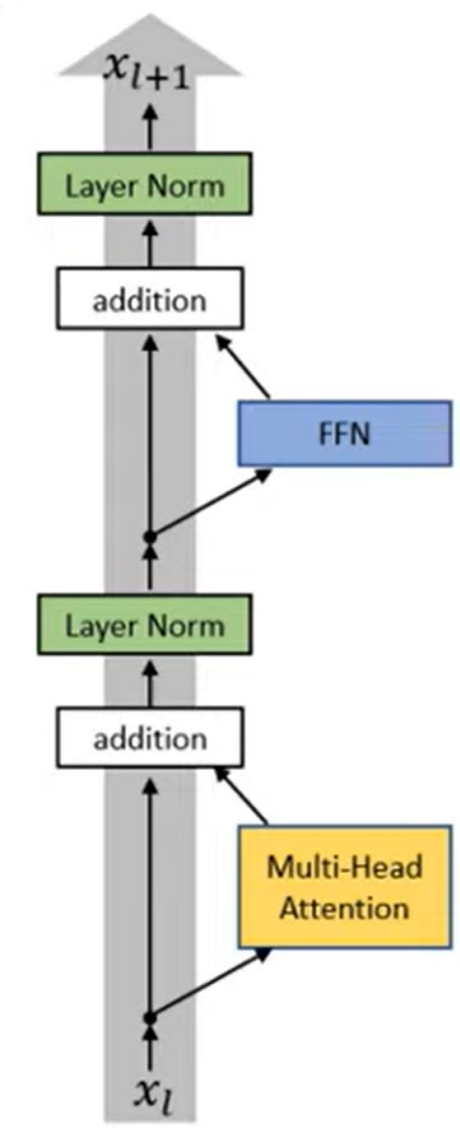
$$FFN(x) = \max(0, xW1 + b1)W2 + b2$$

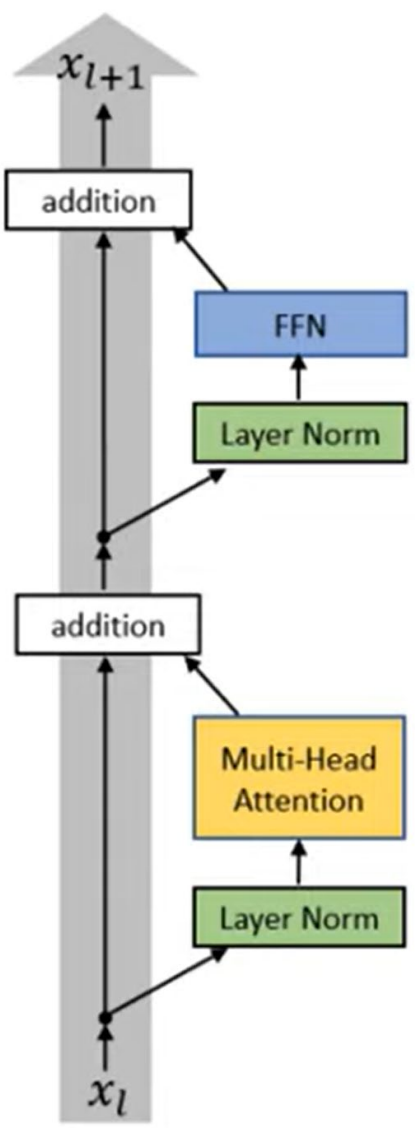
- Hidden dimension is raised by **4x**
  - i.e., $d_{ff} : d_{model} = 4:1$ (2048 : 512)





