Attention Mechanism and Transformer

FA690 Machine Learning in Finance

Dr. Zonghao Yang

2026 Spring

Learning Objective

- Understand the self-attention mechanism and how it enables models to focus on relevant parts of input sequences
- Explore the architecture of Transformer models, including encoders, decoders, multi-head attention, cross attention
- Compare the advantages of Transformer over RNNs, particularly regarding parallel processing and long-range dependencies
- Examine transformer-based models like BERT and GPT, and understand how self-supervised pre-training enables these models to learn from massive datasets

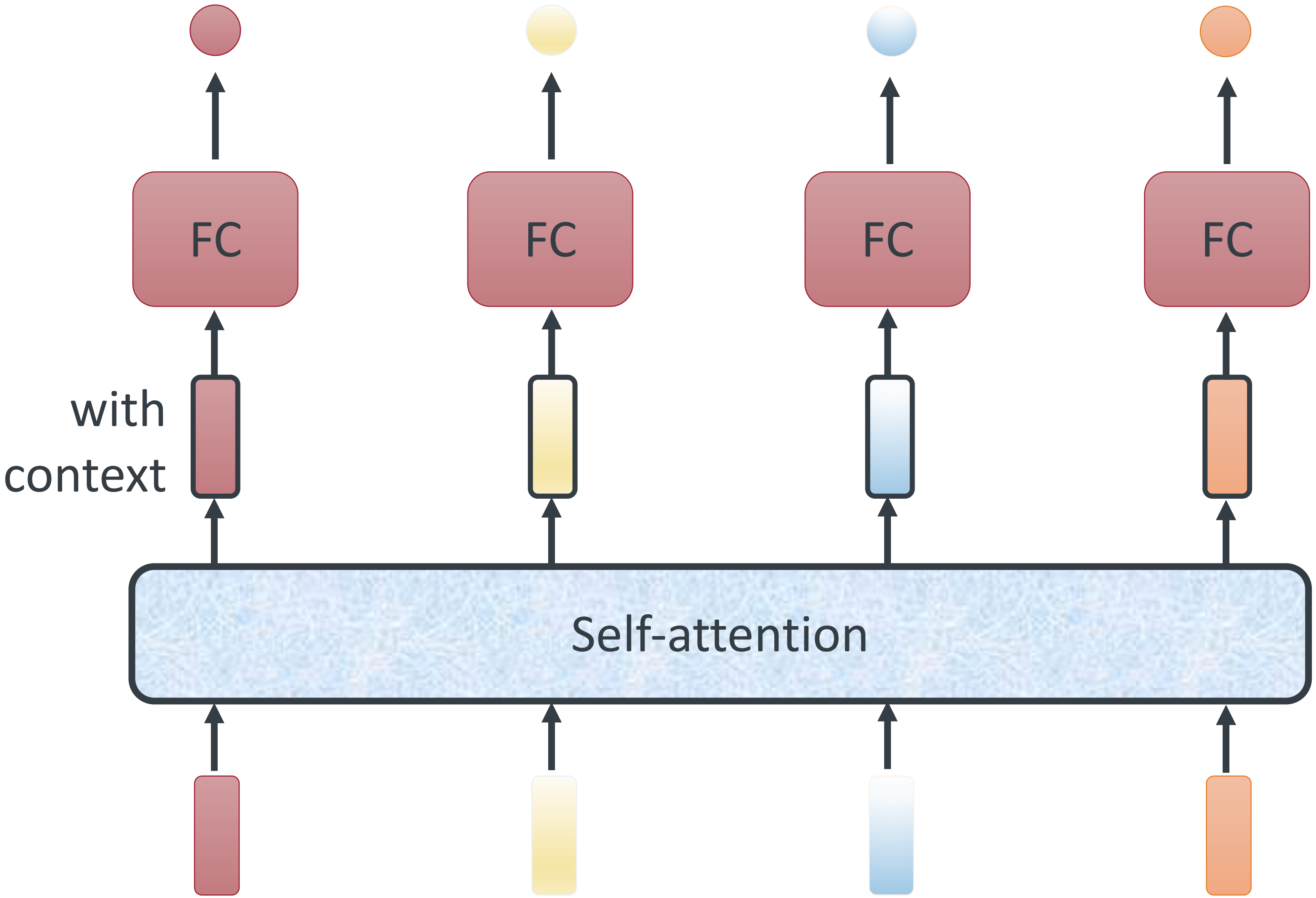
Self Attention

Motivating Example

- Sentence completion: When she tried to print her tickets, she found that the printer was out of toner. She went to the stationery store to buy more toner. It was very overpriced. After installing the toner into the printer, she finally printed her ____.

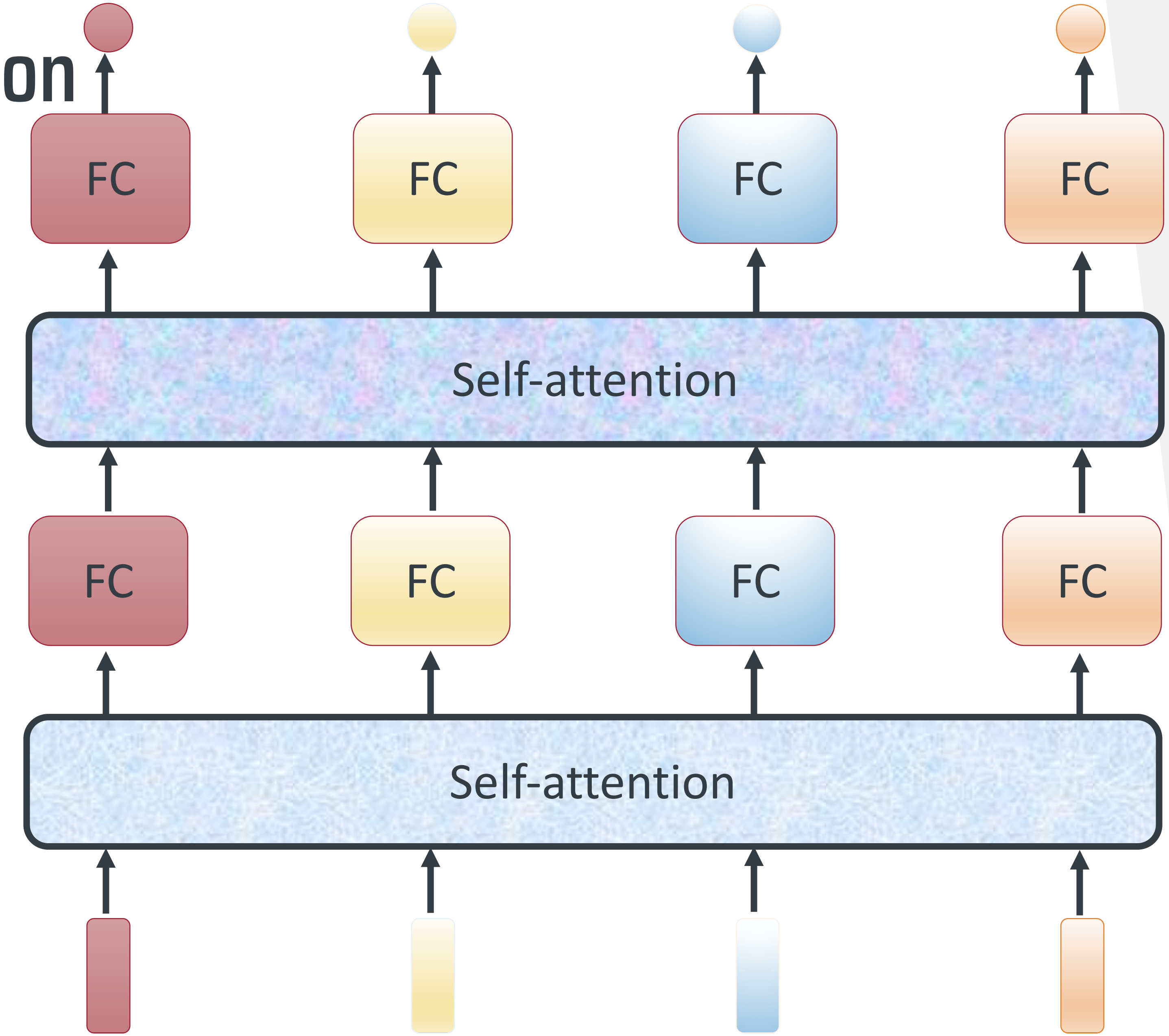
- To learn from this training example, the RNN needs to model the dependency between "tickets" on the 7th step and the target word "tickets" at the end
- Vanilla RNN: Vanishing gradient problem
- LSTM: The design of three gates (input, forget, and output gates) to keep both long-term and short-term memories
- Transformer: Input the entire sequence; the model learns where to pay attention to

Self-attention

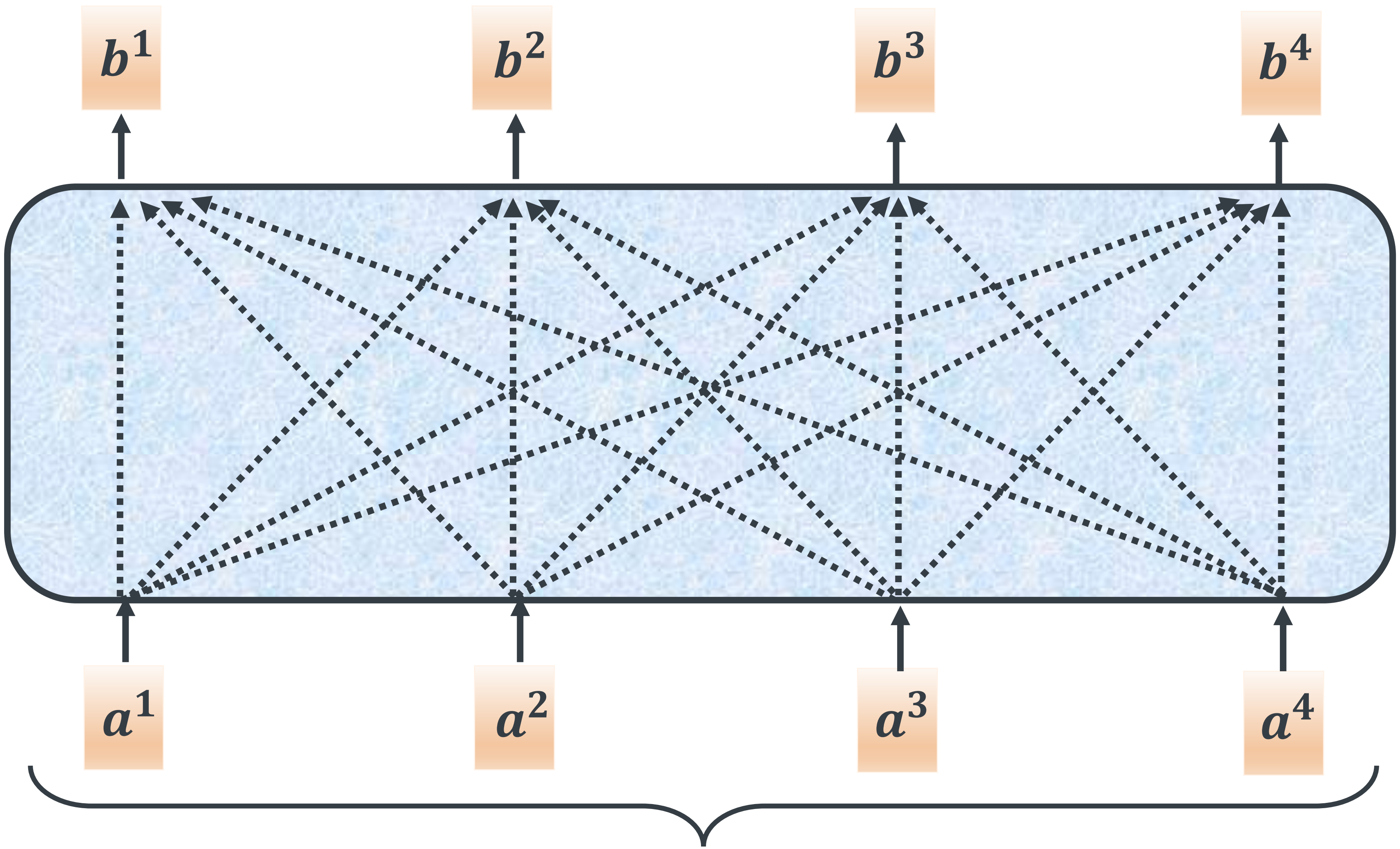


Multi-layer Self-attention

Attention is all you need
A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc
… to attend to **all** positions in the decoder up to and including that position. **We need** to prevent
… **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) …

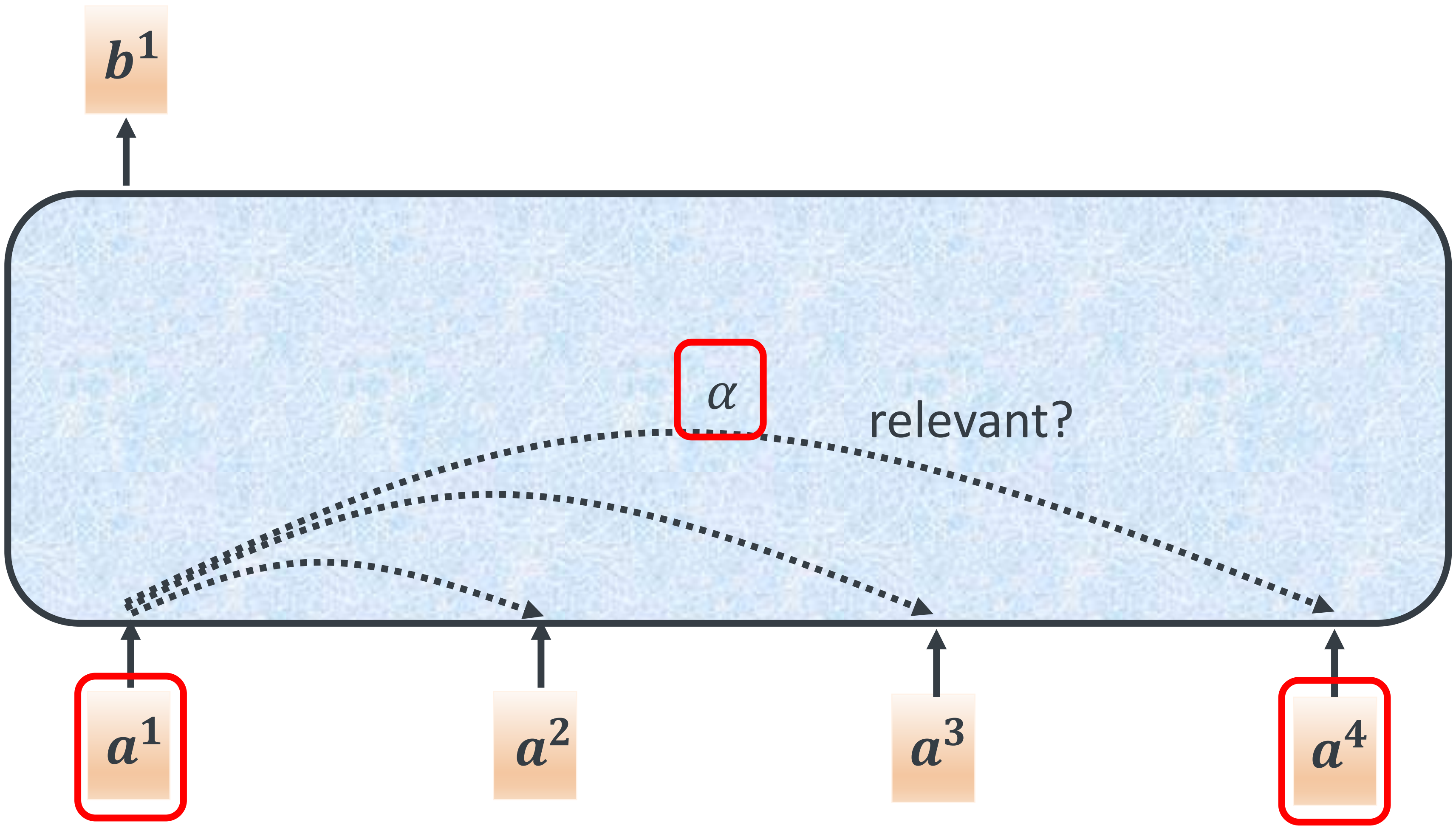


Self-attention



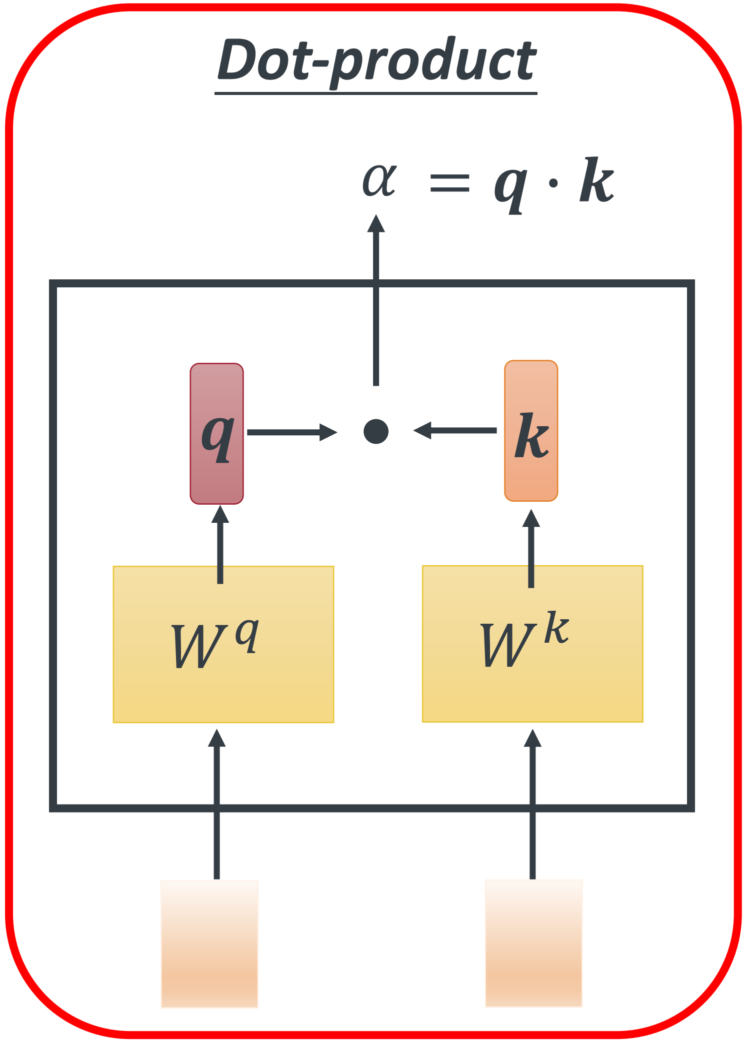
Can be either **input** or **a hidden layer**

Self-attention



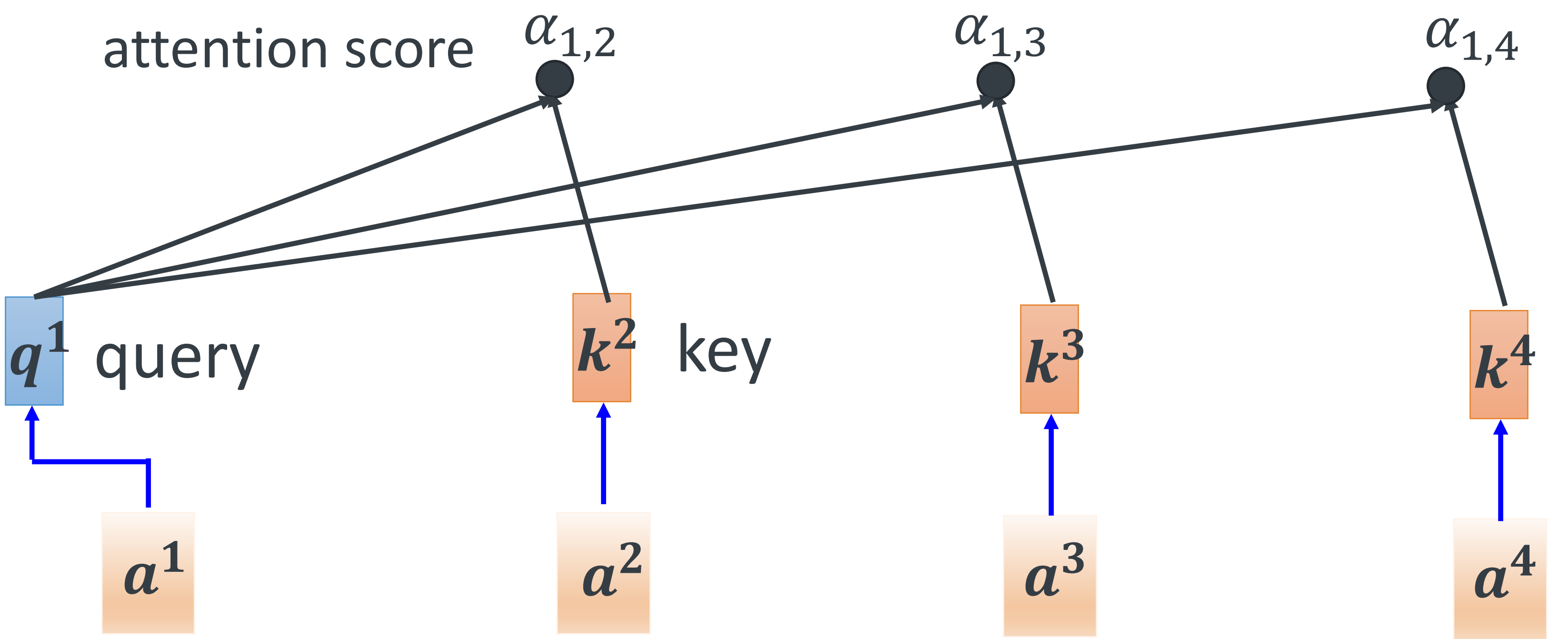
Find the relevant vectors in a sequence

Attention Measure



Self-attention

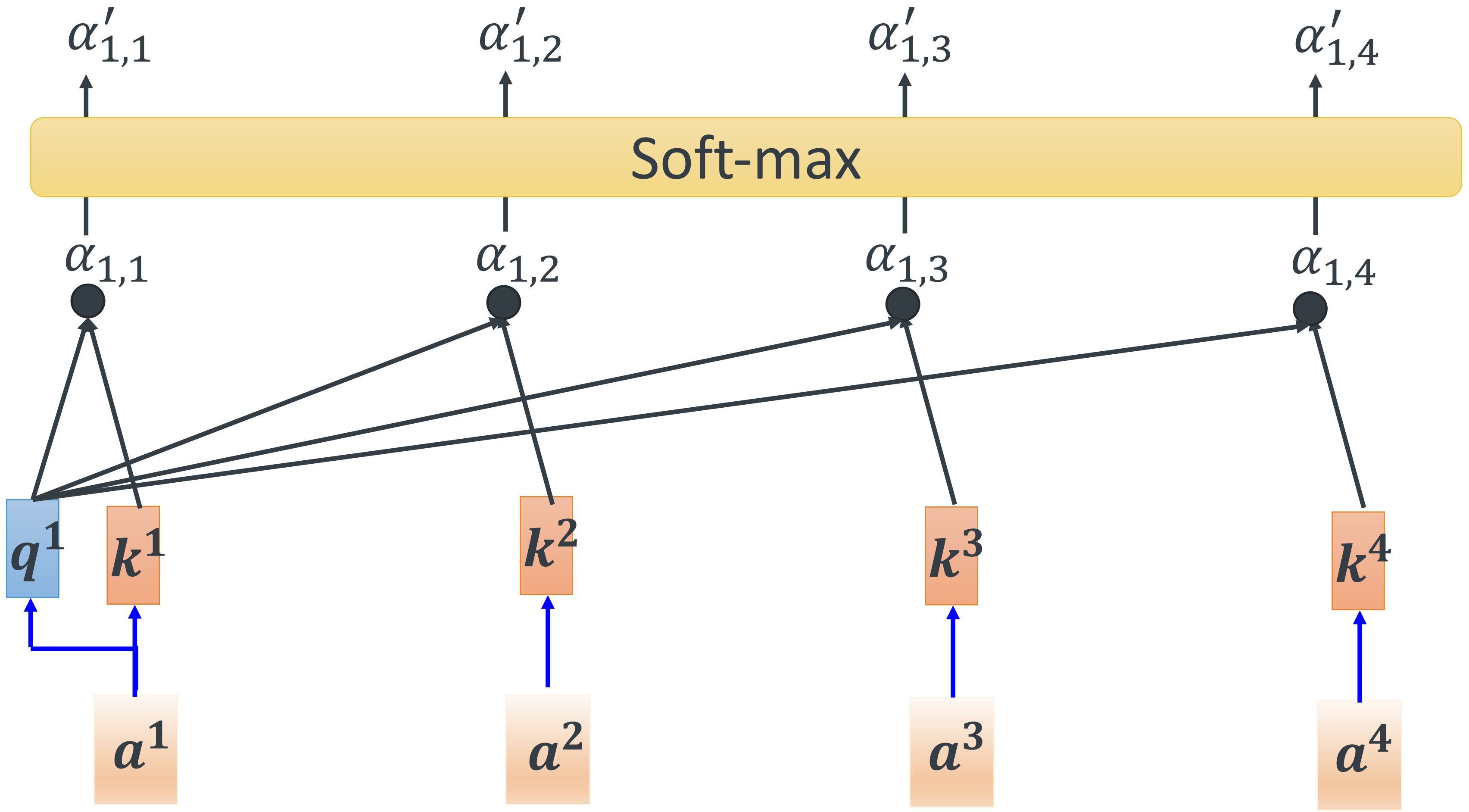
$\alpha_{1,2} = \boldsymbol{q}^1 \cdot \boldsymbol{k}^2 \qquad \alpha_{1,3} = \boldsymbol{q}^1 \cdot \boldsymbol{k}^3 \qquad \alpha_{1,4} = \boldsymbol{q}^1 \cdot \boldsymbol{k}^4$



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

Self-attention

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



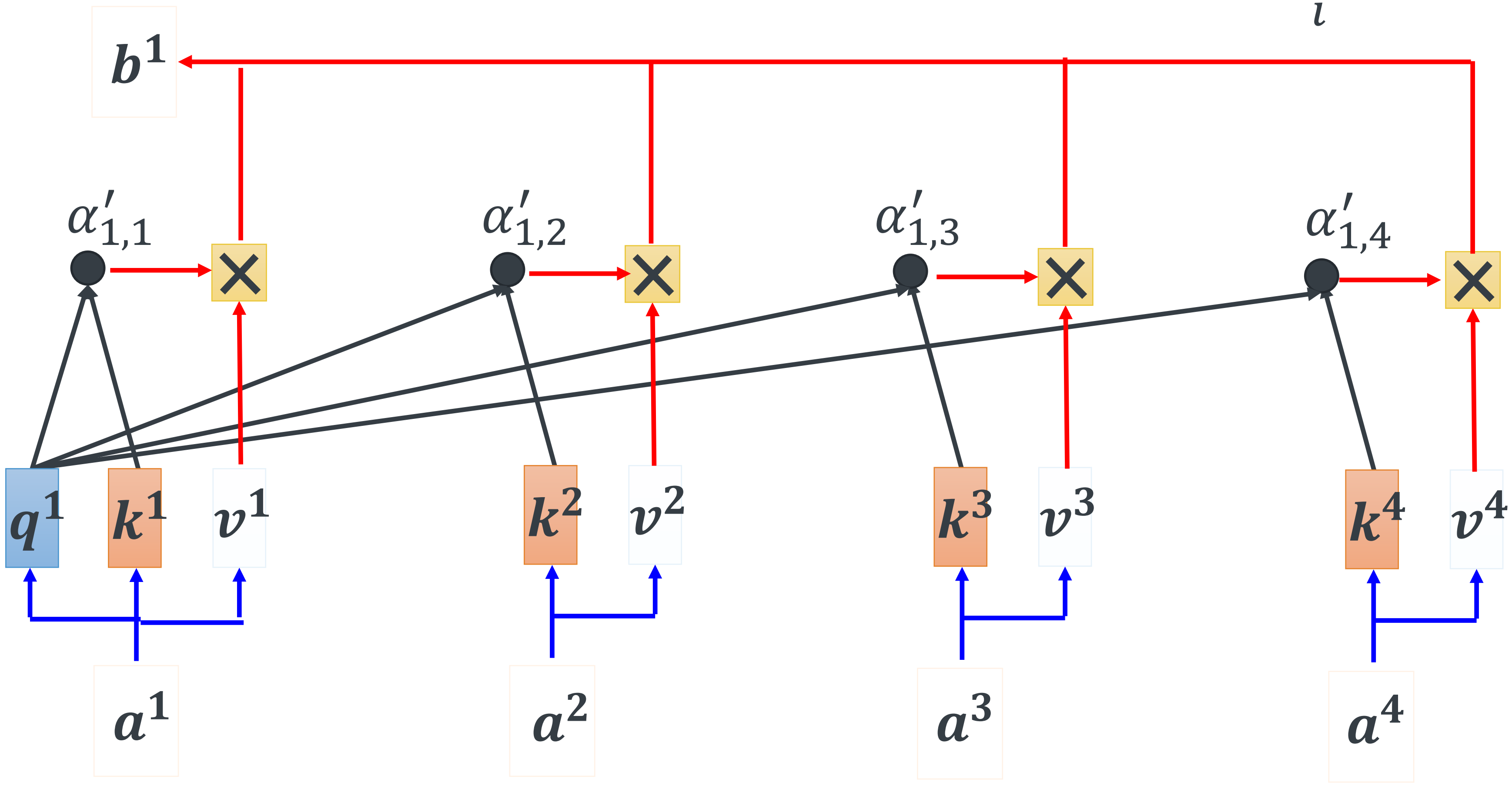
$q^1 = W^q a^1$ $\quad k^2 = W^k a^2$ $\quad k^3 = W^k a^3$ $\quad k^4 = W^k a^4$

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

Self-attention

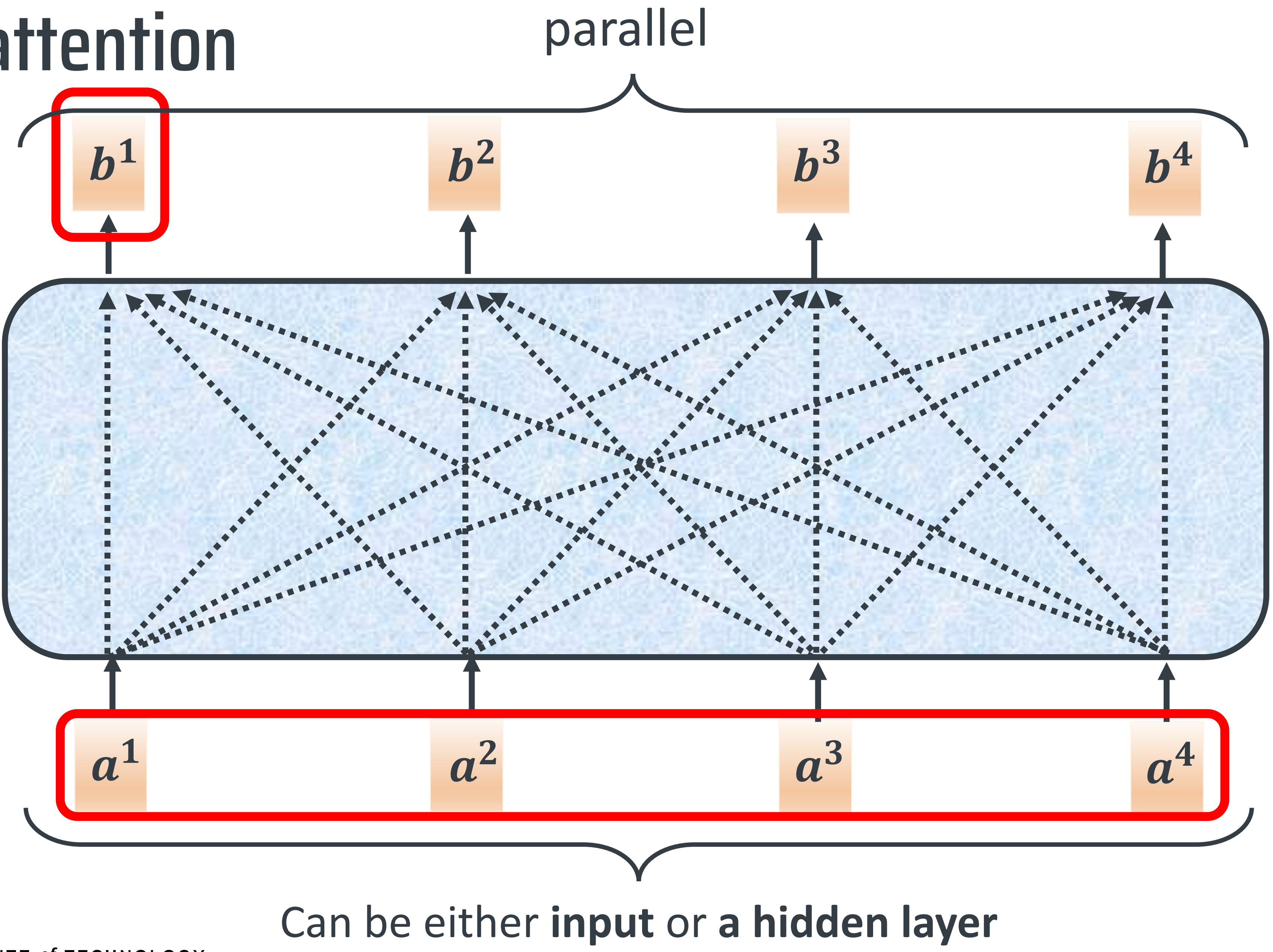
Extract information based on attention scores

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



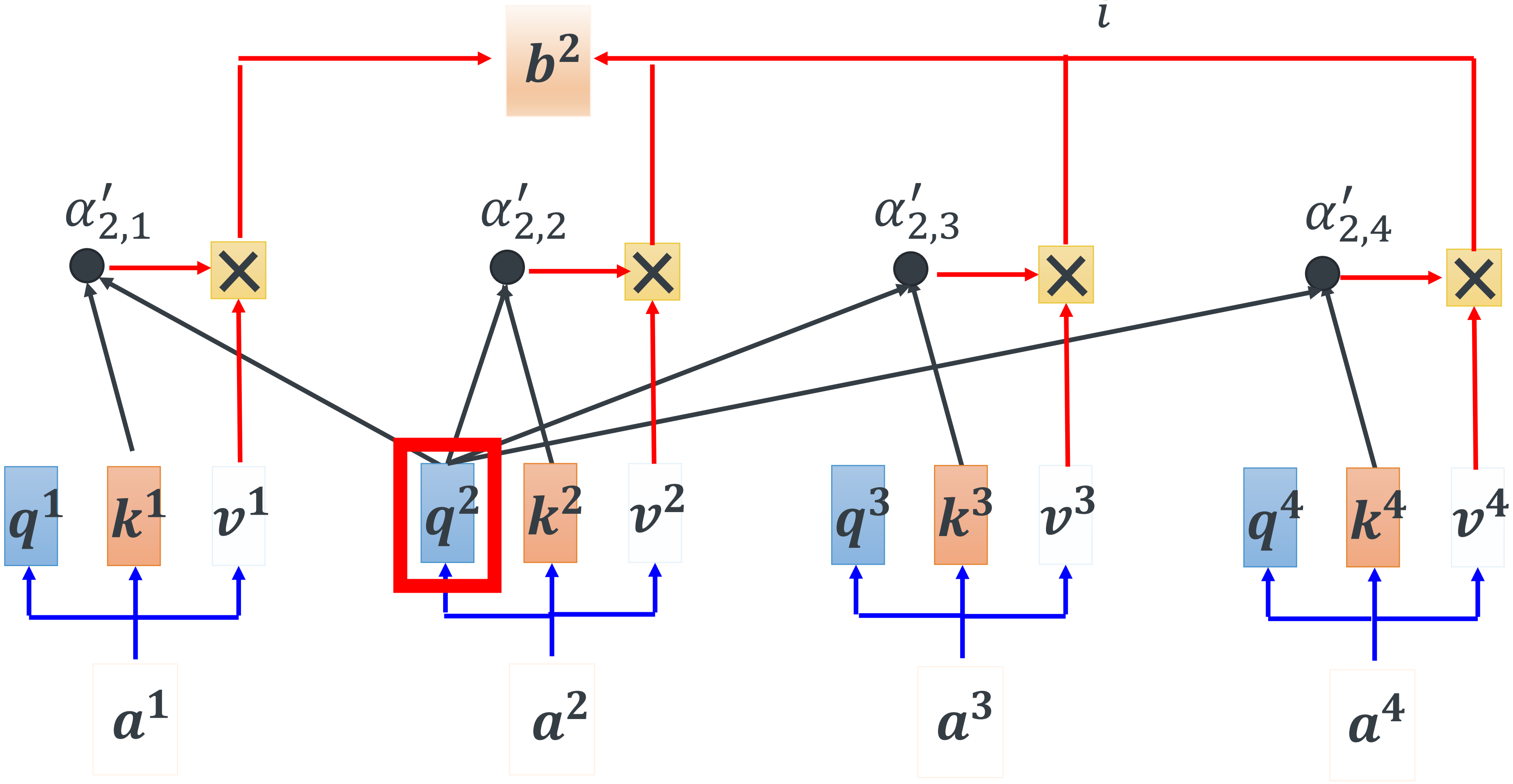
$v^1 = W^v a^1$ $\quad v^2 = W^v a^2$ $\quad v^3 = W^v a^3$ $\quad v^4 = W^v a^4$

Self-attention



Self-attention

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

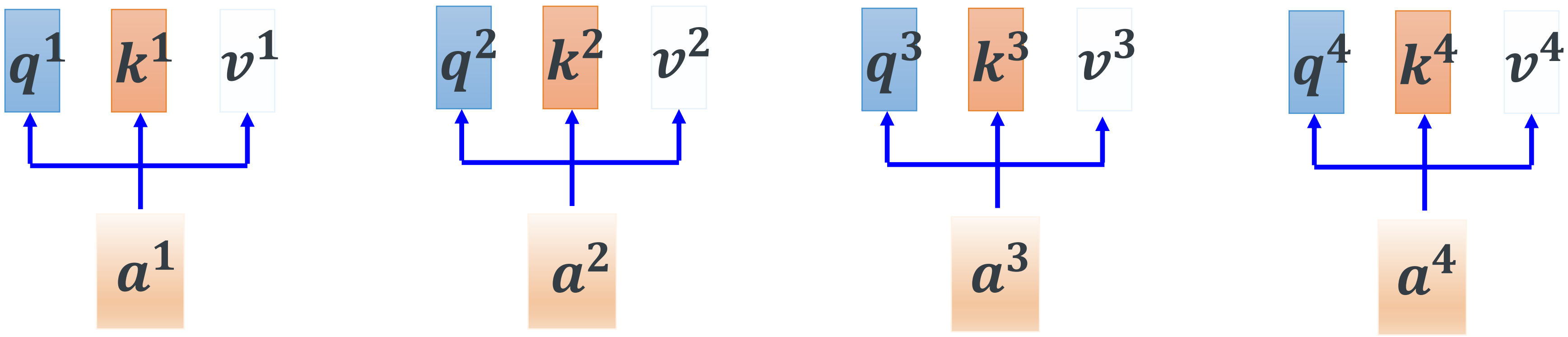


Self-attention

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

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

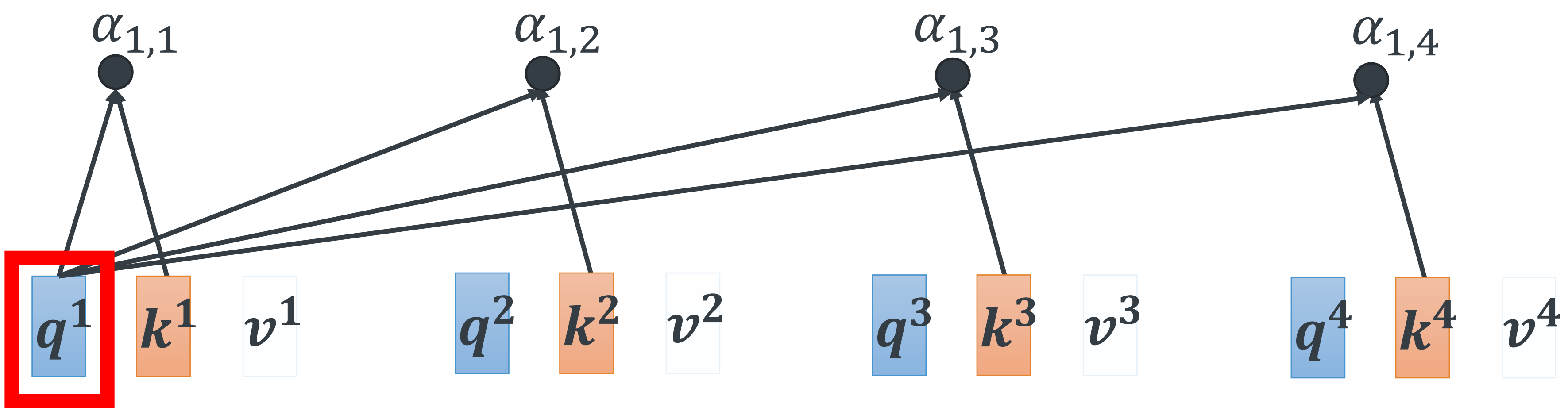
$$v^i = W^v a^i \qquad \underset{V}{v^1 v^2 v^3 v^4} = W^v \underset{I}{a^1 a^2 a^3 a^4}$$



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

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

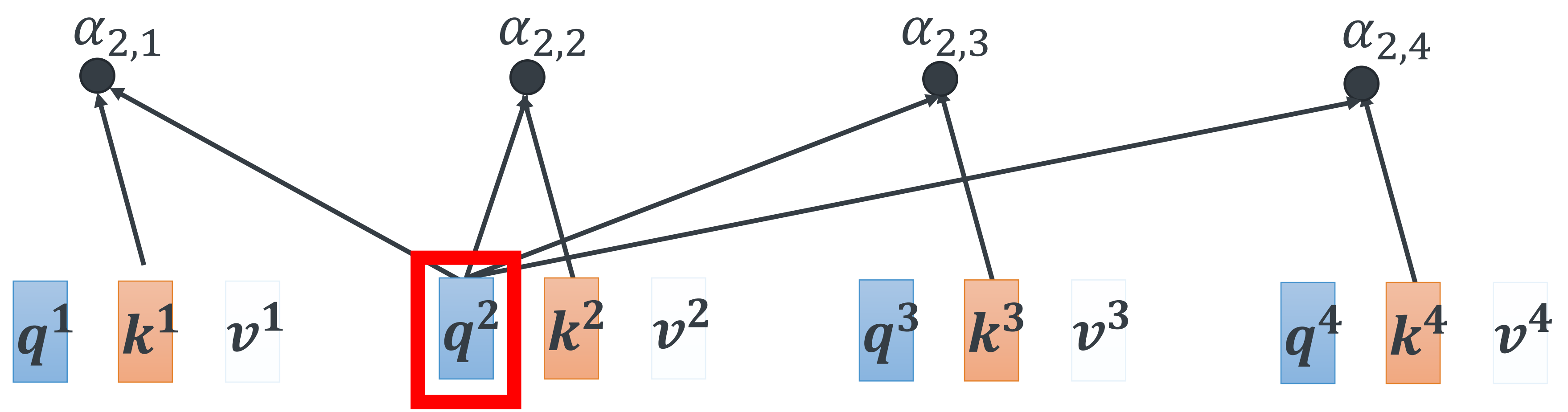
$$\begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}^1 \\ \boldsymbol{k}^2 \\ \boldsymbol{k}^3 \\ \boldsymbol{k}^4 \end{bmatrix} \boldsymbol{q}^1$$