Model Variants





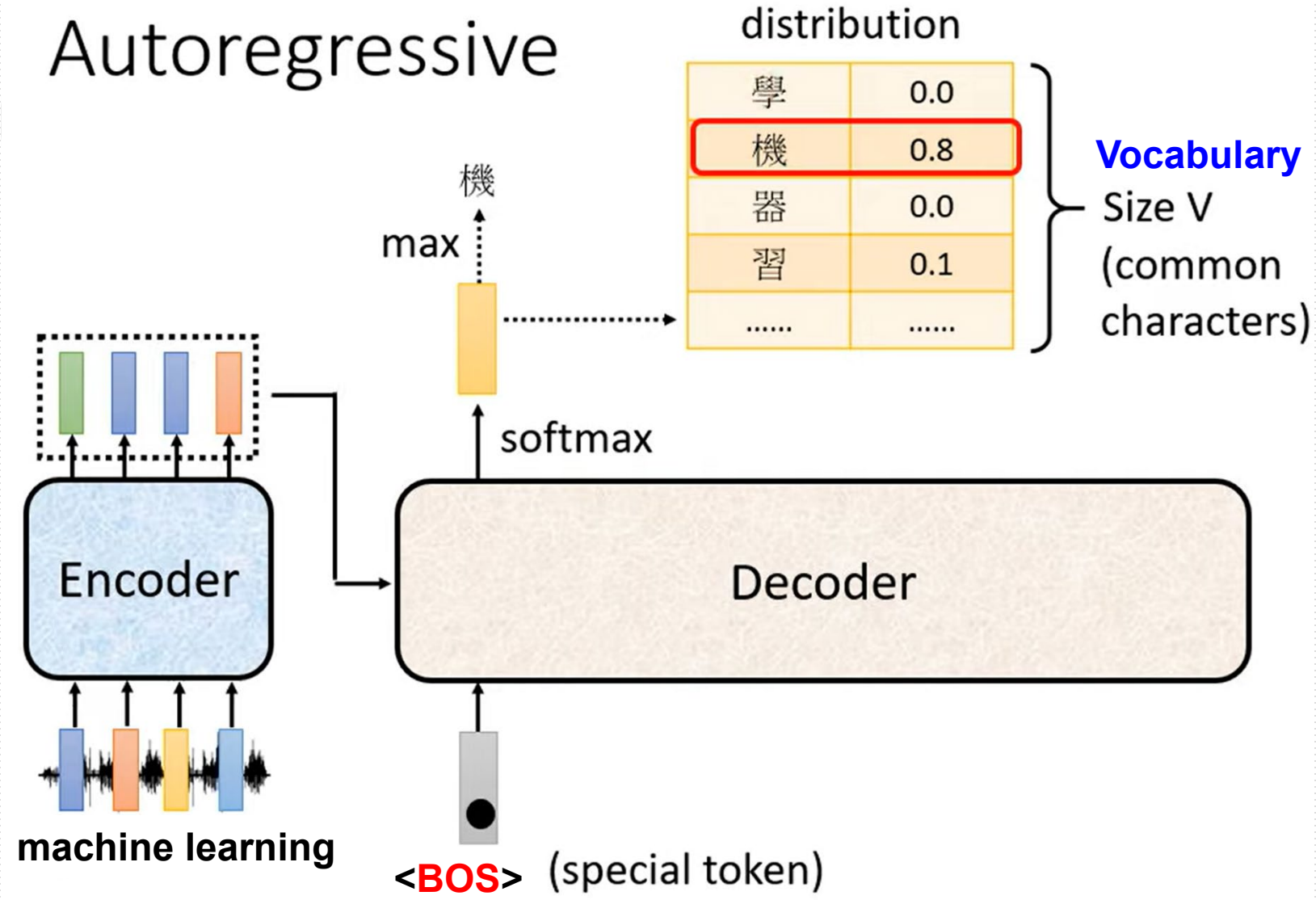
ReLU → GELU

Transformer
BERT
GPT-1

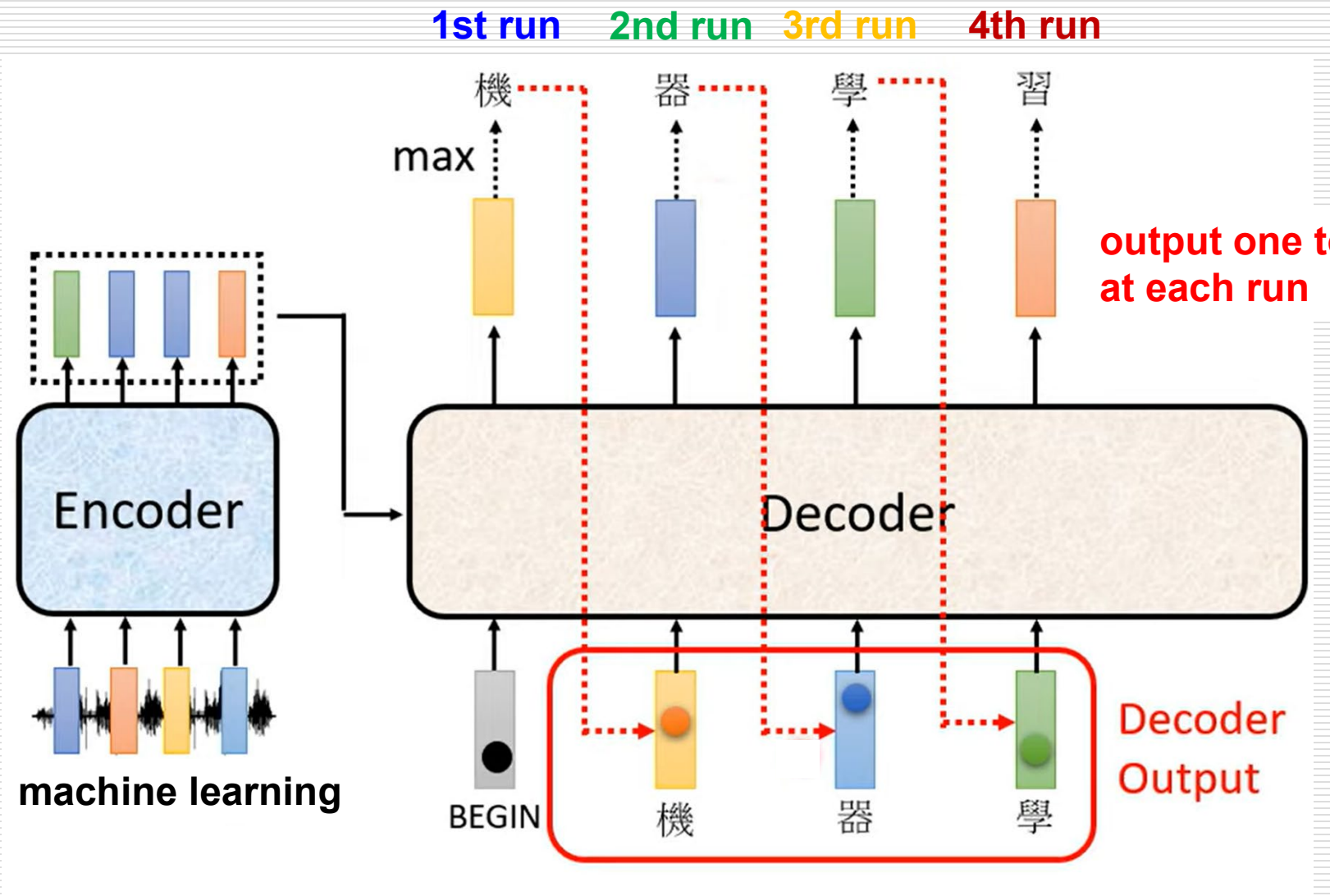
ViT
GPT-2
GPT-3