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

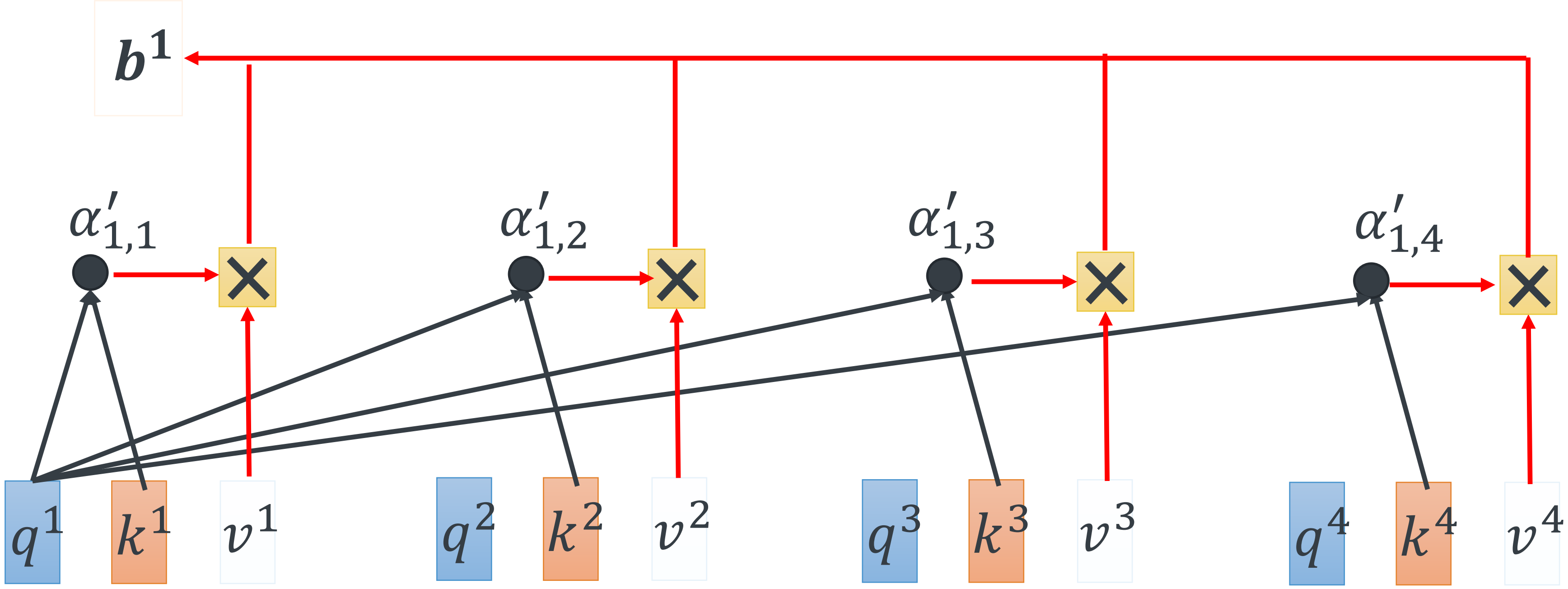
$$\alpha_{1,3} = k^3 \; q^1 \qquad \alpha_{1,4} = k^4 \; q^1$$

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



$$\underset{A'}{\begin{bmatrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{bmatrix}} \xleftarrow{\text{softmax}} \underset{A}{\begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \alpha_{4,1} \\ \alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \alpha_{4,2} \\ \alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \alpha_{4,3} \\ \alpha_{1,4} & \alpha_{2,4} & \alpha_{3,4} & \alpha_{4,4} \end{bmatrix}} = \underset{K^T}{\begin{bmatrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{bmatrix}} \underset{Q}{\begin{bmatrix} q^1 & q^2 & q^3 & q^4 \end{bmatrix}}$$

Self-attention



$$\underbrace{\begin{bmatrix} b^1 & b^2 & b^3 & b^4 \end{bmatrix}}_{O} = \underbrace{\begin{bmatrix} v^1 & v^2 & v^3 & v^4 \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} \alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1} \\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2} \\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3} \\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4} \end{bmatrix}}_{A'}$$

Self-attention

$$Q = W^q \, I$$

$$K = W^k \, I$$

$$V = W^v \, I$$

Parameters to be learned

$$A' \leftarrow A = K^T Q$$

Attention Matrix

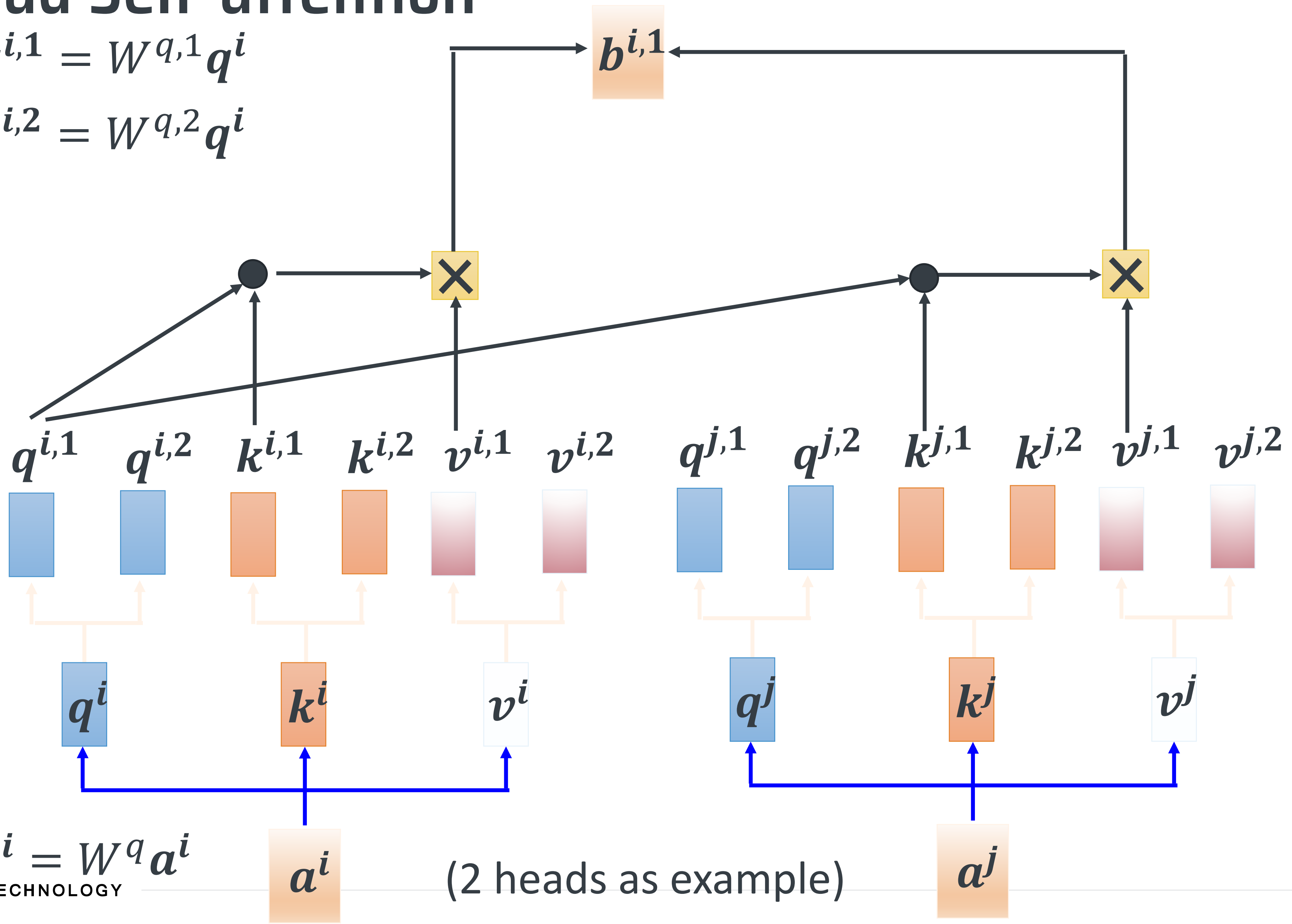
$$O = V \, A'$$

Multi-head Self-attention

Different types of relevance

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

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



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

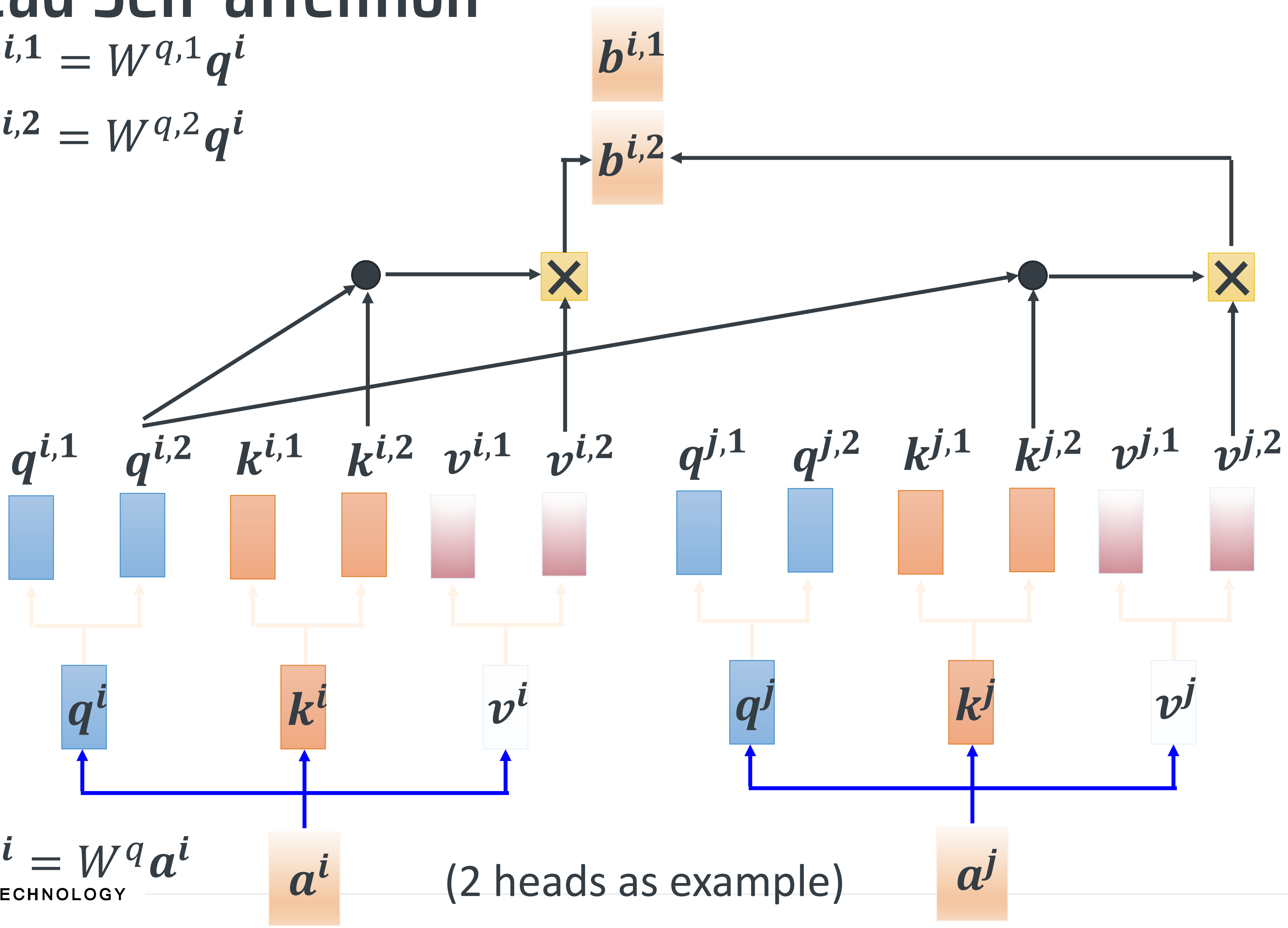
(2 heads as example)

Multi-head Self-attention

Different types of relevance

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

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

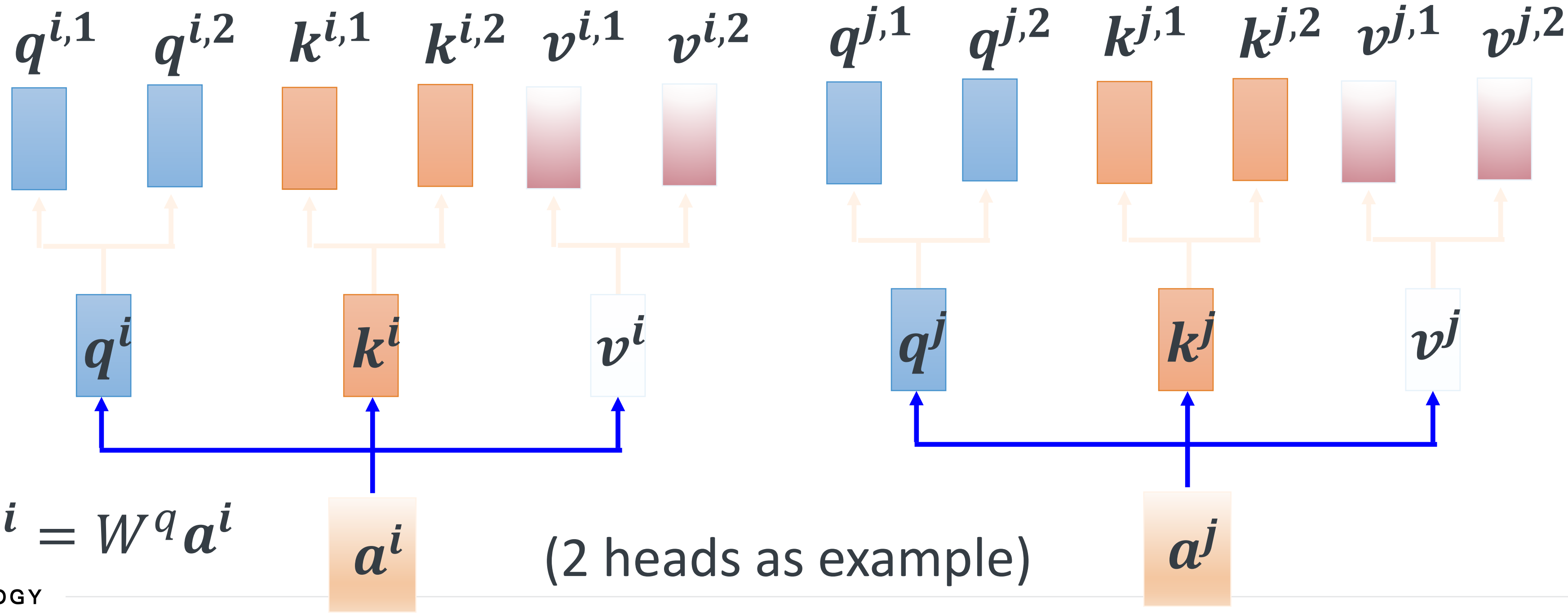


$$q^i = W^q a^i$$

Multi-head Self-attention

Different types of relevance

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

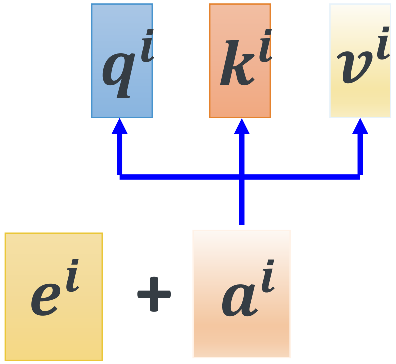


$q^i = W^q a^i$

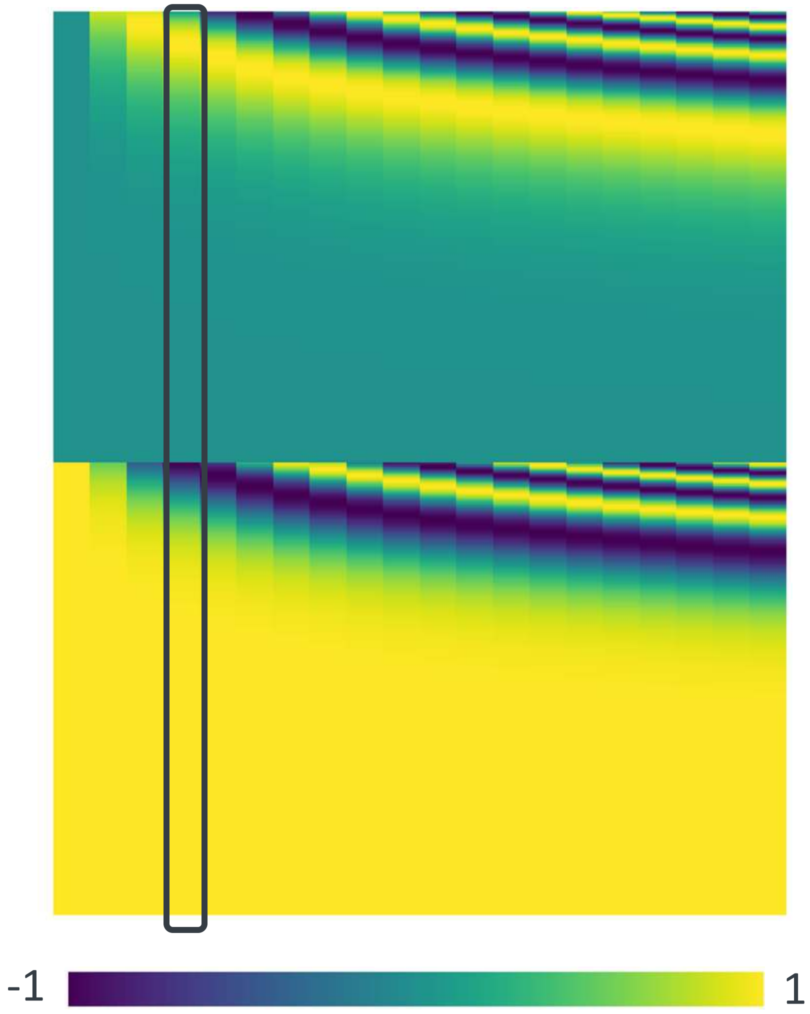
(2 heads as example)

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



Applications of Self-attention

Transformer

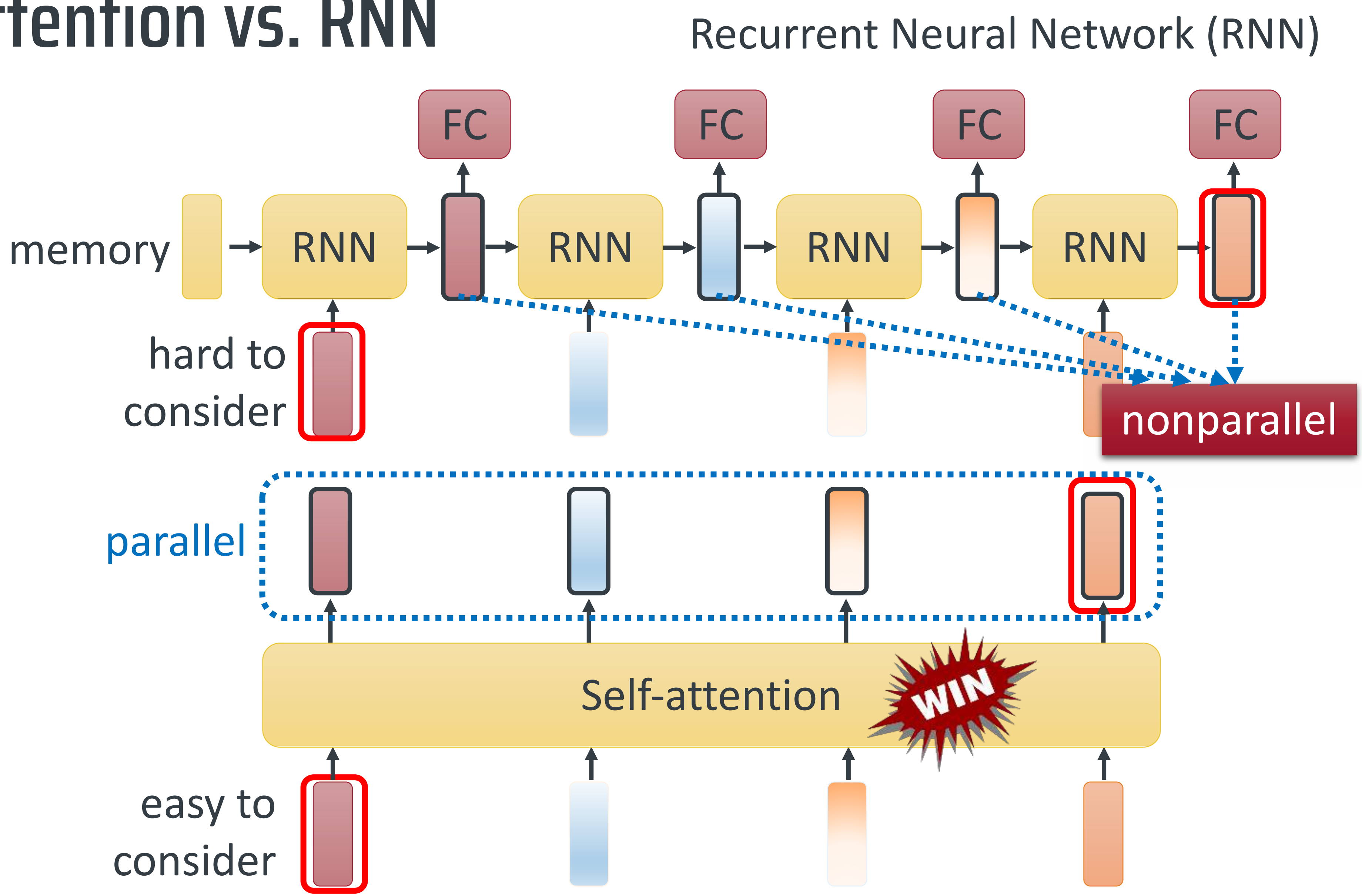
https://arxiv.org/abs/1706.03762

BERT

https://arxiv.org/abs/1810.04805

Widely used in Natural Langue Processing (NLP)!