Autoregressive Decoder in Transformer

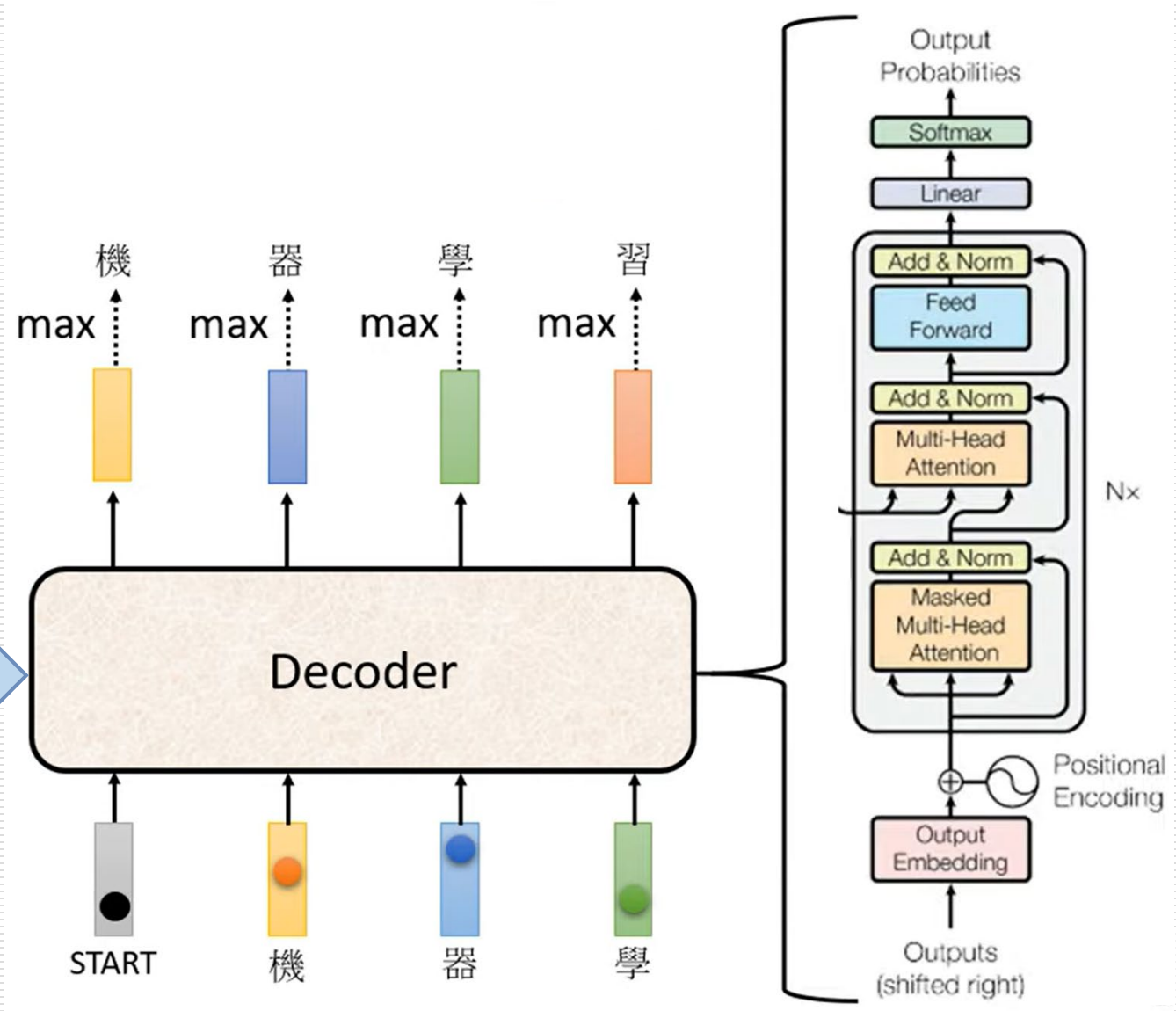
Autoregressive (AR) Decoder (1/2)



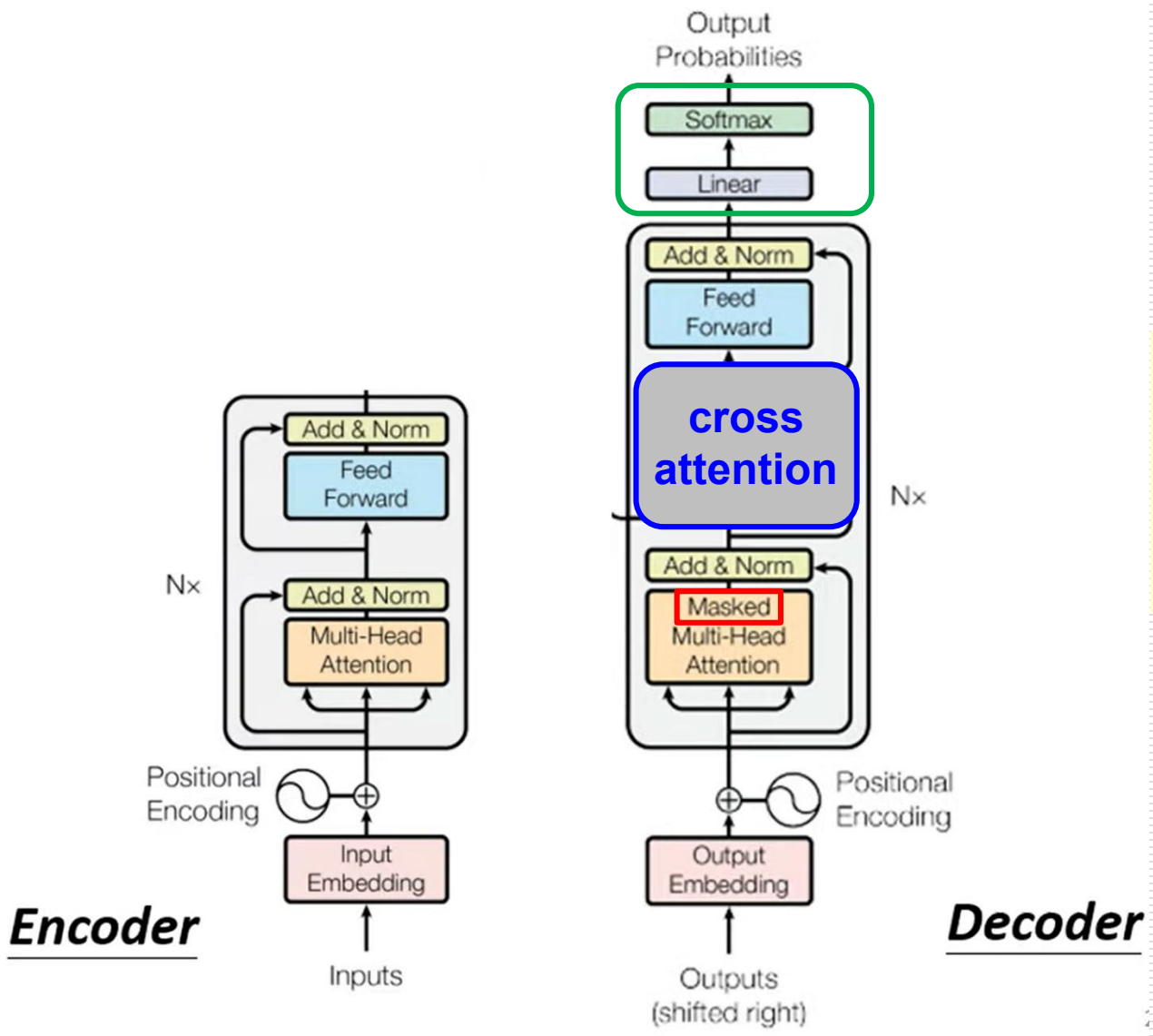
Autoregressive (AR) Decoder (2/2)



Decoder Implementation in Transformer

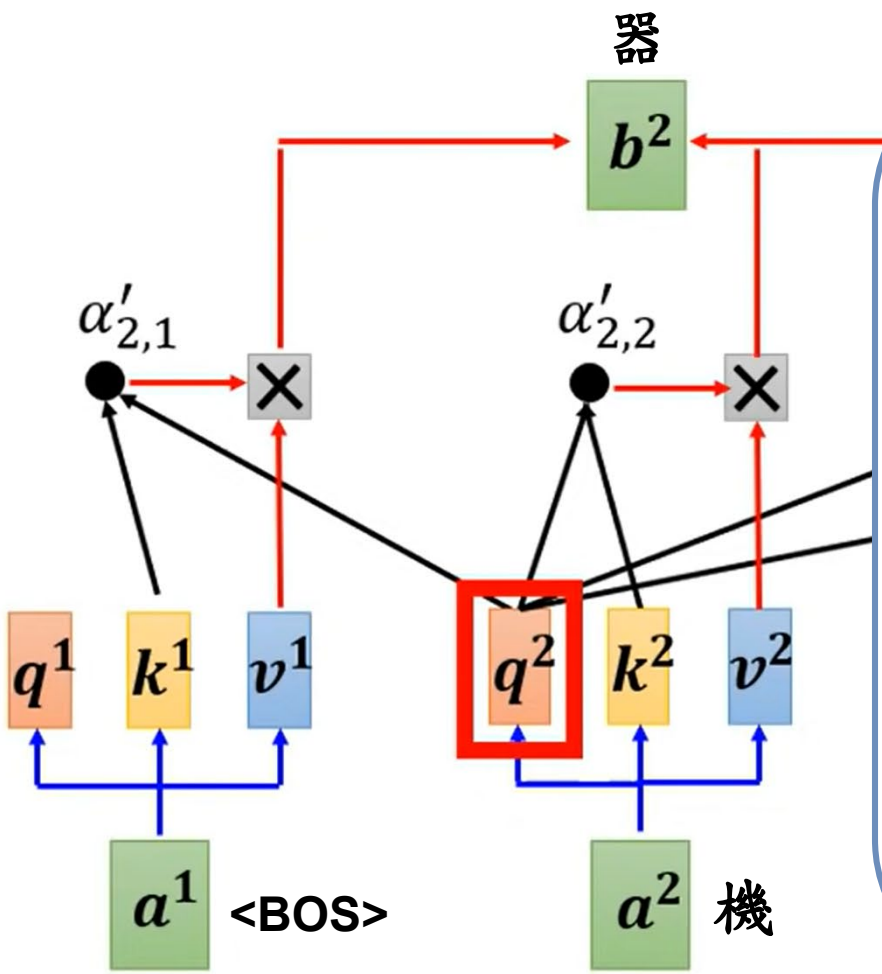


Encoder vs. Decoder



1. masked MSA
2. cross attention
3. multi-class output classifier (discussed previously)