Self-attention vs. RNN



Transformer

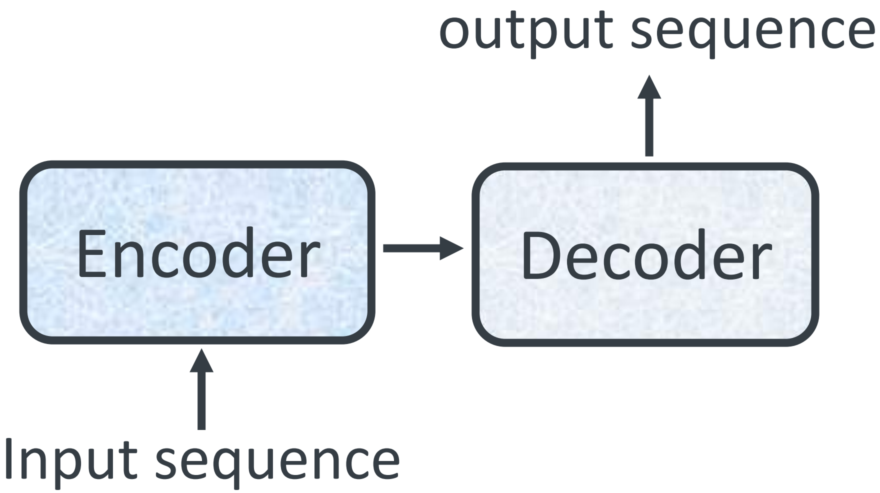
Sequence-to-sequence (Seq2seq)

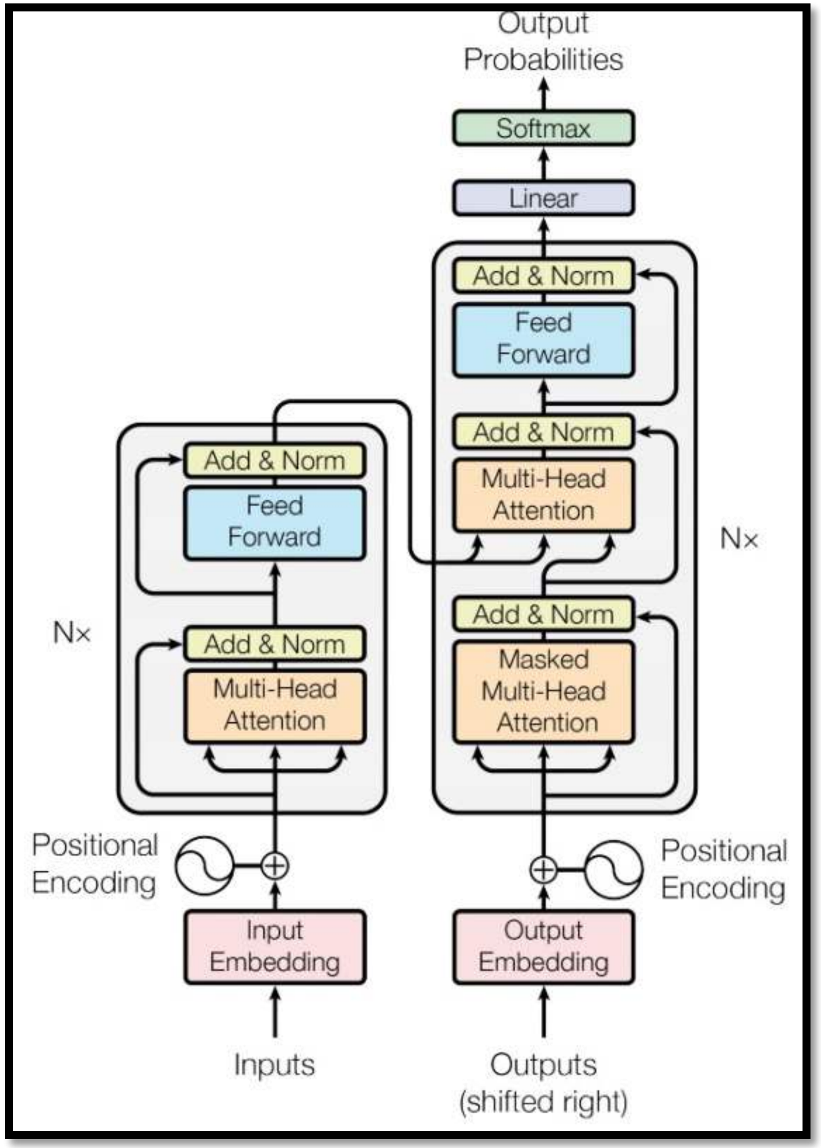
- Seq2seq: Input a sequence, output a sequence
  - The output length is determined by model, and can vary from the input
- Examples of seq2seq
  - Speech recognition: The input is audio signal, and the output is the transcript of the audio
  - Machine translation: The input and output are text in different languages
  - Large language models (Chatbot): Question answering
  - Sentiment analysis: The input is text and the output is a number indicating sentiment

question, context → Seq2seq → answer

- **Transformer** is a sequence-to-sequence model built on self attention

Seq2seq





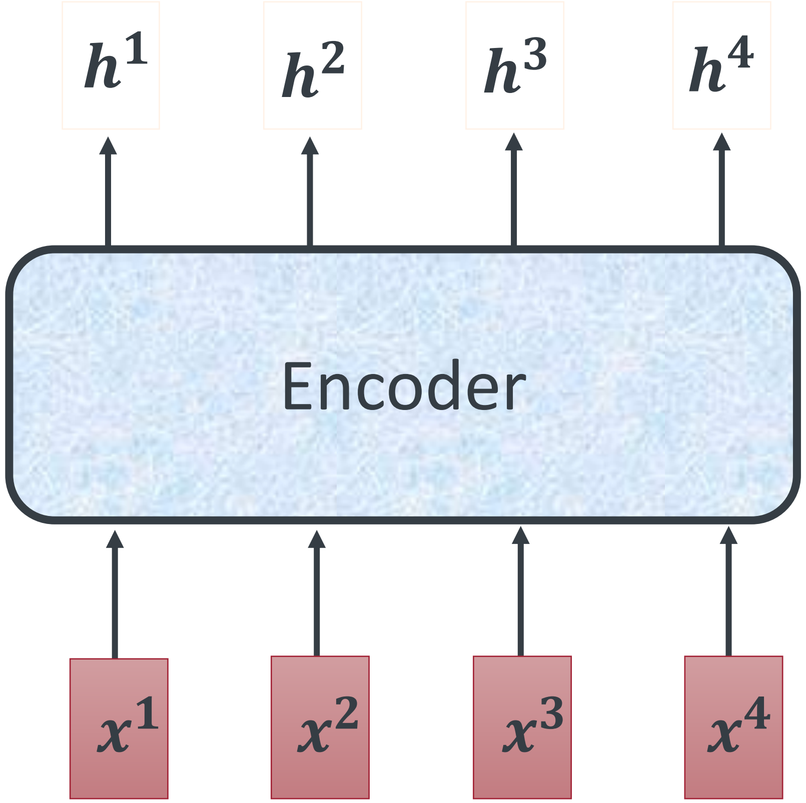
Transformer

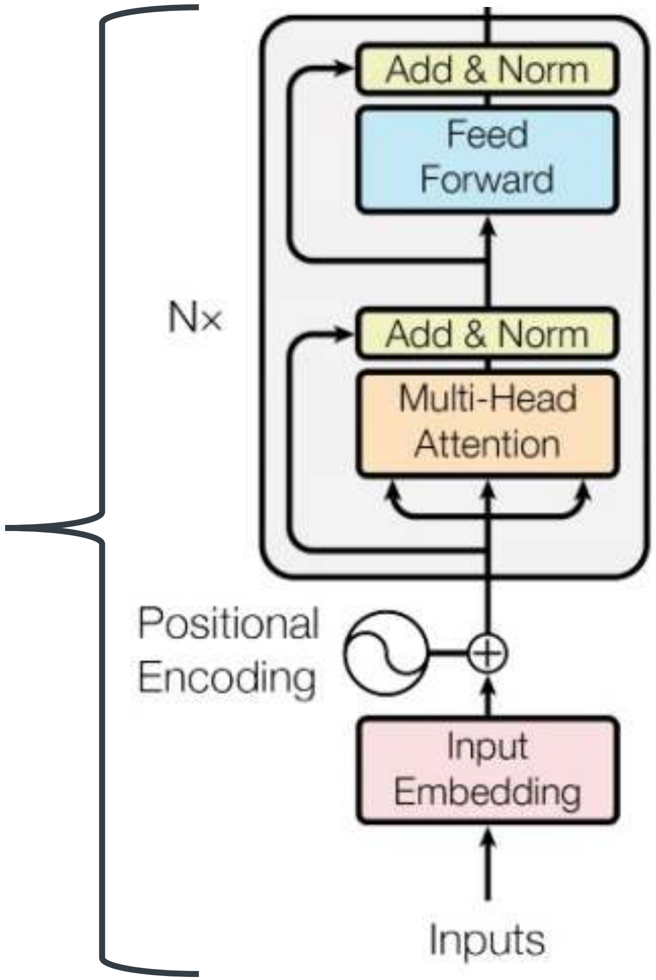
https://arxiv.org/abs/1706.03762

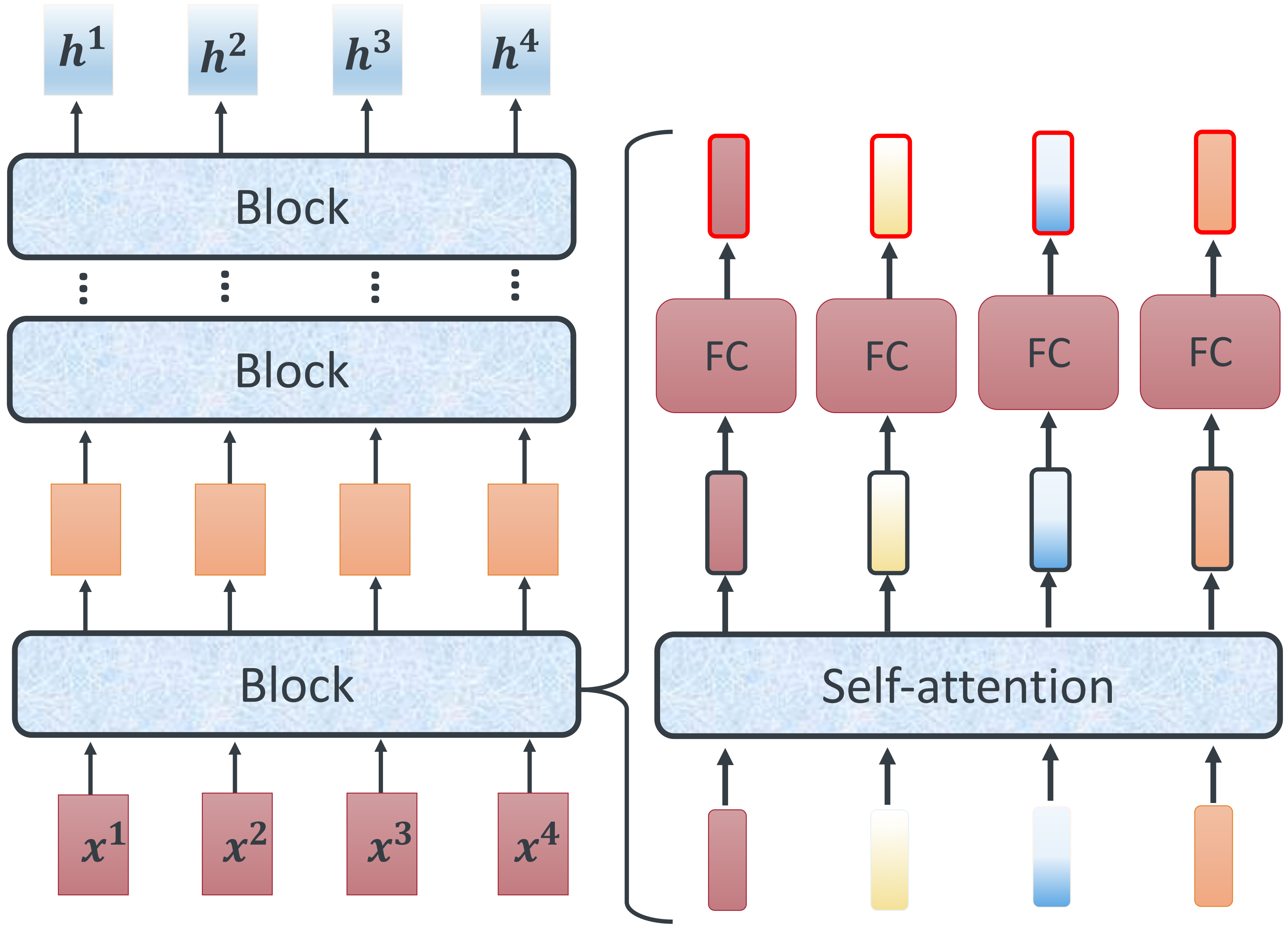
Encoder

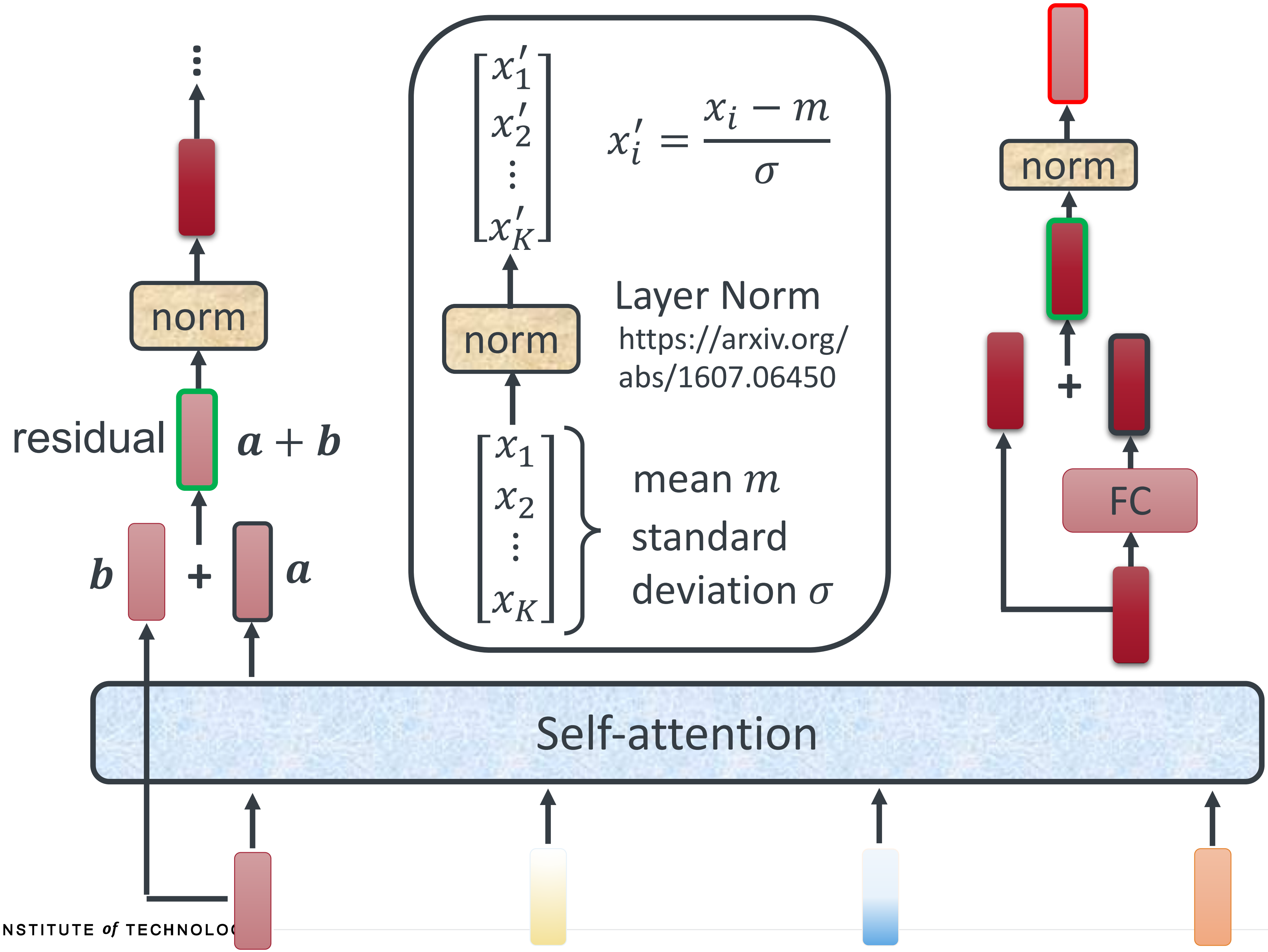
Transformer's Encoder

You can use any **RNN** as the encode





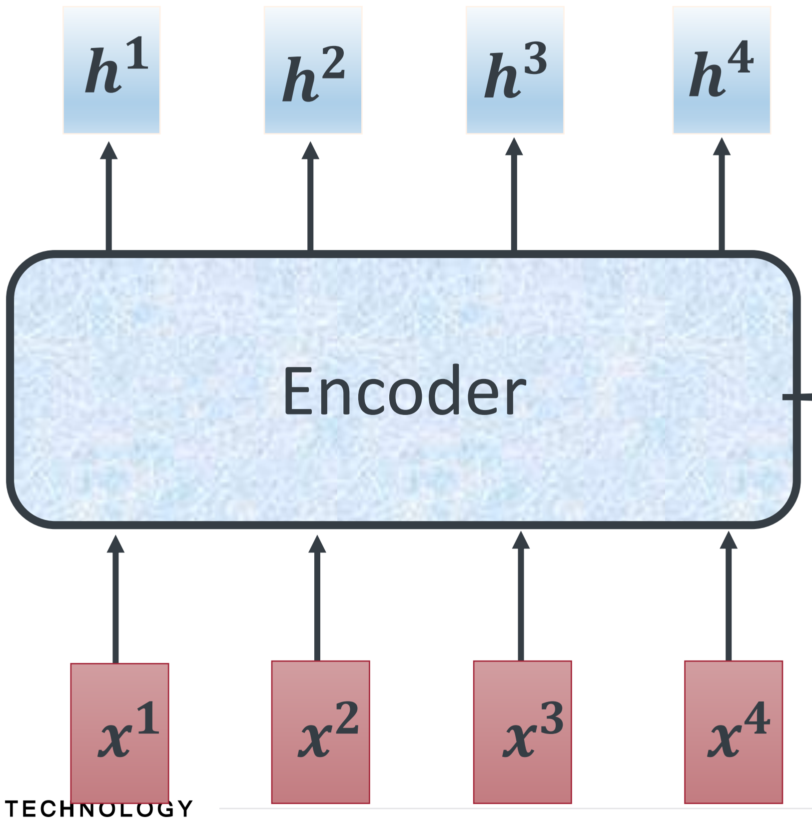

h^1
h^2
h^3
h^4
Block
Block
Block
FC
FC
FC
FC
Self-attention
x^1
x^2
x^3
x^4

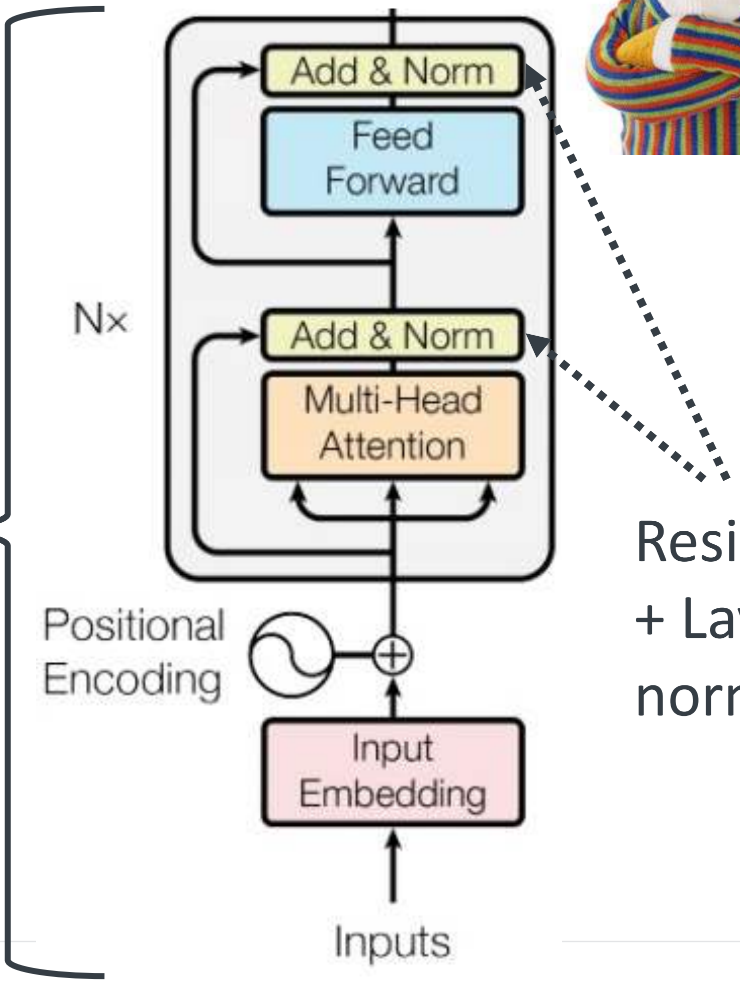


$\begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_K \end{bmatrix}$
$x'_i = \frac{x_i - m}{\sigma}$
norm
Layer Norm
https://arxiv.org/abs/1607.06450
$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix}$
mean m
standard deviation σ
norm
residual
$a + b$
b
+
a
norm
+
FC
Self-attention

BERT

I use the **same** network architecture as **transformer encoder**.

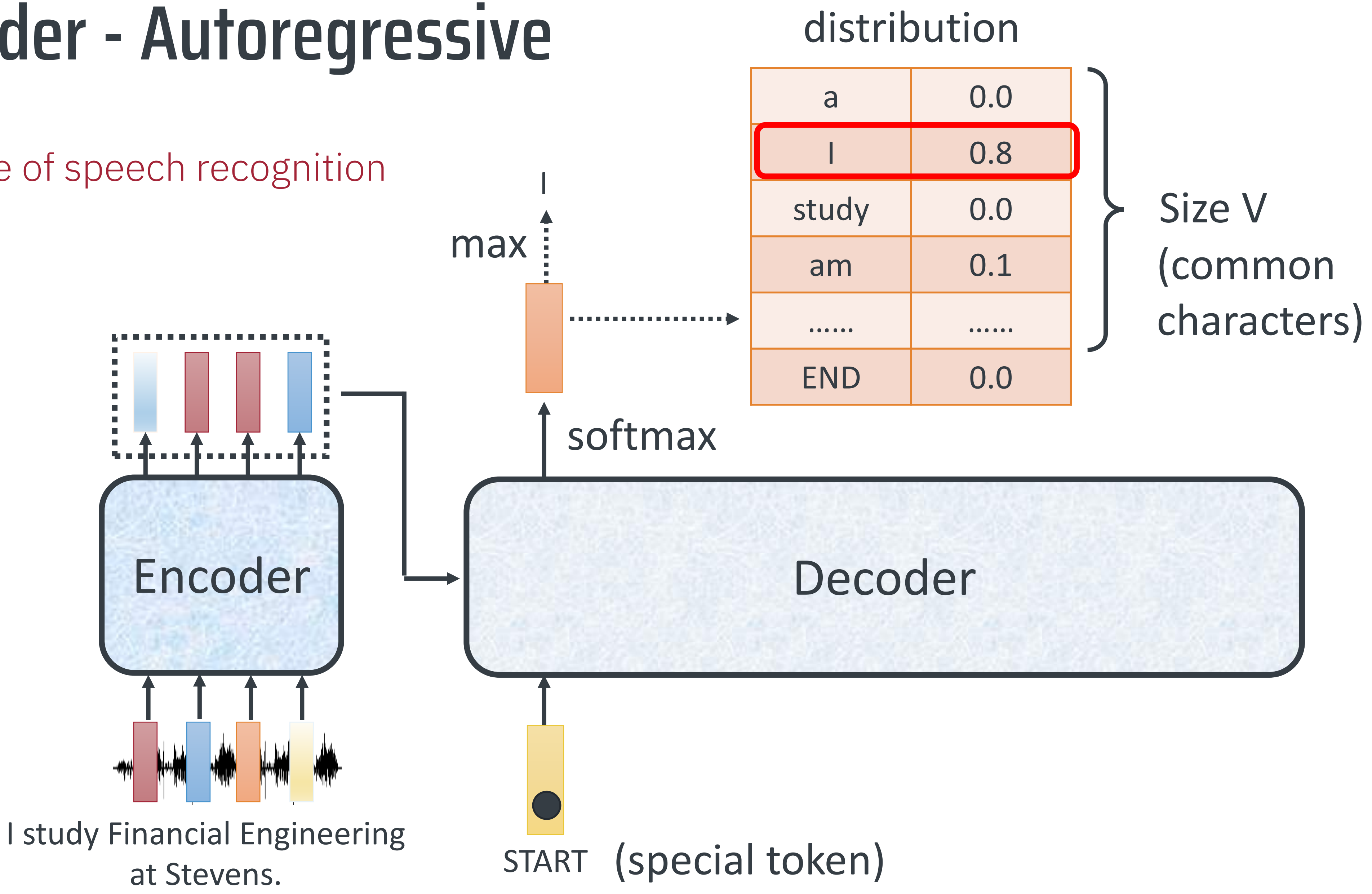




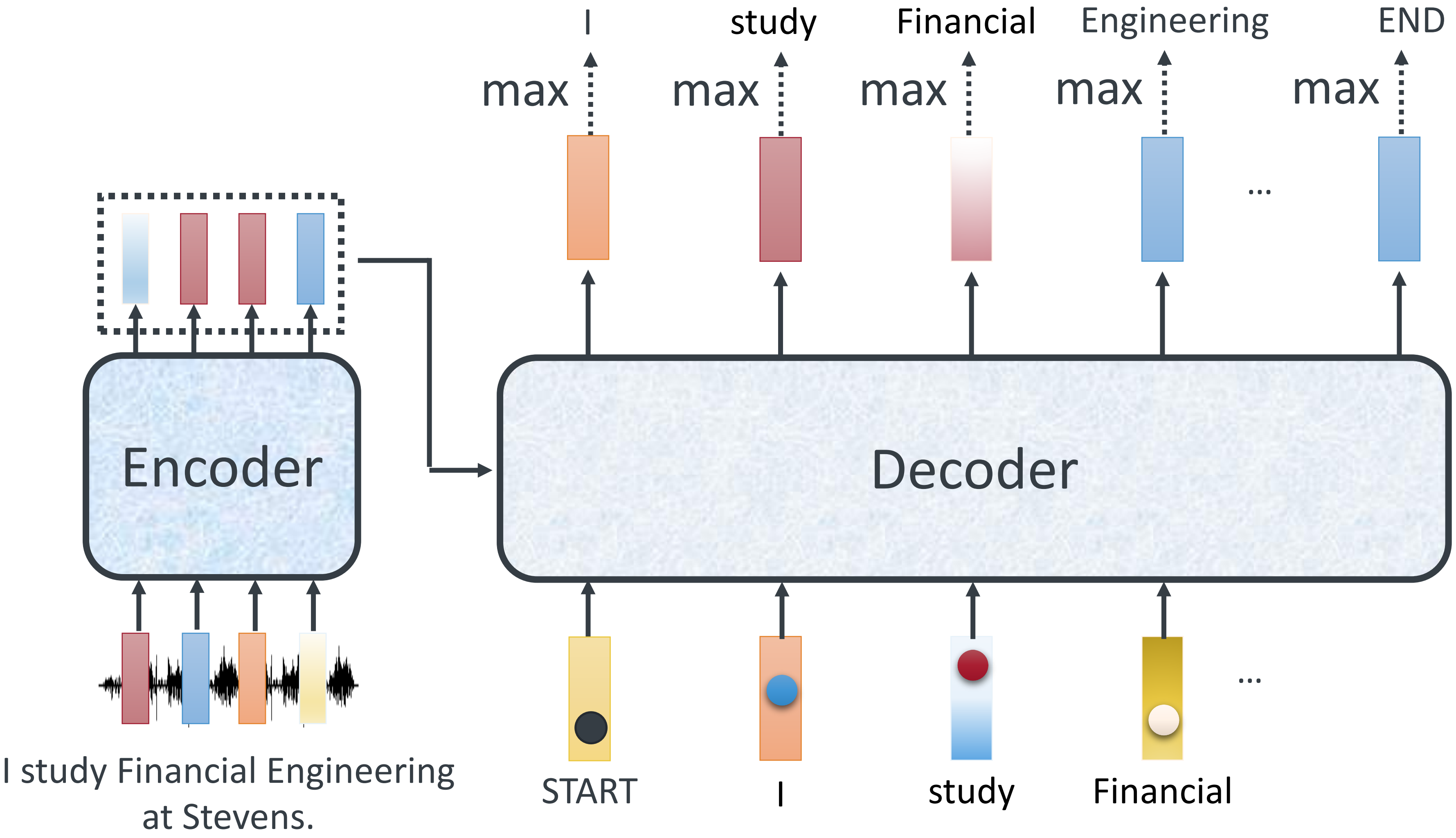
Residual + Layer norm

Decoder - Autoregressive

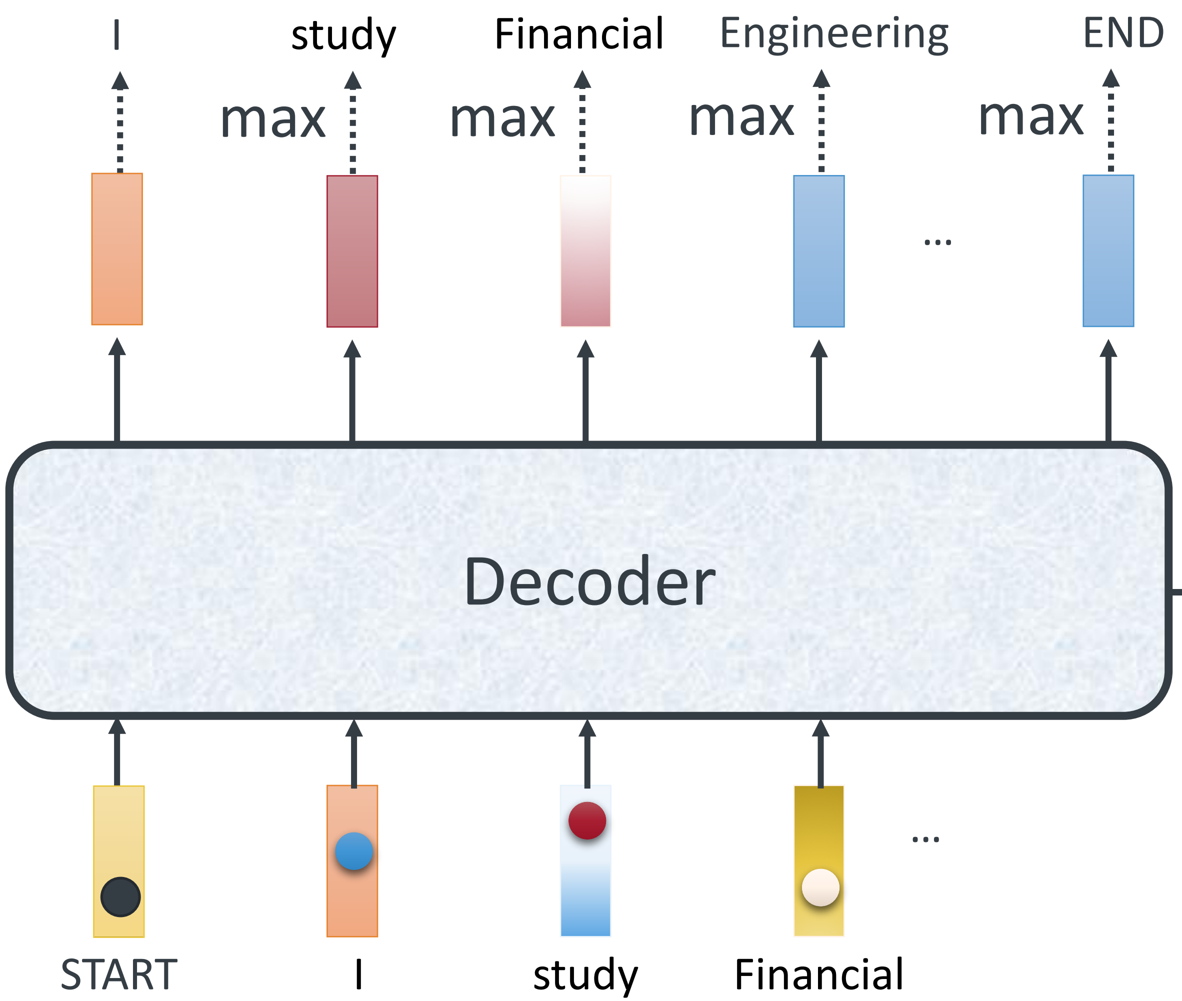
Example of speech recognition

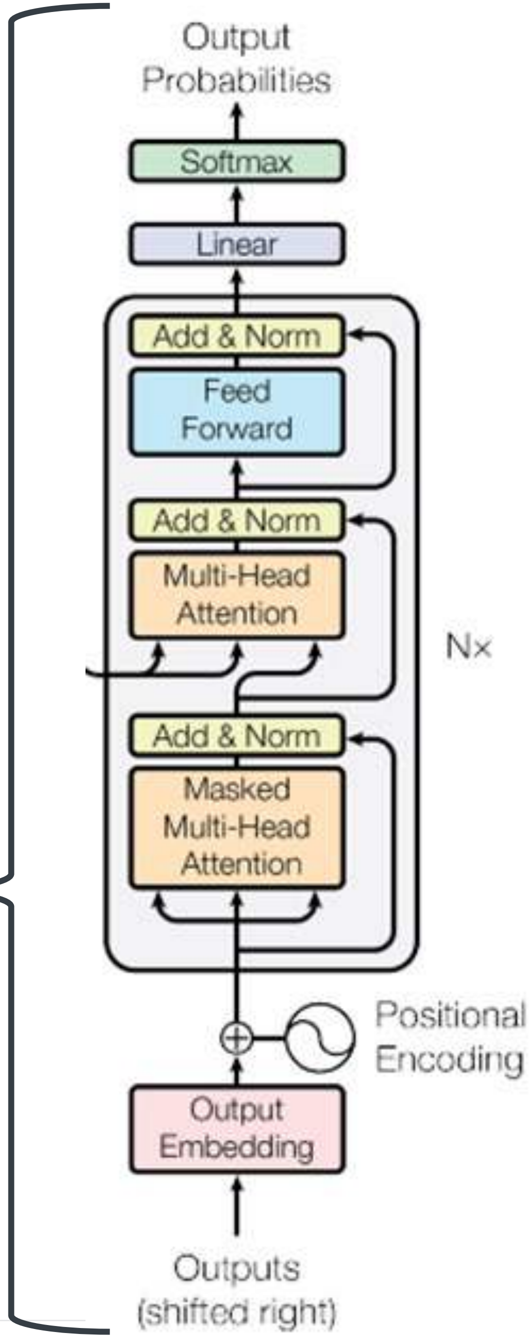


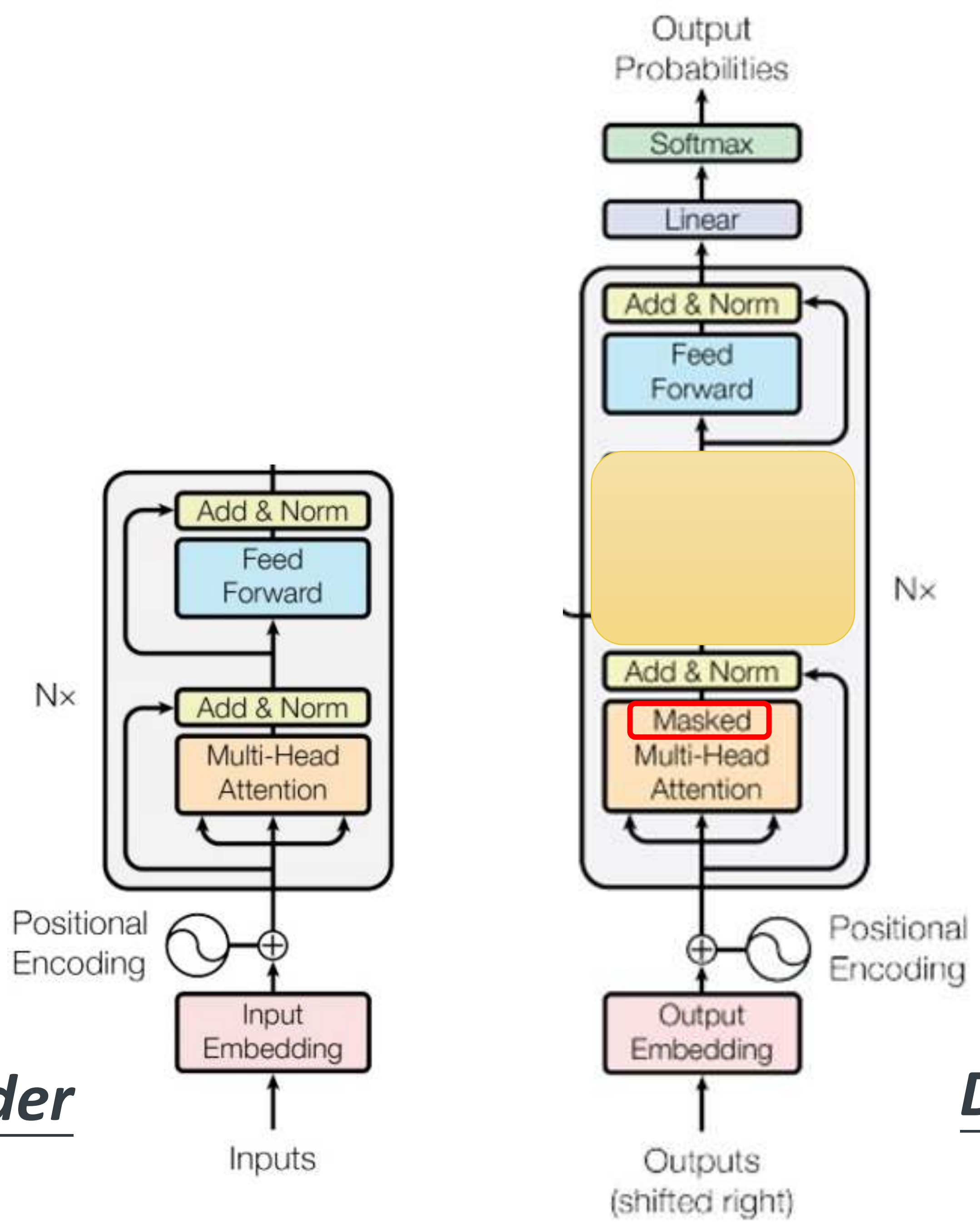
Decoder - Autoregressive



ignore the input from the encoder here ☺



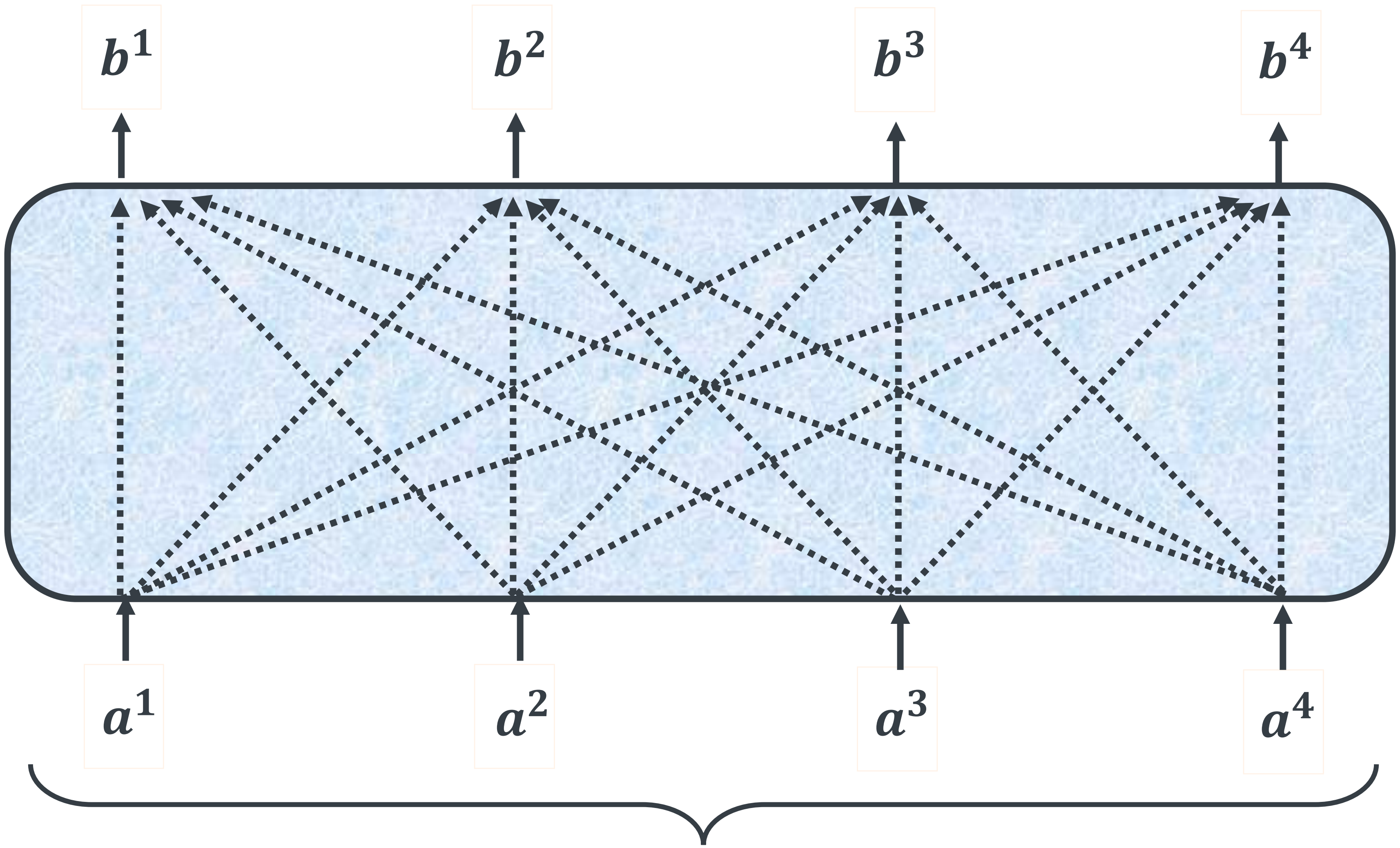




Encoder

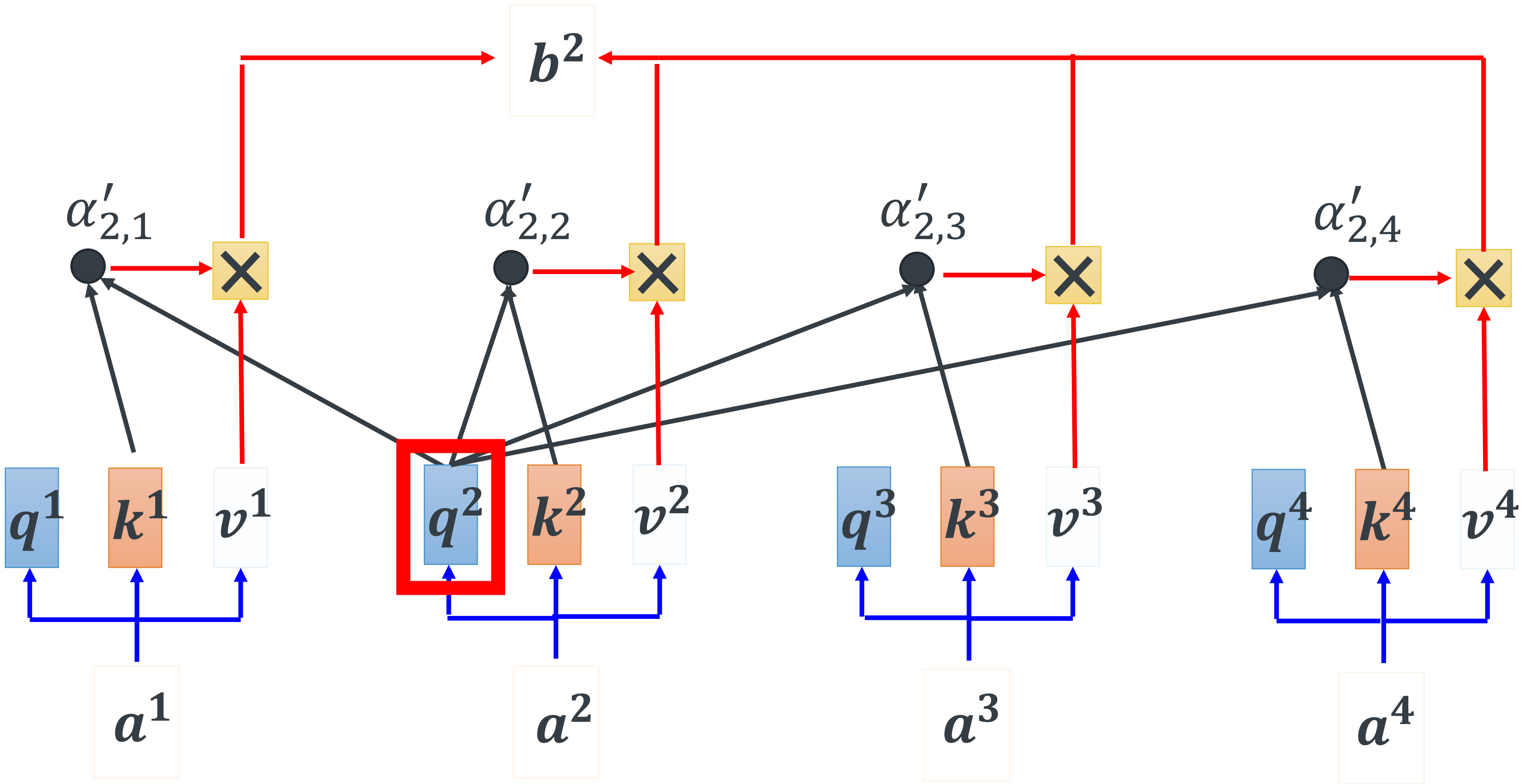
Decoder

Self-attention → Masked Self-attention



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

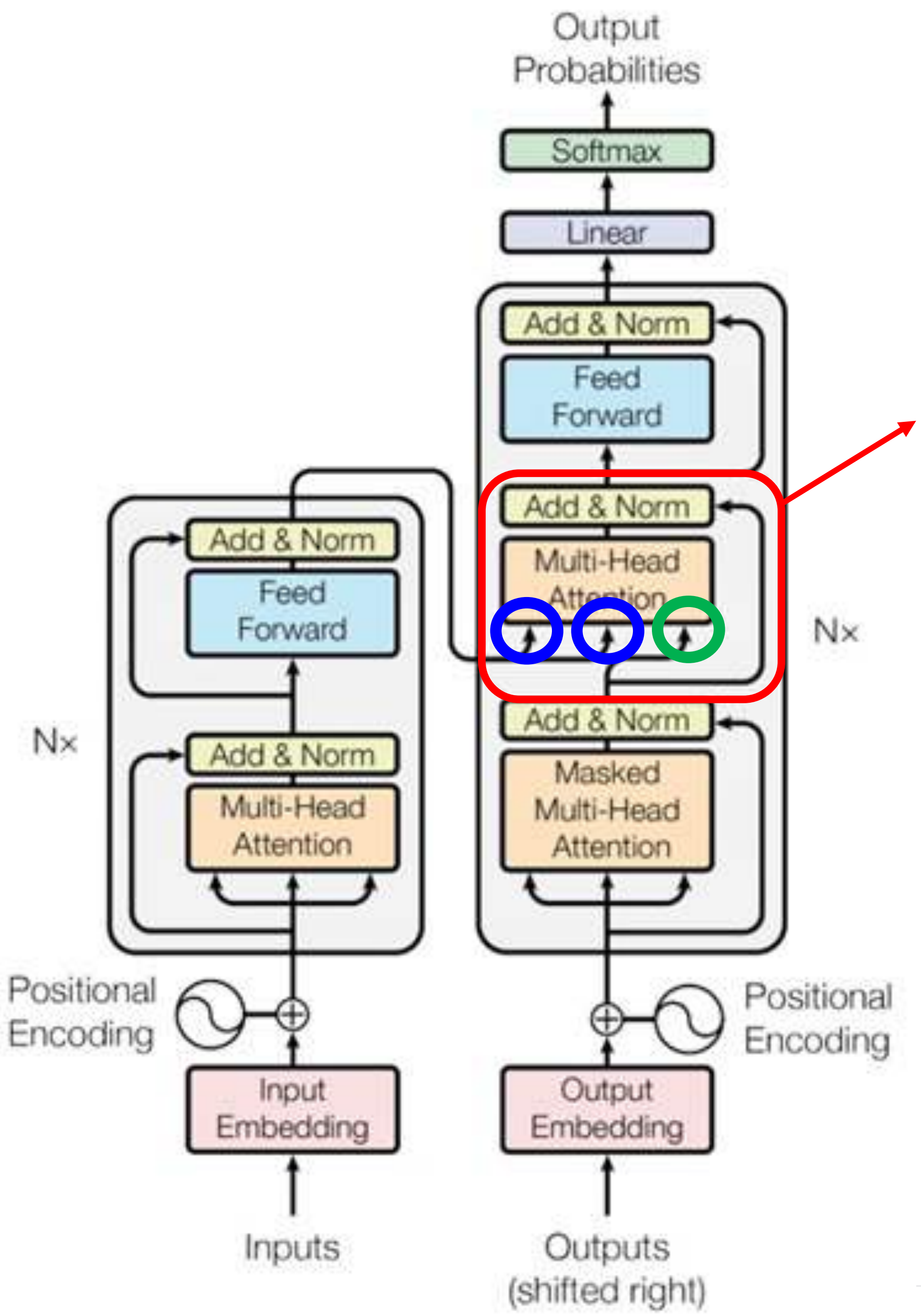
Self-attention → Masked Self-attention



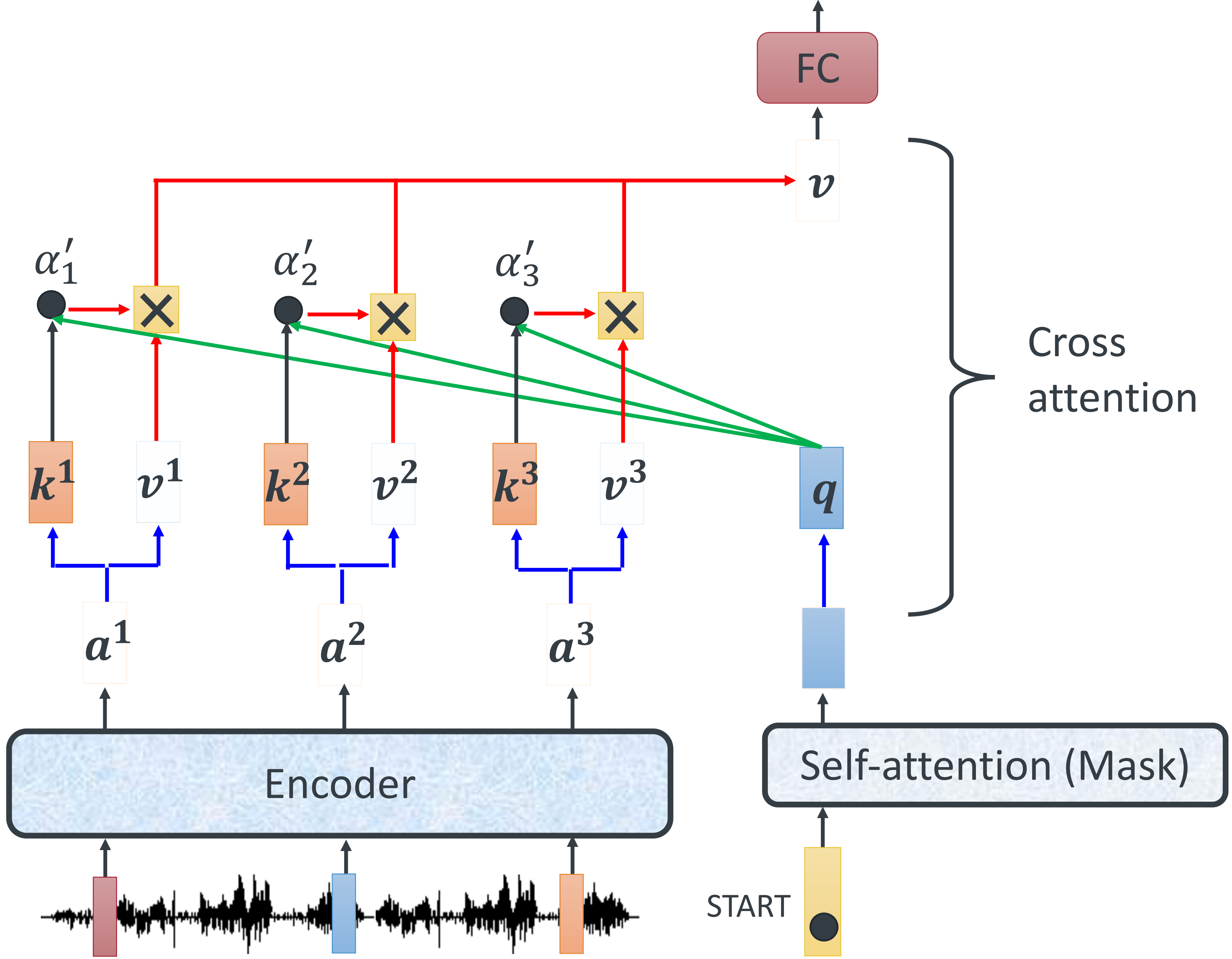
Why masked? Consider how does decoder work

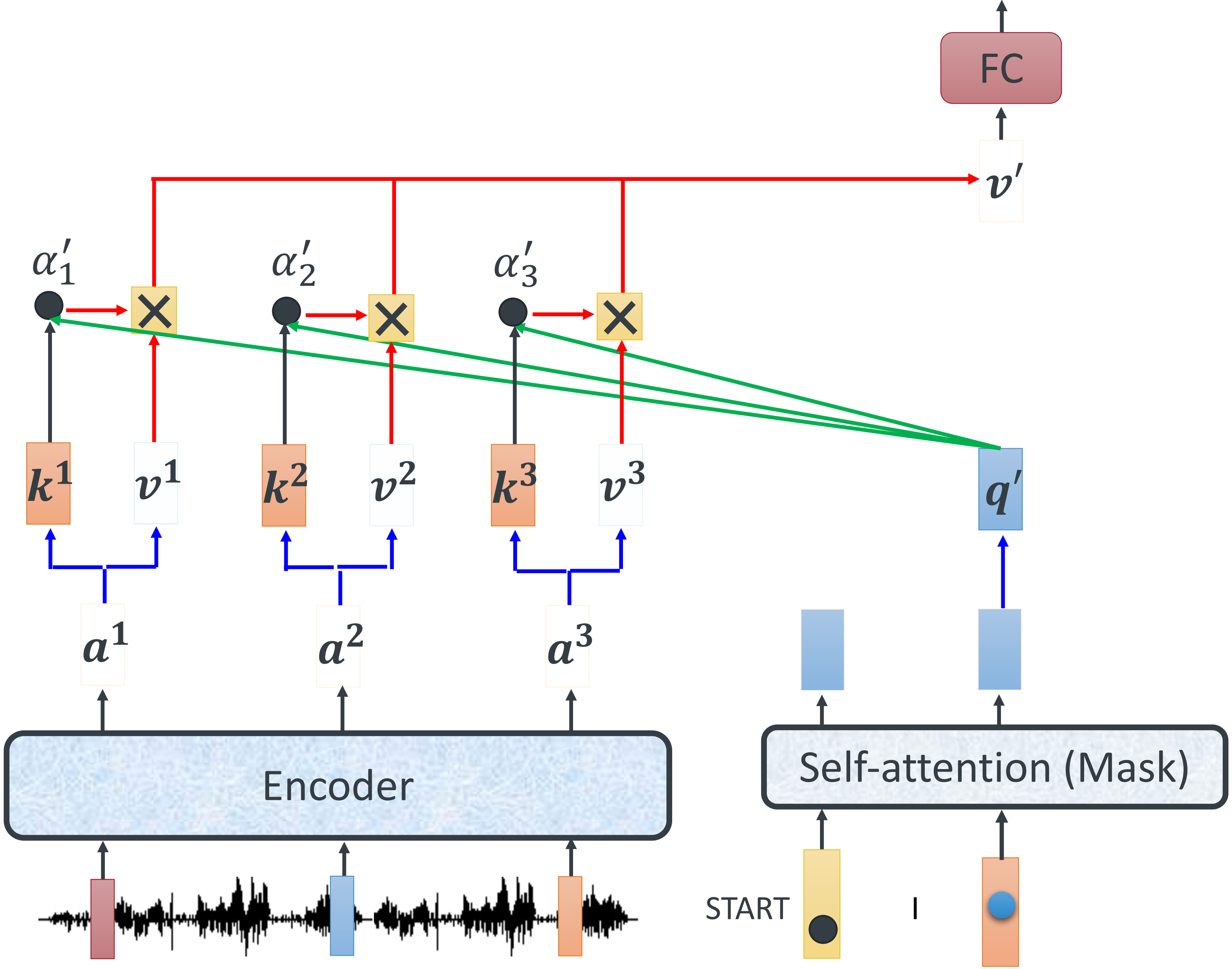
Transformer - Cross Attention