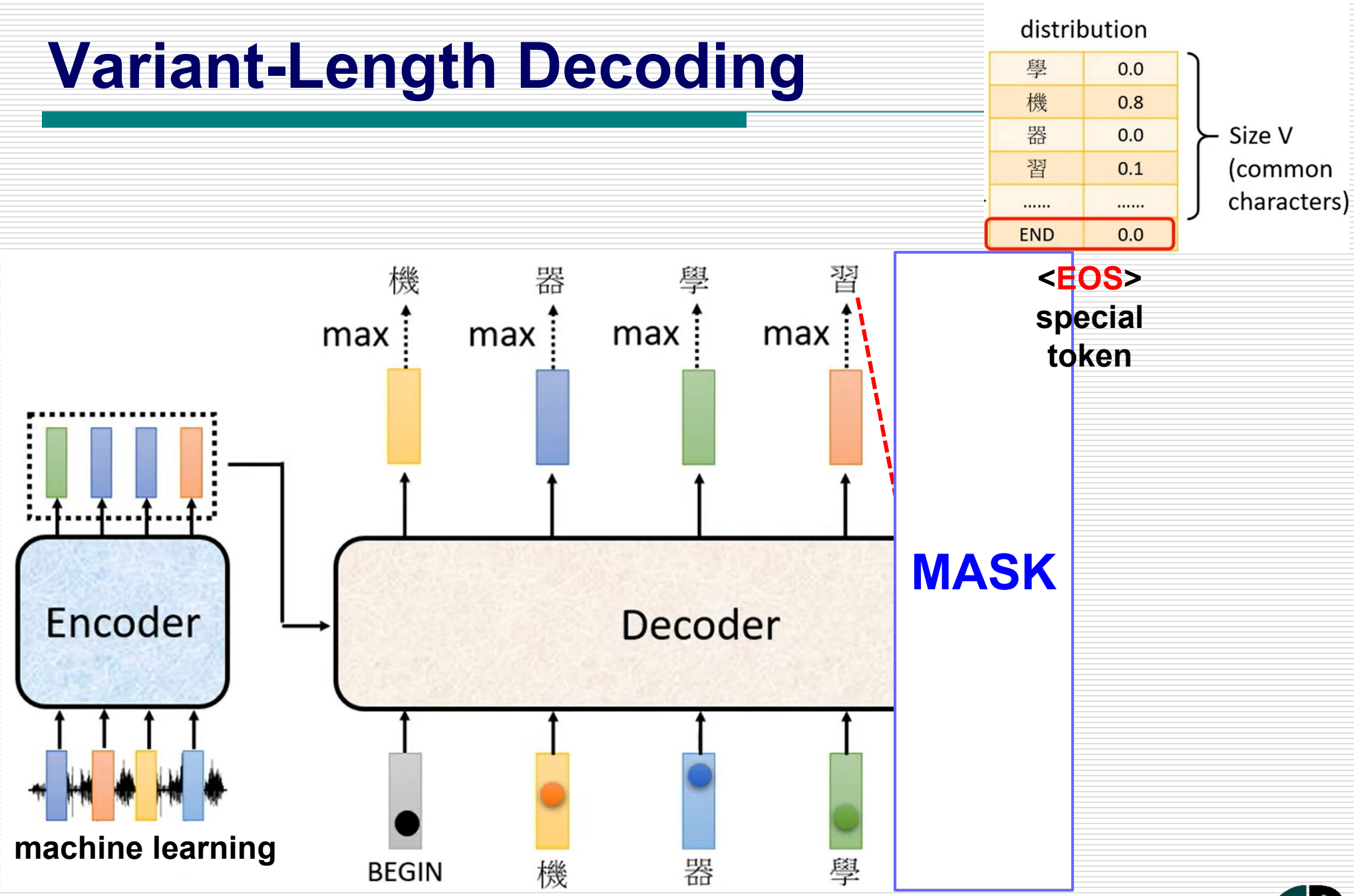
Masked Multi-Head Self-Attention



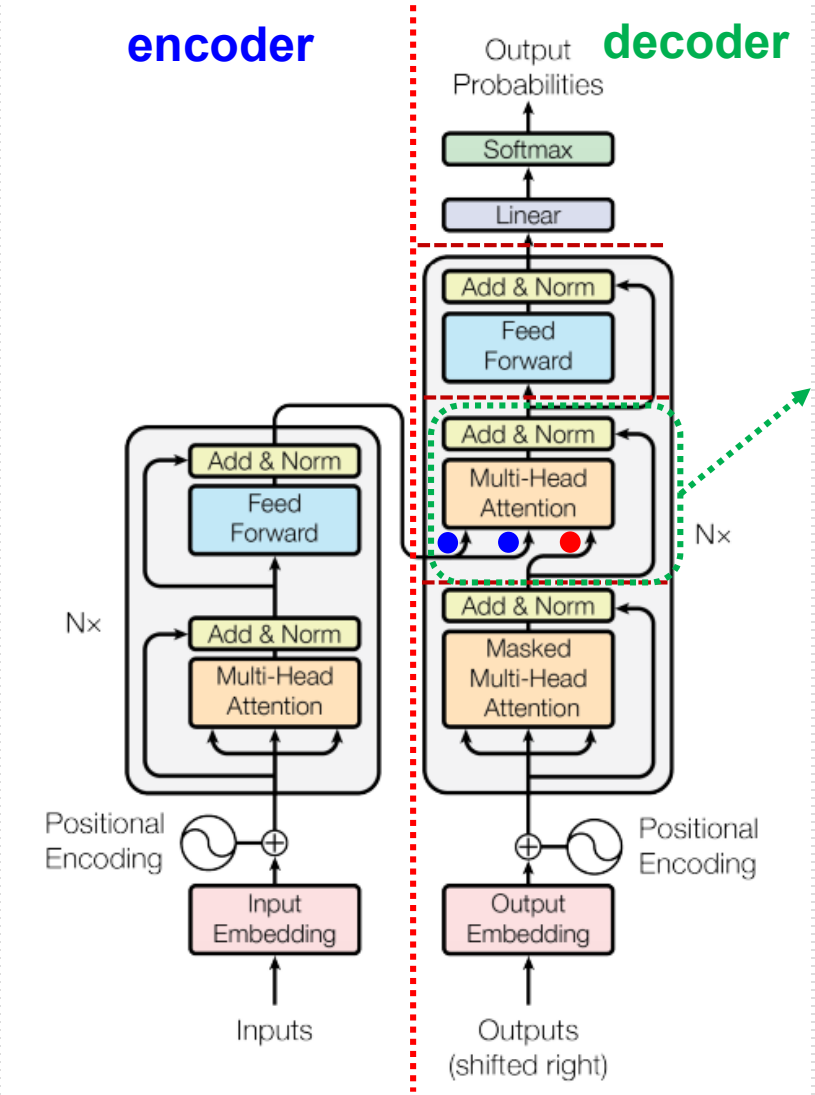
because of causality

MASK

Variant-Length Decoding



Cross Attention (1/2)

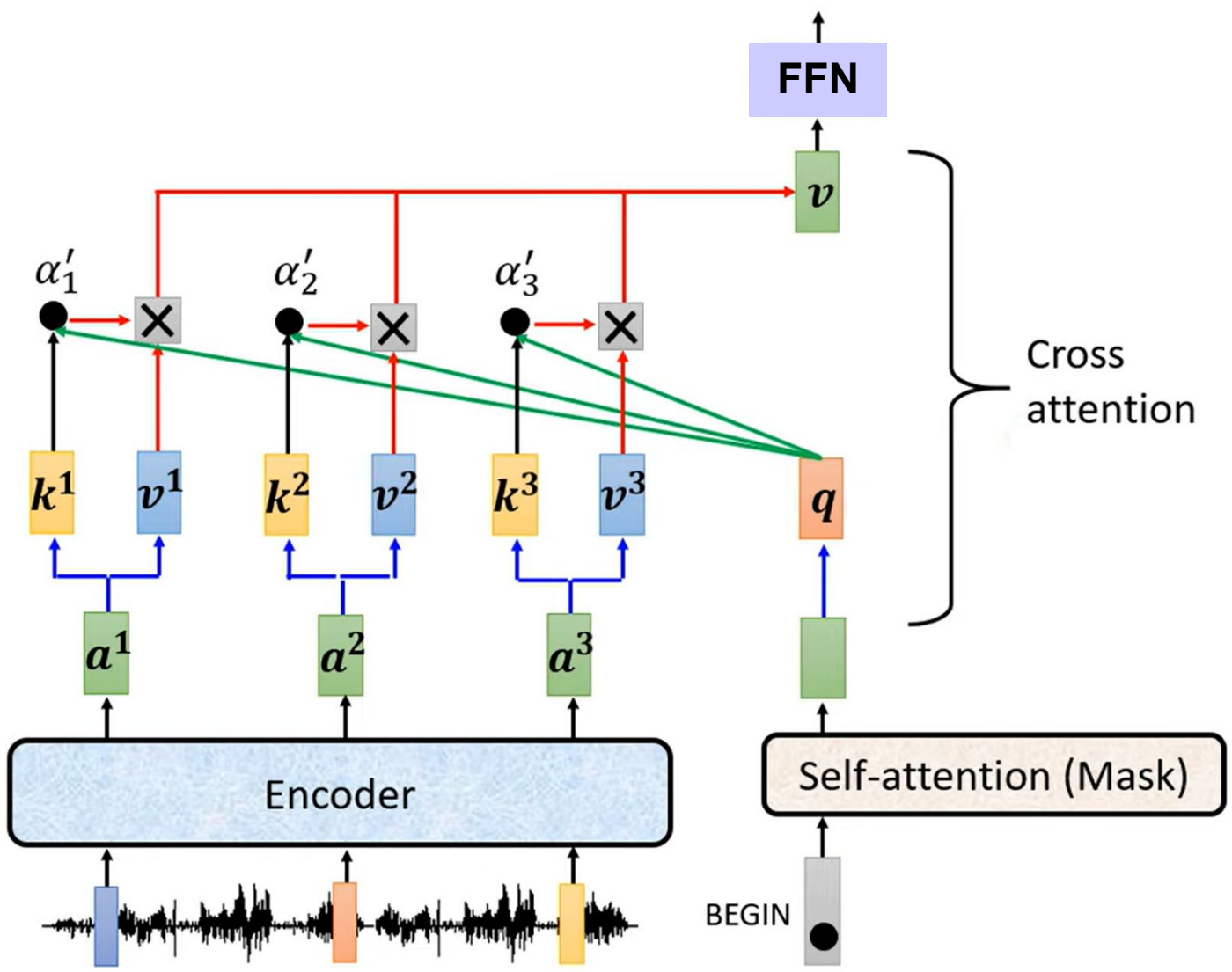


Cross Attention

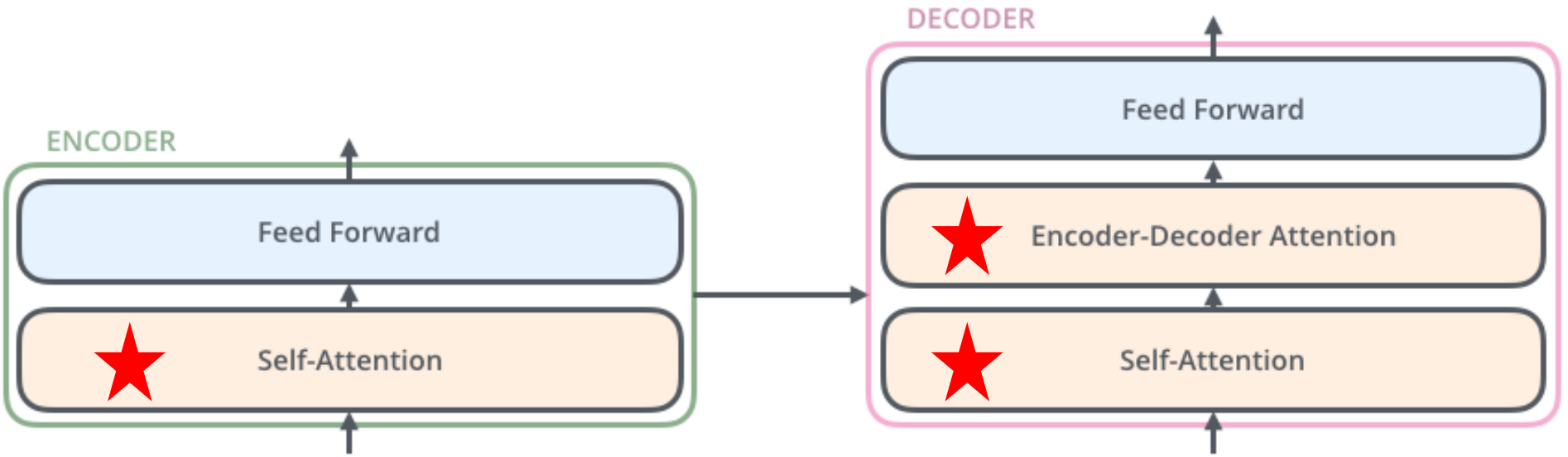
K, V from encoder

Q from decoder

Cross Attention (2/2)



Attention: Core of Transformer



Demo – MT by Transformer

source

Seq2Seq Is Versatile

Question Answering (QA)

Question	Context	Answer
What is a major importance of Southern California in relation to California and the US?	...Southern California is a major economic center for the state of California and the US....	major economic center
What is the translation from English to German?	Most of the planet is ocean water.	Der Großteil der Erde ist Meerwasser
What is the summary?	Harry Potter star Daniel Radcliffe gains access to a reported £320 million fortune...	Harry Potter star Daniel Radcliffe gets £320M fortune...
Hypothesis: Product and geography are what make cream skimming work. Entailment, neutral, or contradiction?	Premise: Conceptually cream skimming has two basic dimensions – product and geography.	Entailment
Is this sentence positive or negative? (sentiment analysis)	A stirring, funny and finally transporting re-imagining of Beauty and the Beast and 1930s horror film.	positive