Information transformation from encoder to decoder

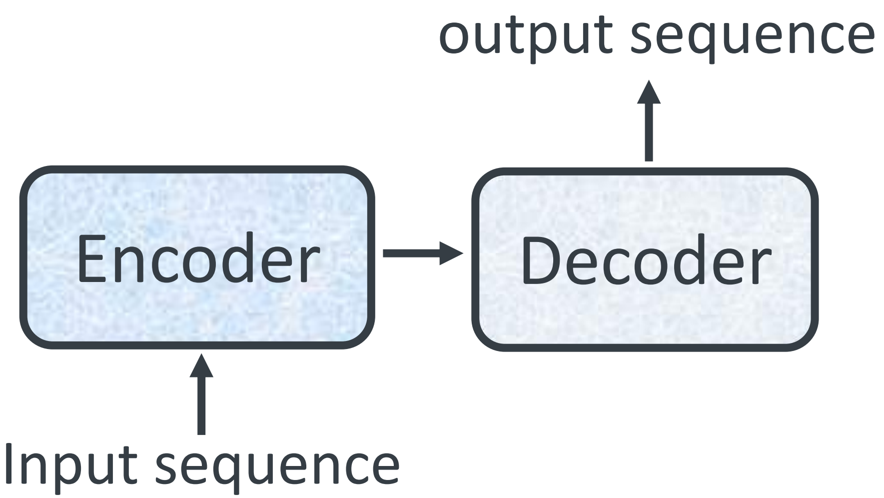


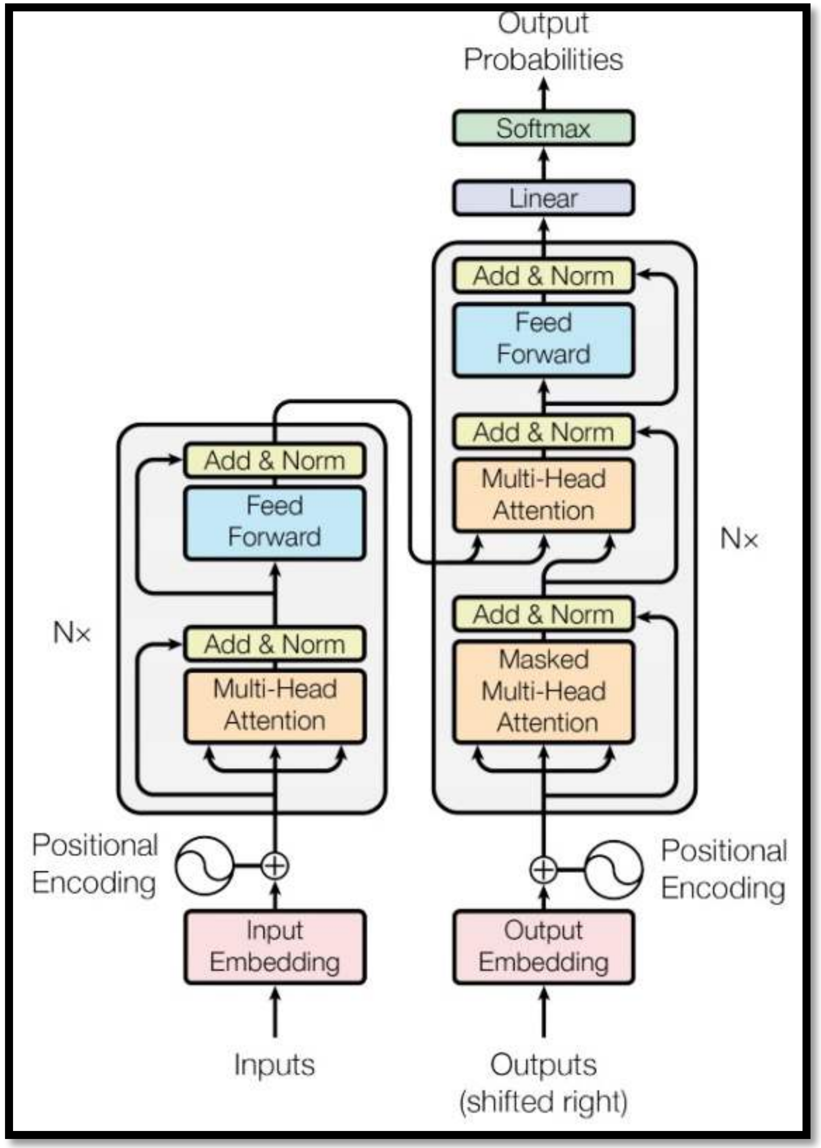
Cross attention


FC
v
α'_1
α'_2
α'_3
Cross attention
k^1
v^1
k^2
v^2
k^3
v^3
q
a^1
a^2
a^3
Encoder
Self-attention (Mask)
START


FC
v'
α'_1
α'_2
α'_3
k^1
v^1
k^2
v^2
k^3
v^3
q'
a^1
a^2
a^3
Encoder
Self-attention (Mask)
START
I

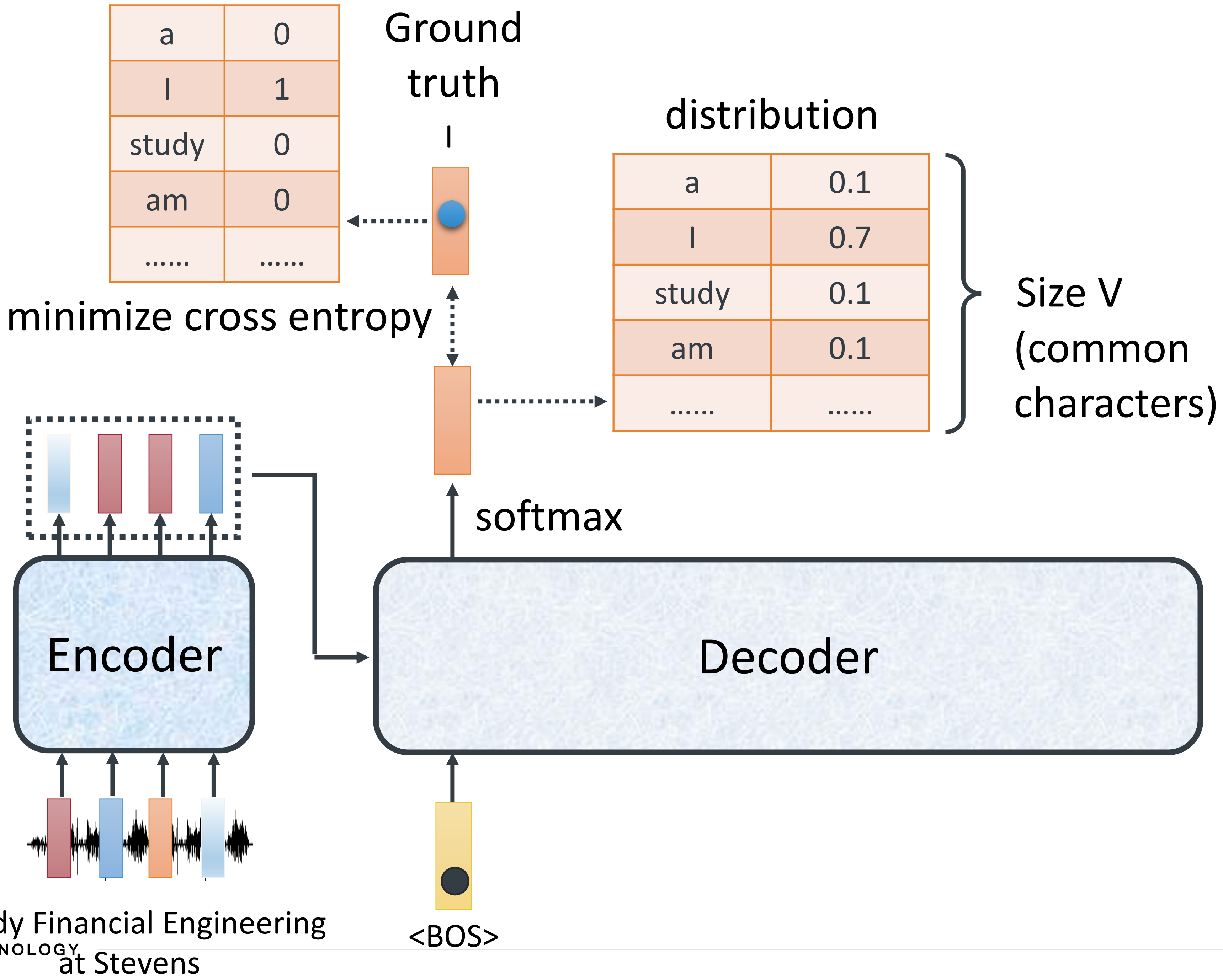
Transformer





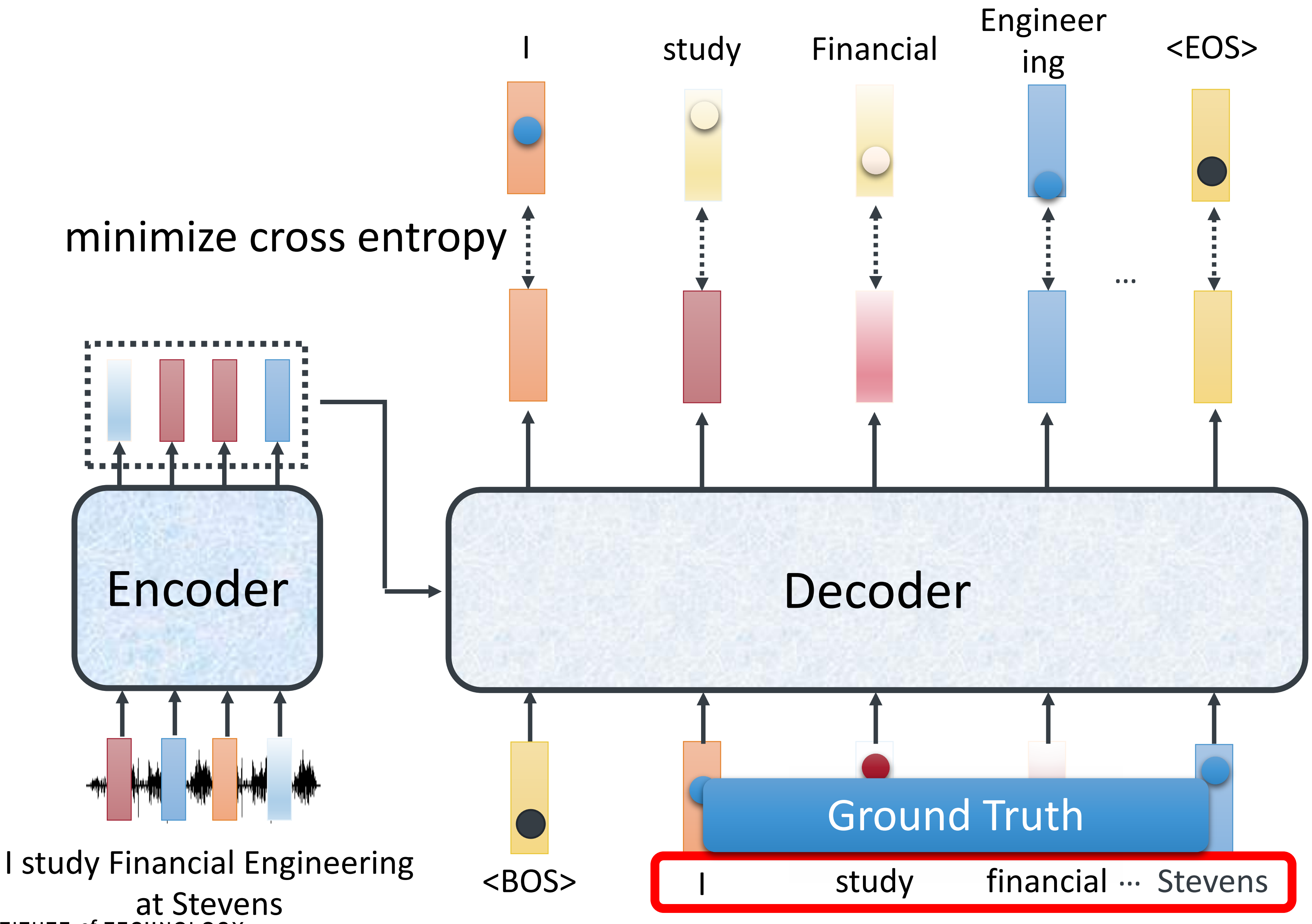
Transformer
https://arxiv.org/abs/1706.03762

Training



Ground truth
I
a 0
I 1
study 0
am 0
…… ……
minimize cross entropy
distribution
a 0.1
I 0.7
study 0.1
am 0.1
…… ……
Size V (common characters)
softmax
Encoder
Decoder
I study Financial Engineering at Stevens
<BOS>

Teacher Forcing: using the ground truth as input.