QA can be done by **seq2seq**

https://arxiv.org/abs/1806.08730
https://arxiv.org/abs/1909.03329

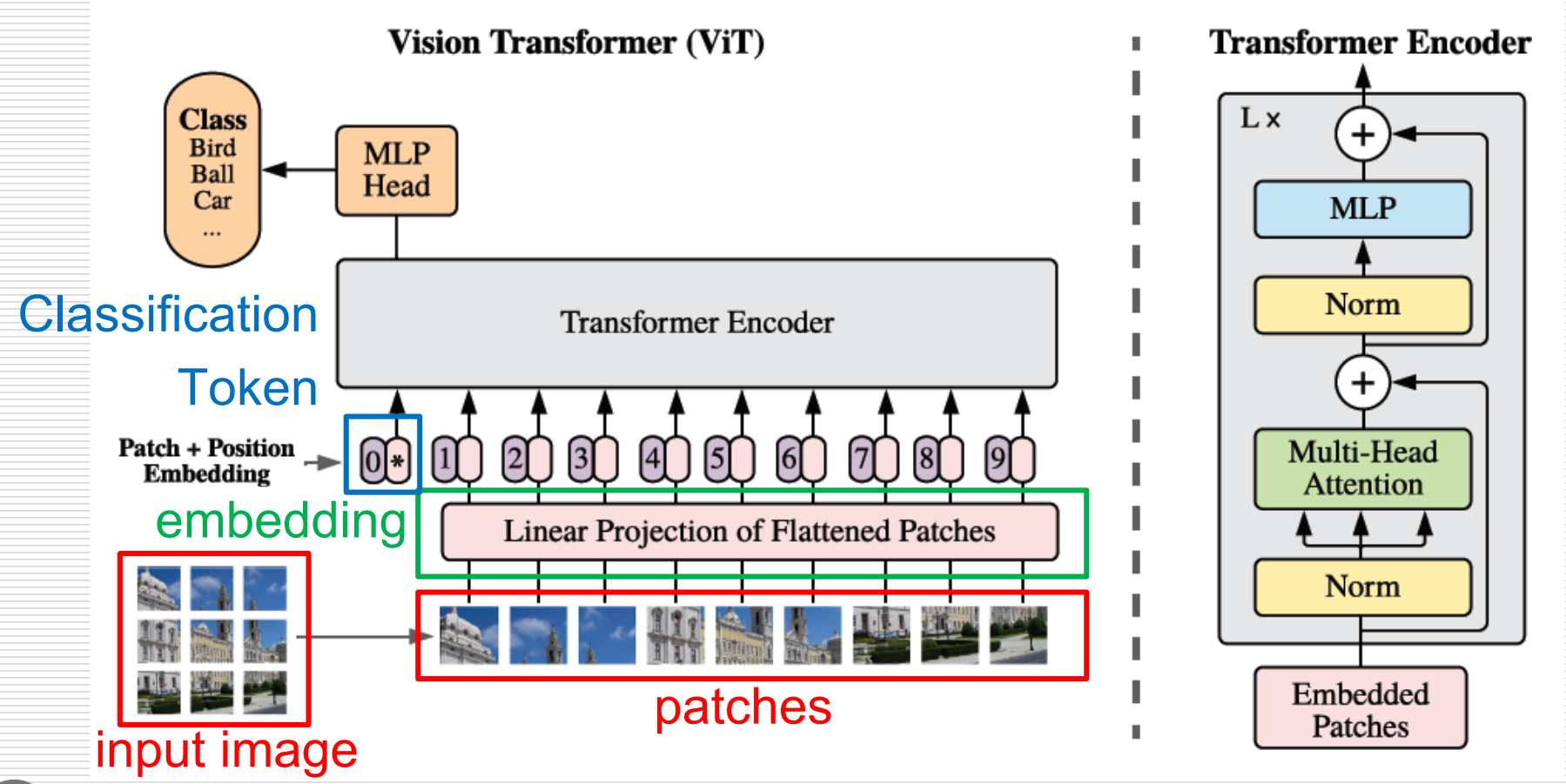
Transformer for Image Classification

Vision Transformer (ViT)

Google, Oct. 2020

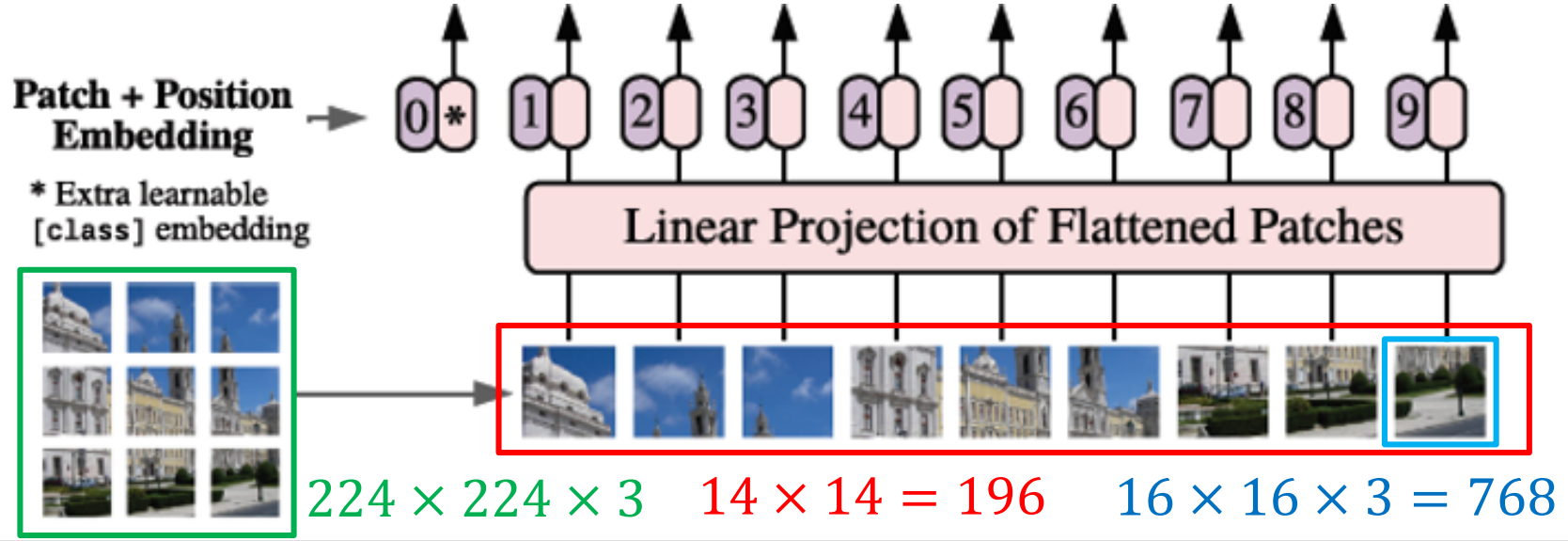
Vision Transformer (ViT)

- Proposed by Google in **2020**
  - a pure Transformer model (**NO** convolutions at all)
  - encoder **ONLY**



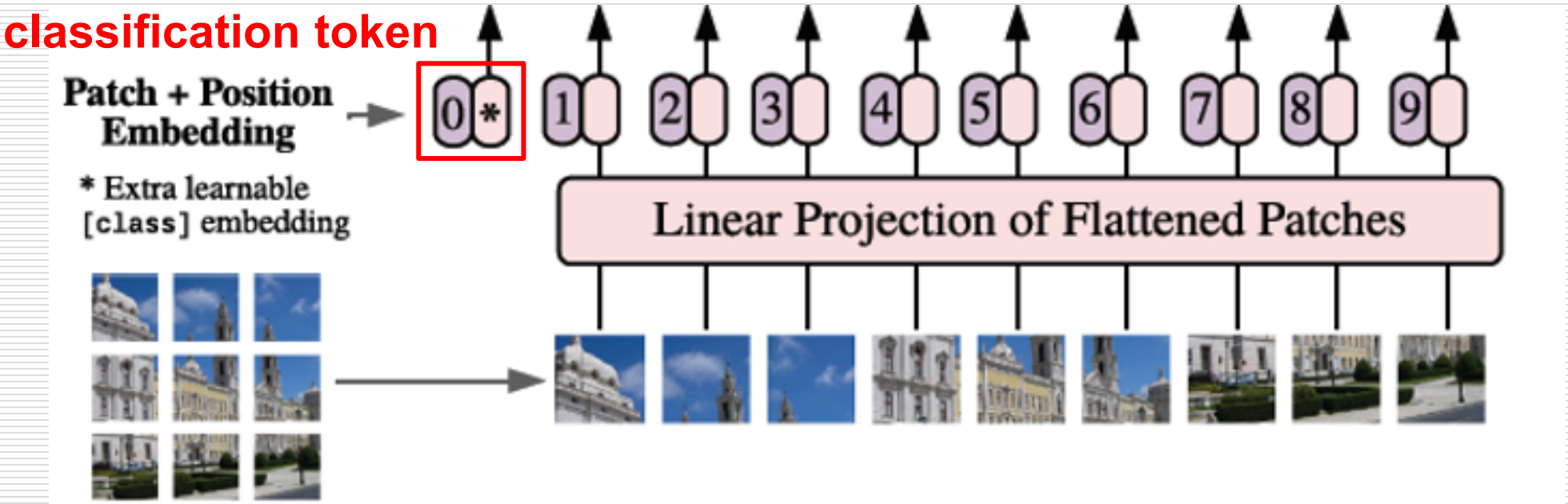
Convert an Image to a Sequence of Vectors