Bidirectional Encoder Representations from Transformers (BERT)

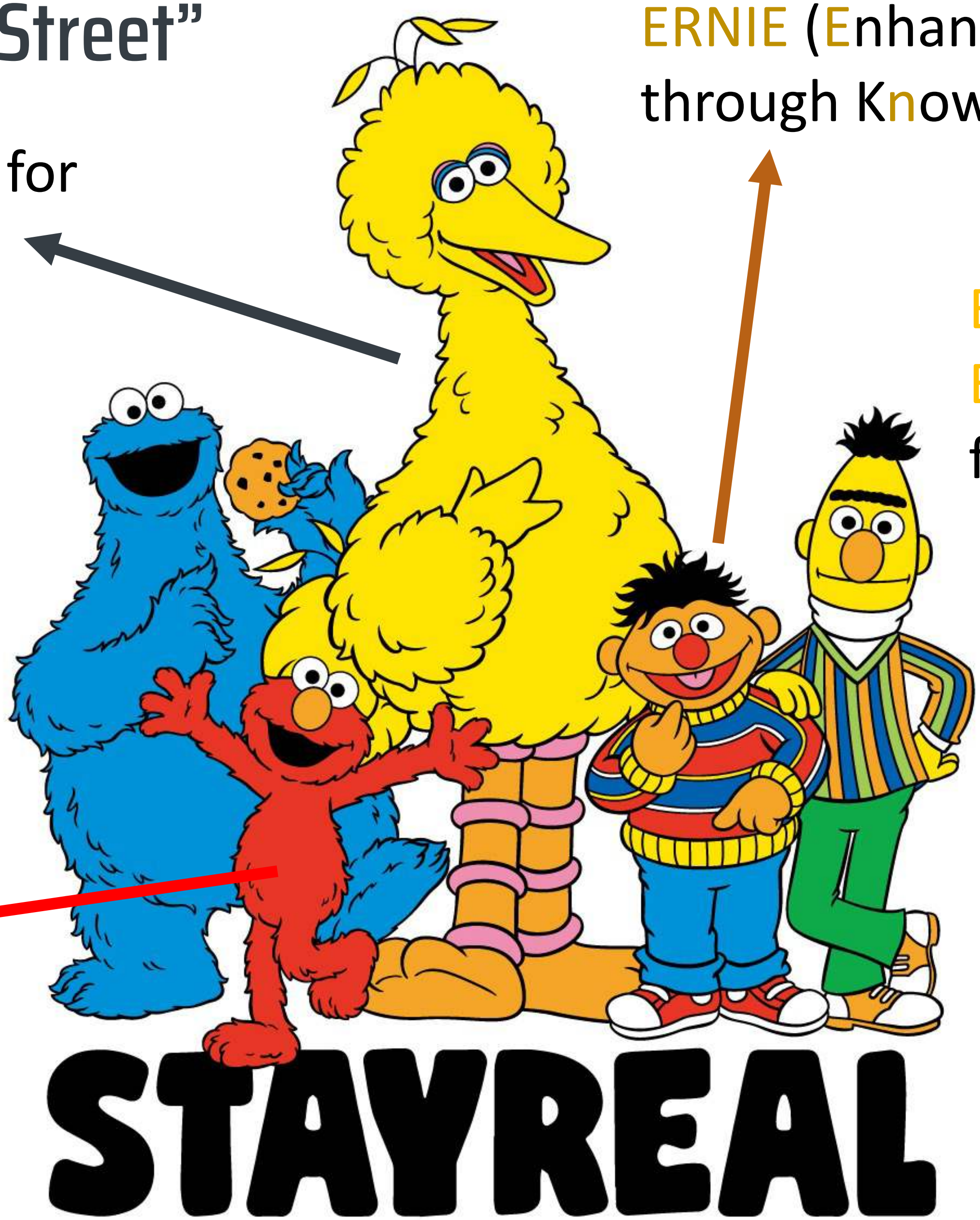
TV Show “Sesame Street”

Big Bird: Transformers for Longer Sequences

ERNIE (Enhanced Representation through Knowledge Integration)

BERT (Bidirectional Encoder Representations from Transformers)

ELMo (Embeddings from Language Models)



The models have become larger and larger …

ELMO
(94M)

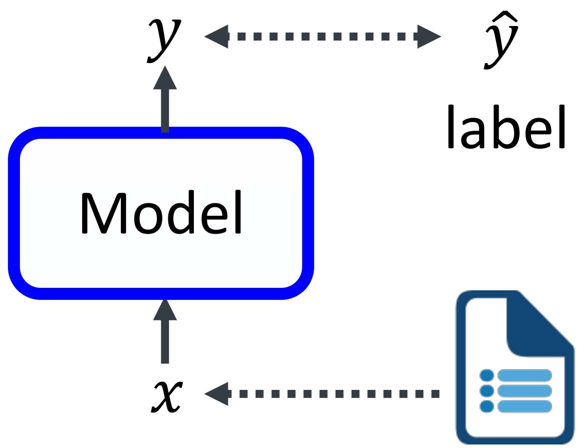
BERT
(340M)

GPT-2
(1542M)

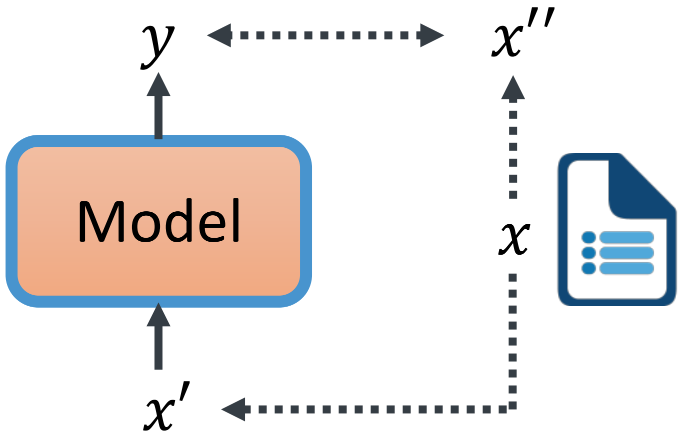
GPT-3
(175B)

Self-supervised Learning

Supervised



Self-supervised



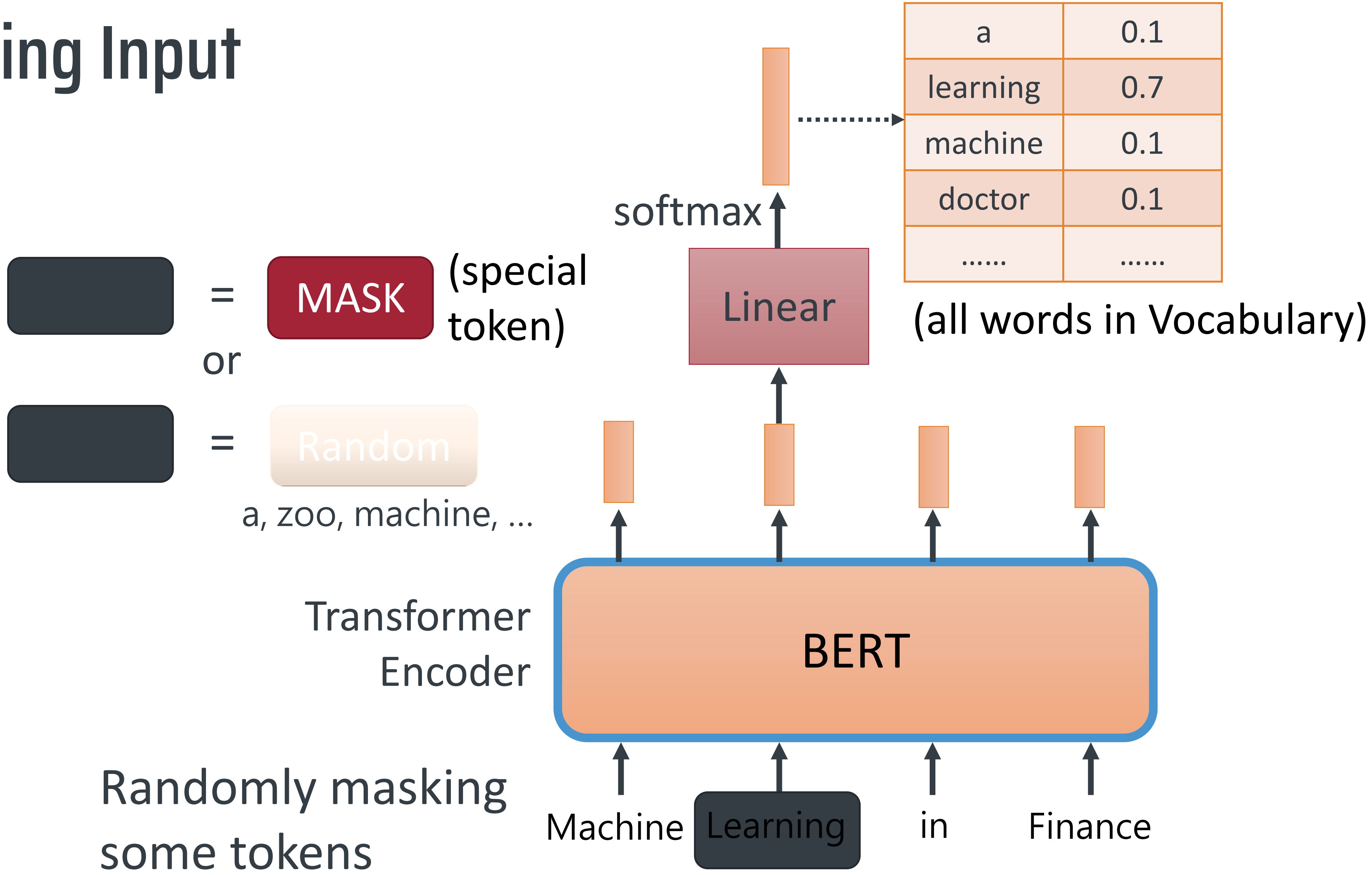
Yann LeCun

2019年4月30日 ·

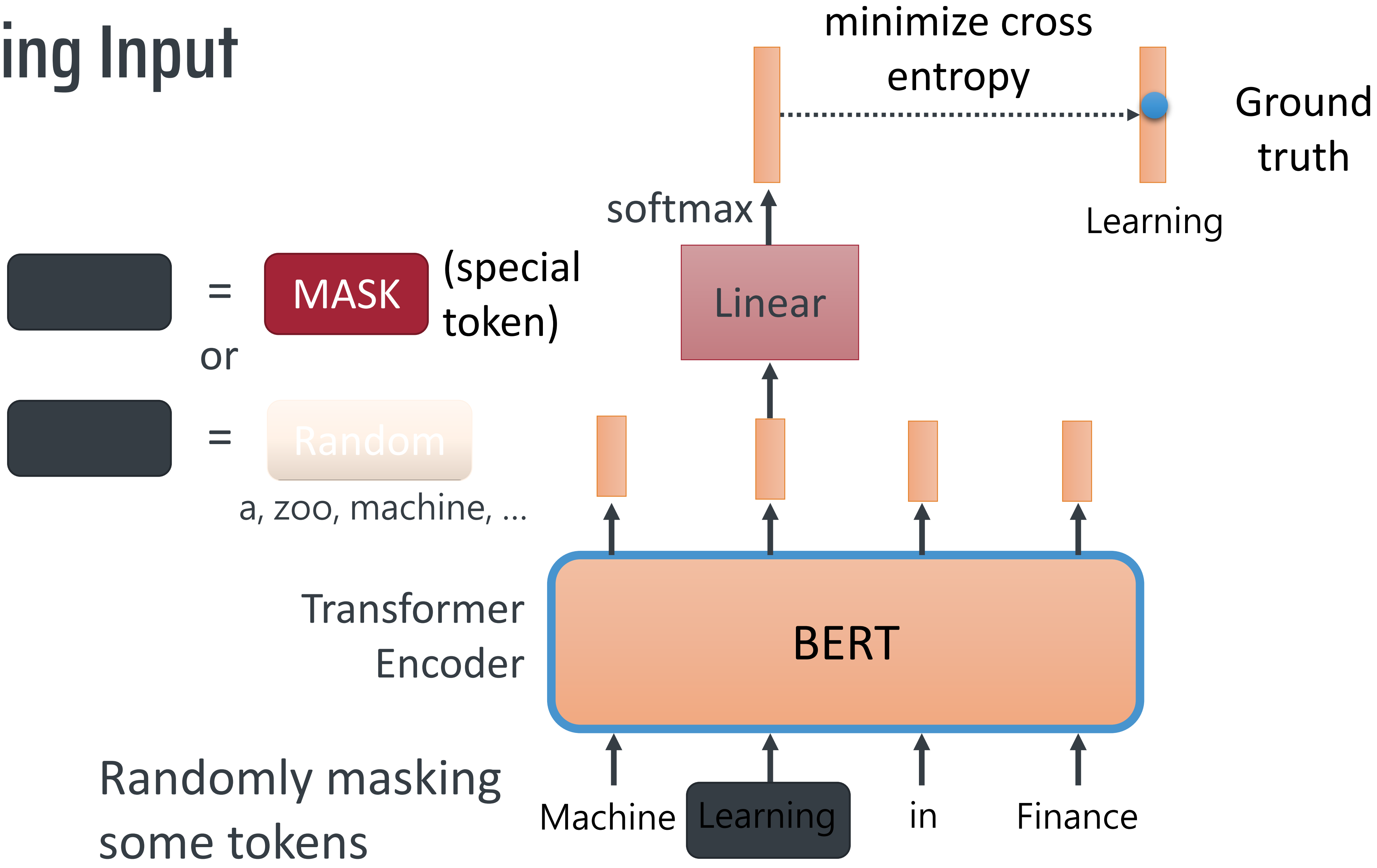
I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

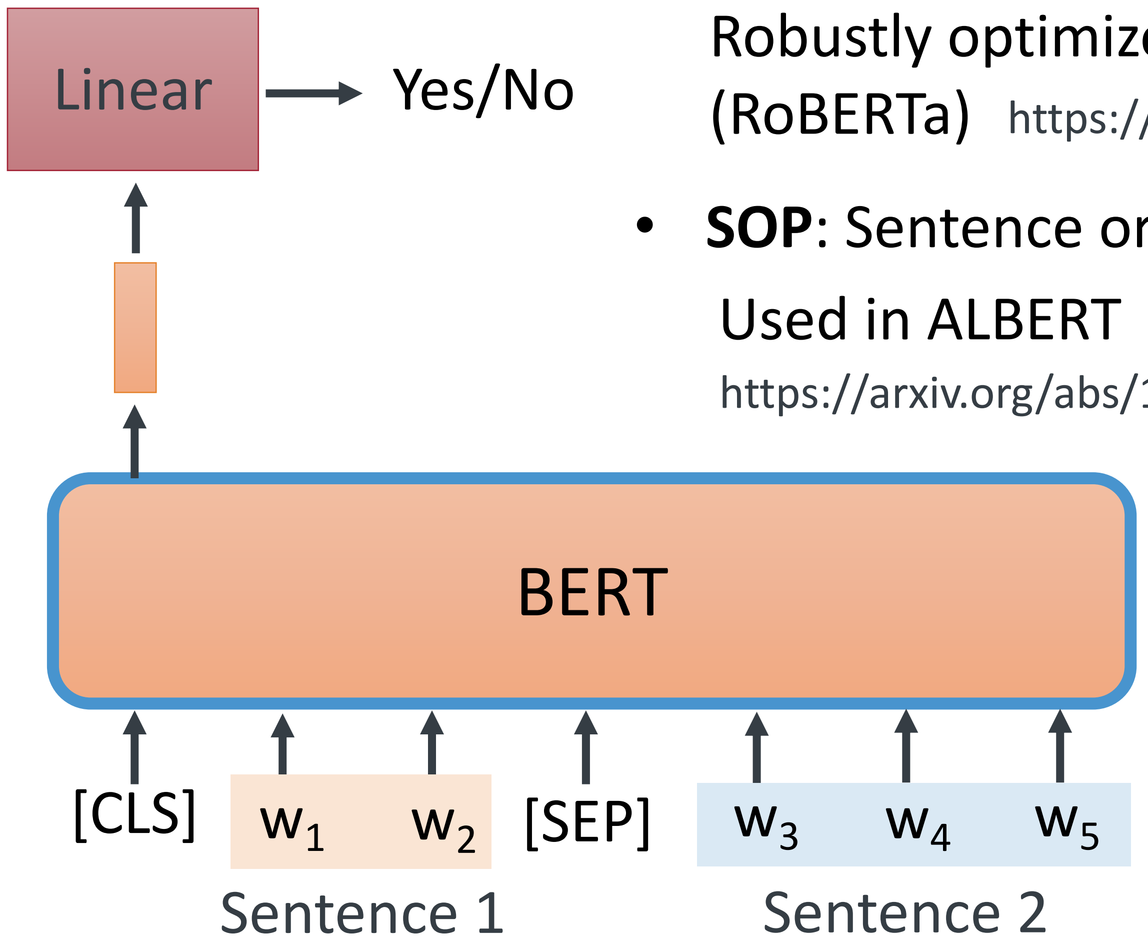
Masking Input



Masking Input



Next Sentence Prediction



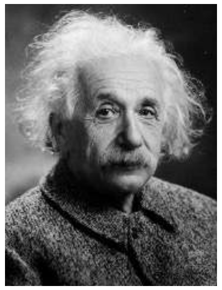
- This approach is not helpful.