- Input image to a sequence of vectors
  - Split an image (H,W,C) into fixed-size patches (P,P,C)
  - Example: Input (224,224,3) → **Patch** (16,16,3) × 196



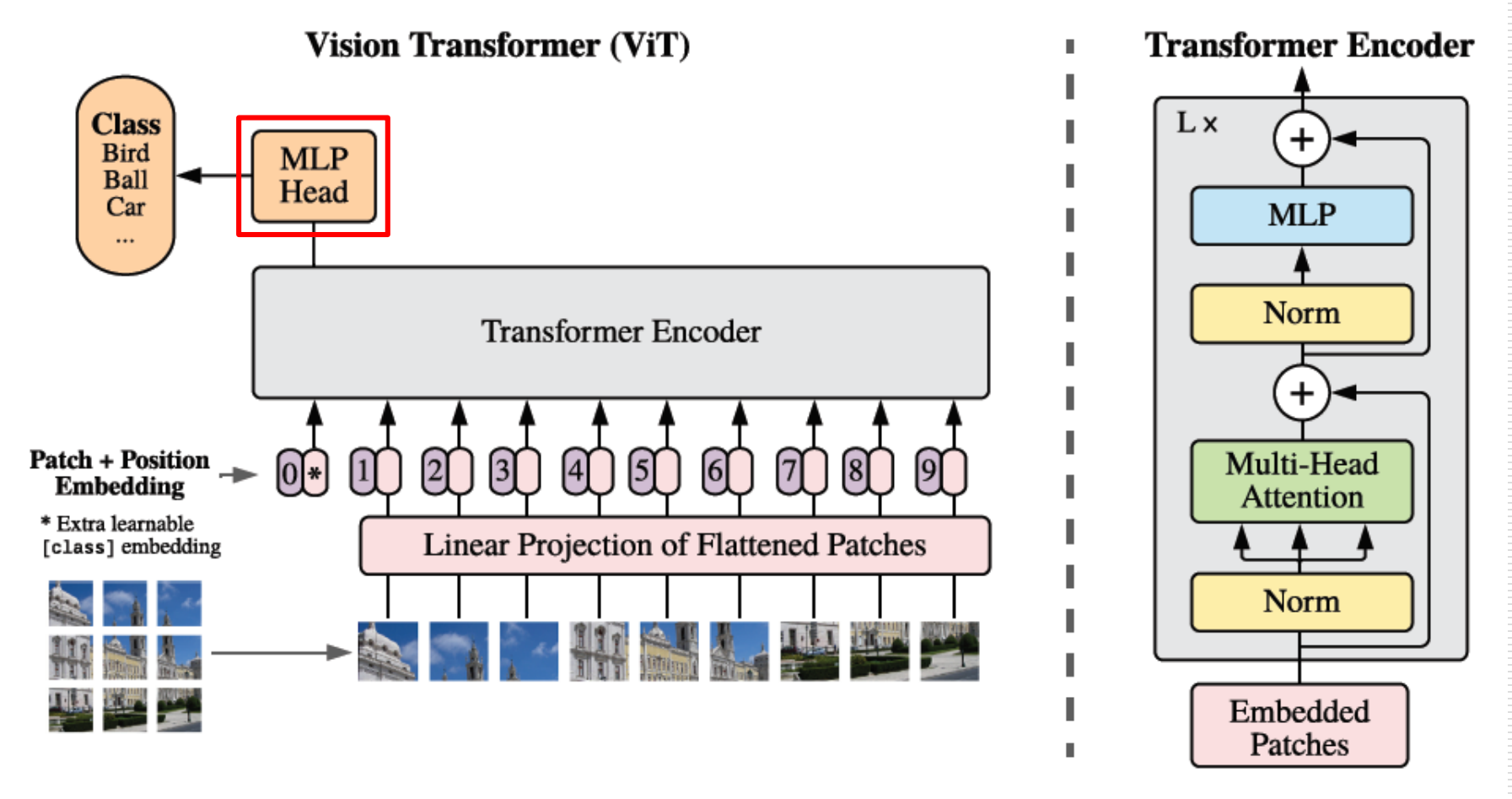
Classification Token

- Add an extra learnable "classification token"
  - Similar to BERT's [class] token
- Add learnable position embedding to retain position information



Multi-Class Output Classifier

- MLP Head
  - Use the classification token of last layer for classification



Demo System: Pose Estimation (1/2)

Demo System: Pose Estimation (2/2)

- **End-to-End** Transformer accelerator on FPGA:
  1024 MACs @ **300MHz** (**MAX speed**) 🡐 **600 GOPS**
- **Softmax/LayerNorm/GELU** solved by dedicated HW logic
- **Zero runtime overhead matrix transpose**
- **Native asymmetric quantization** support
- NN model: **PE-former** (image size: **256x192**)
- **FPS**: **40.1** (**camera in**) / **53.2** (**video in**)
- **No GPUs / No Cloud Servers** in use

Xilinx
ZCU-106

Takeaways

- Transformer is the king in NLP and CV (and in most DL application domains) now and in a foreseeable future
  - RNN and LSTM dead ; CNN struggling
- Self-attention is the key to the success of Transformer
  - computation is highly parallel
  - long-range dependency is no longer an issue
- Transformer adopts the encoder + AR decoder architecture
- Transformer serves as the foundation of BERT and GPT-x
- It's extremely challenging to design a datacenter-scale Transformer accelerator
  - issues: utilization rate, external memory BW, internal buffer capacity, matrix transpose, Softmax, LayerNorm, GELU, quantization, workload partitioning and scheduling, … (just name a few)

More About Transformer

- 台大李宏毅教授的 Transformer 上課影片
  - https://youtu.be/hYdO9CscNes，【機器學習2021】Self-attention (上)
  - https://youtu.be/gmsMY5kc-zw，【機器學習2021】Self-attention (下)
  - https://youtu.be/n9TlOhRjYoc，【機器學習2021】Transformer (上)
  - https://youtu.be/N6aRv06iv2g，【機器學習2021】Transformer (下)
- 台大李宏毅教授的 Transformer 解說文章
  - https://hackmd.io/@abliu/BkXmzDBmr
- Lee Meng 的部落格文章 “淺談神經機器翻譯與 Transformer”
  - https://leemeng.tw/neural-machine-translation-with-transformer-and-tensorflow2.html
- “Attention is all you need” – ArXiv paper
  - https://arxiv.org/abs/1706.03762

Thank you