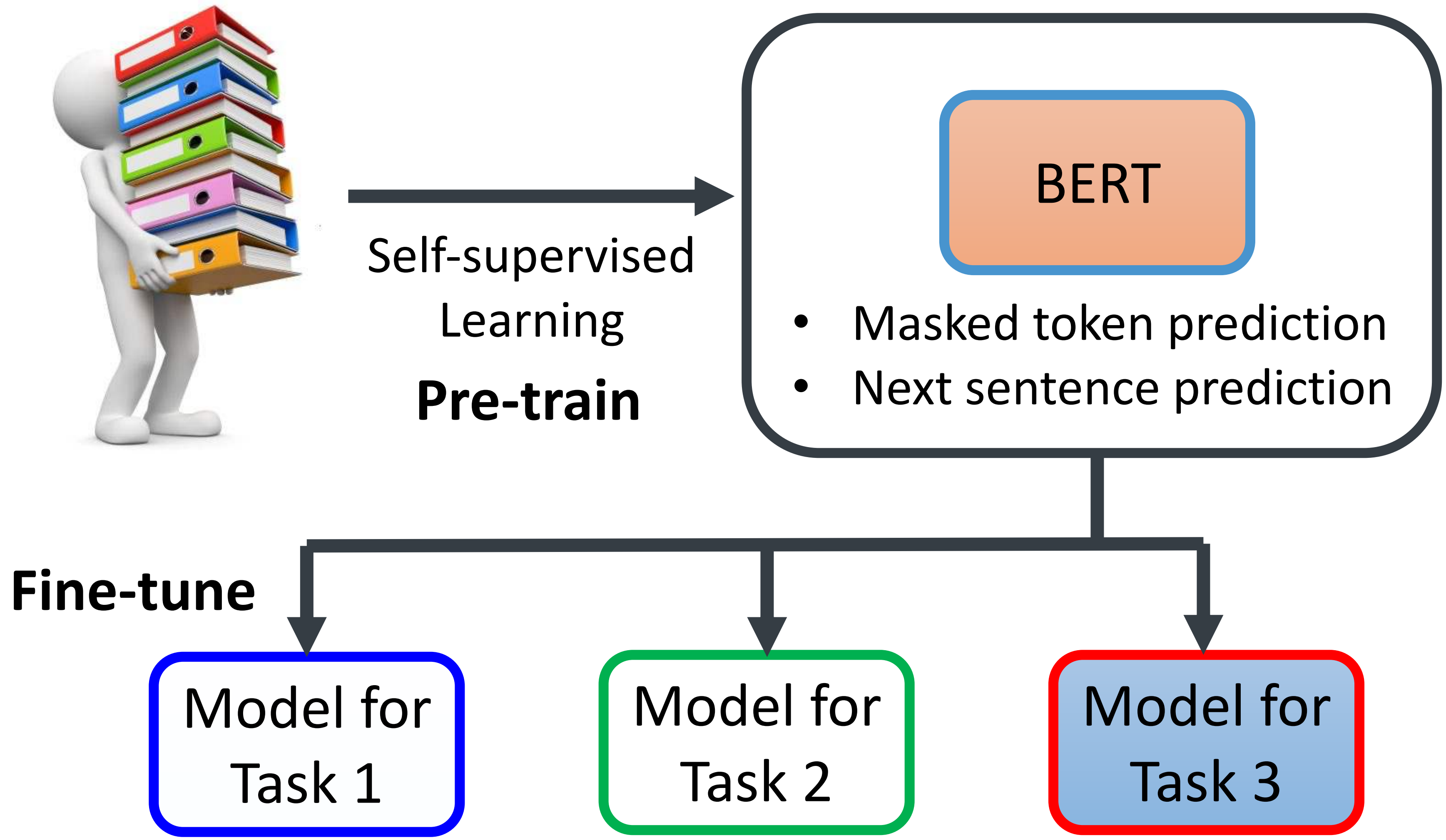
  Robustly optimized BERT approach (RoBERTa) https://arxiv.org/abs/1907.11692

- **SOP**: Sentence order prediction

  Used in ALBERT

  https://arxiv.org/abs/1909.11942

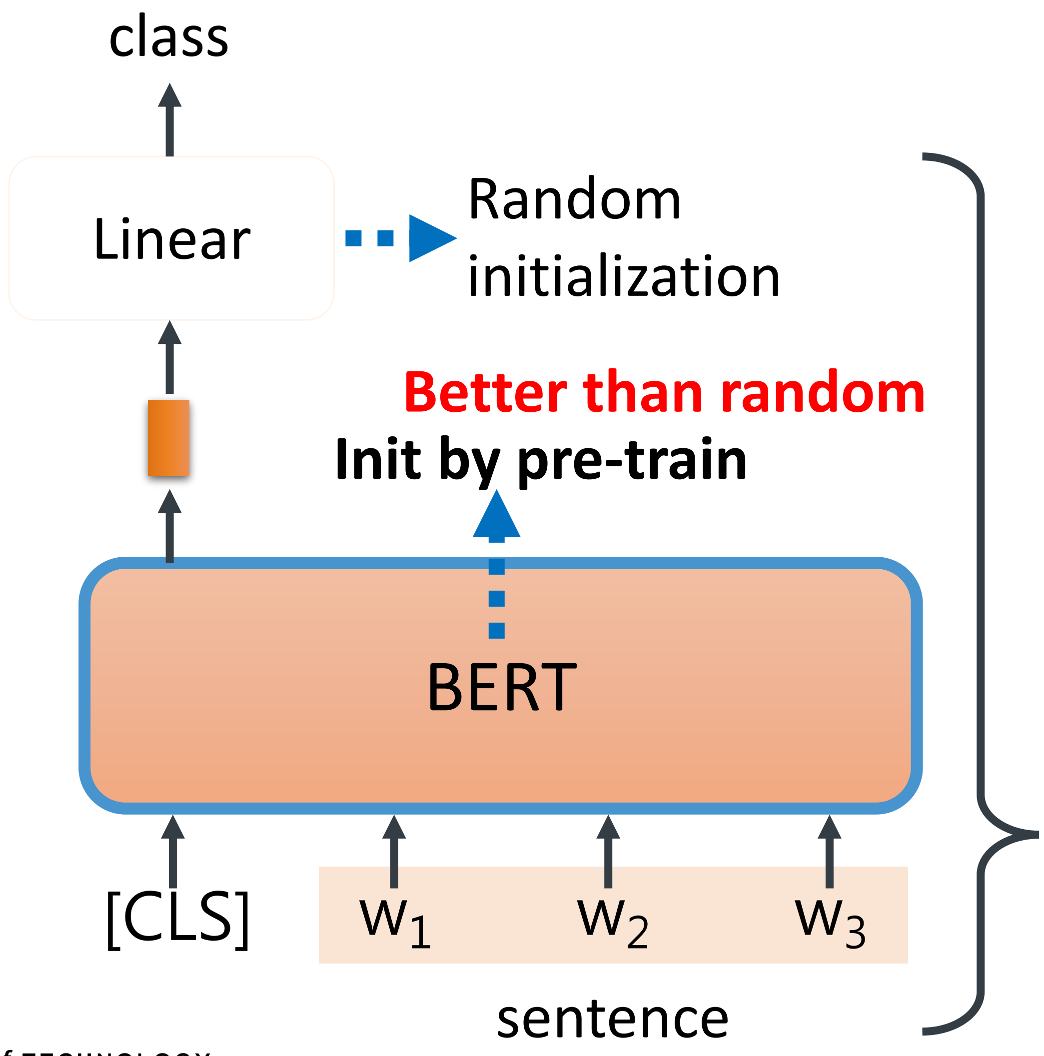




Downstream Tasks

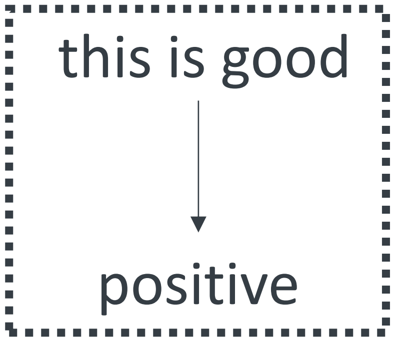
- For example, sentiment analysis

Sentiment Analysis using BERT



Input: sequence
output: class

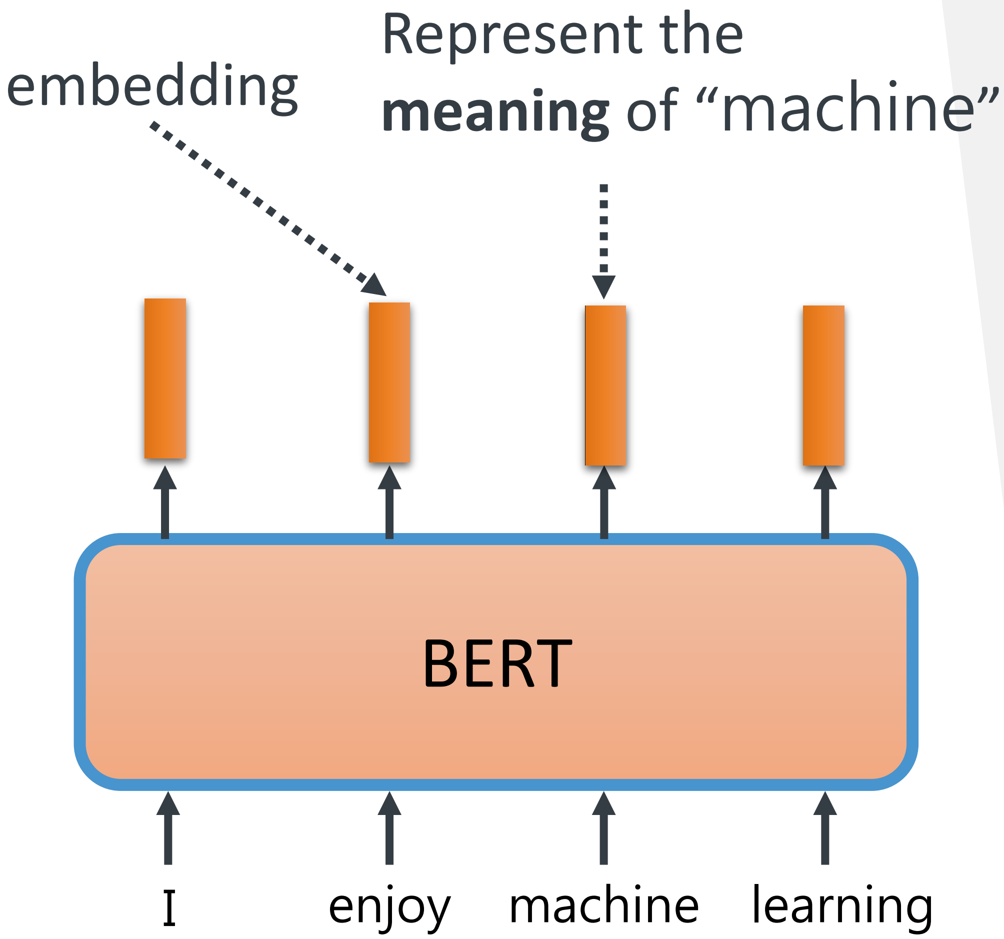
Example:
Sentiment analysis



This is the model to be learned.

Why does BERT Work?

- BERT can be used as embedding
- Compared with embeddings models like Word2Vec, BERT considers context
  - “Apple launches the new iPad.”
  - “My favorite fruit is Apple.”
- Recall how Word2Vec is trained
  - Continuous Bag of Words is similar to the self–supervised learning in BERT



I eat an apple every morning for breakfast.

The apple tree in the garden is full of red fruits.

An apple a day keeps the doctor away.

She sliced the apple into small pieces for the salad.

This apple tastes sweet and juicy.

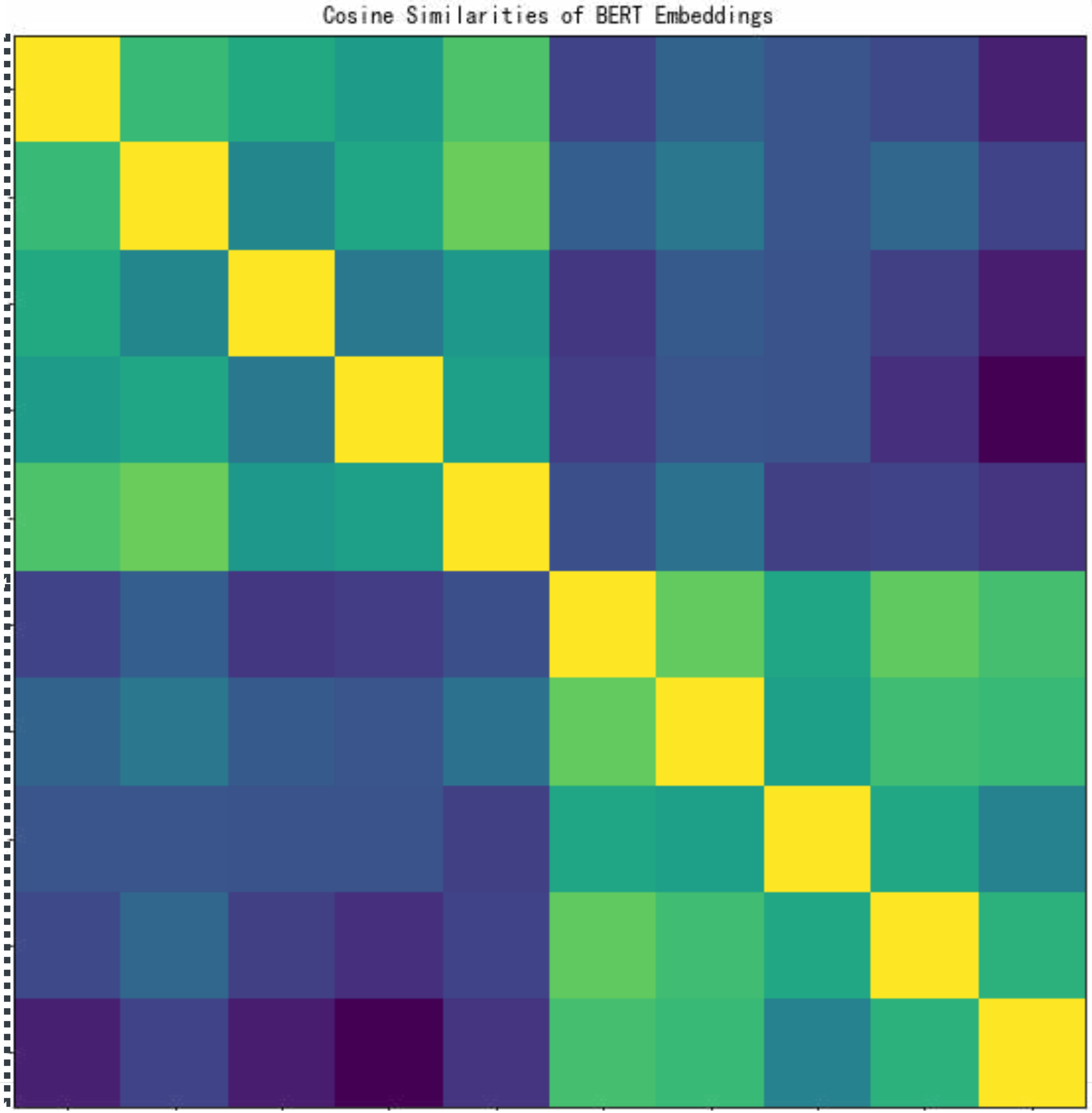
Apple is known for its innovation.

I bought my first Apple laptop last year.

Apple released the latest version of iOS yesterday.

Many people love Apple's user-friendly interface.

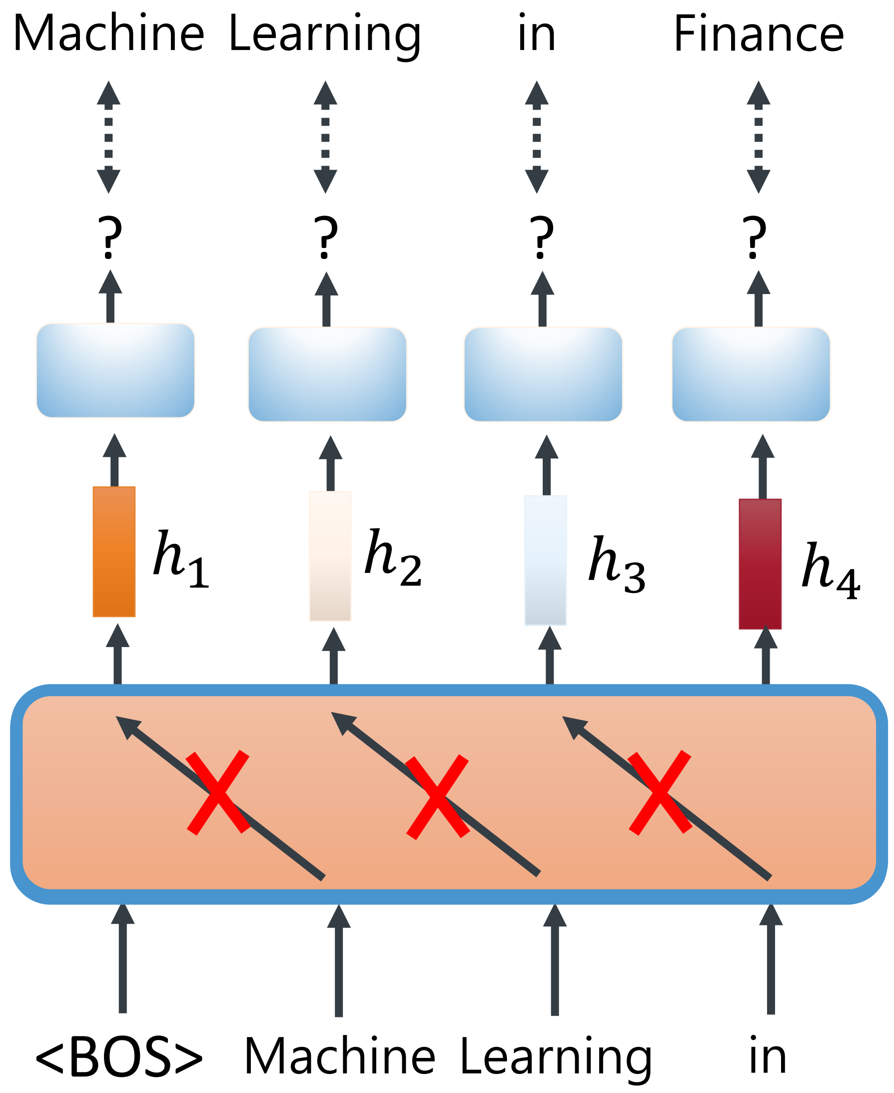
Apple's headquarters is located in California.



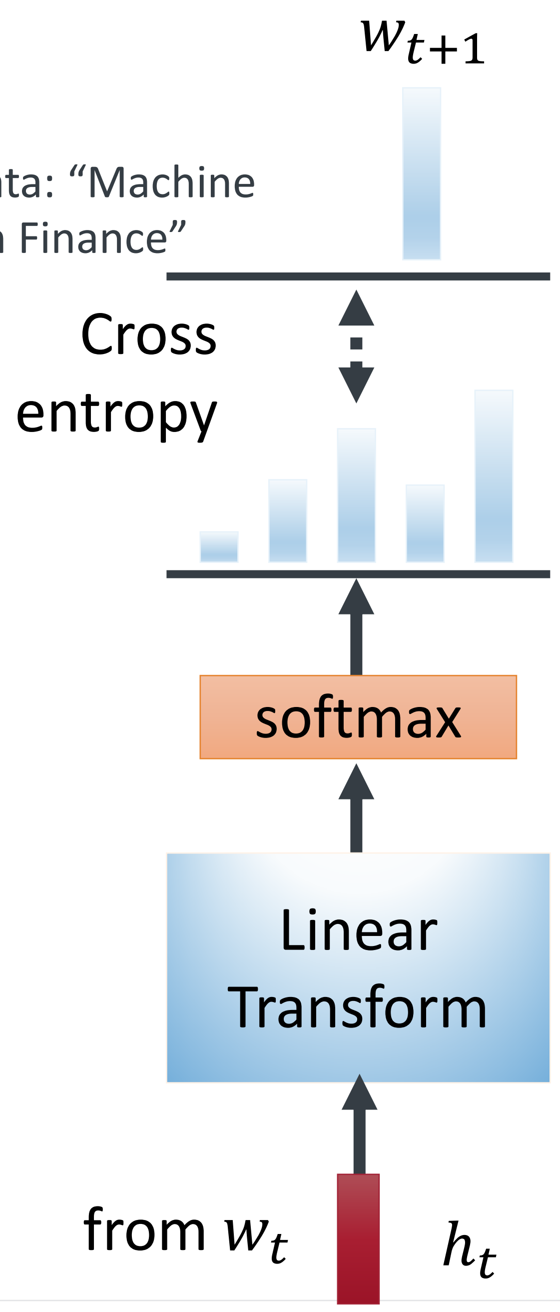
Generative Pre-trained Transformer (GPT)

Generative Pre-trained Transformer (GPT)

Training GPT: Next Token Prediction

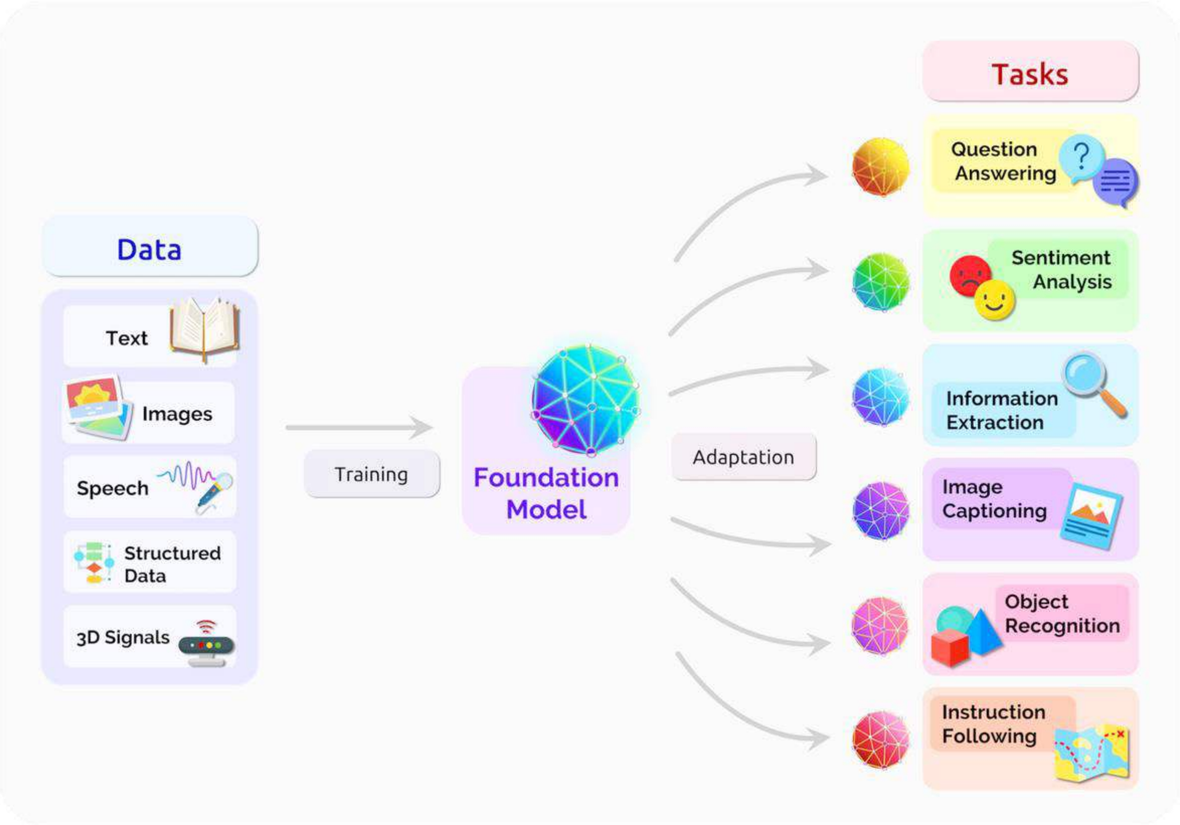


Training data: “Machine Learning in Finance”

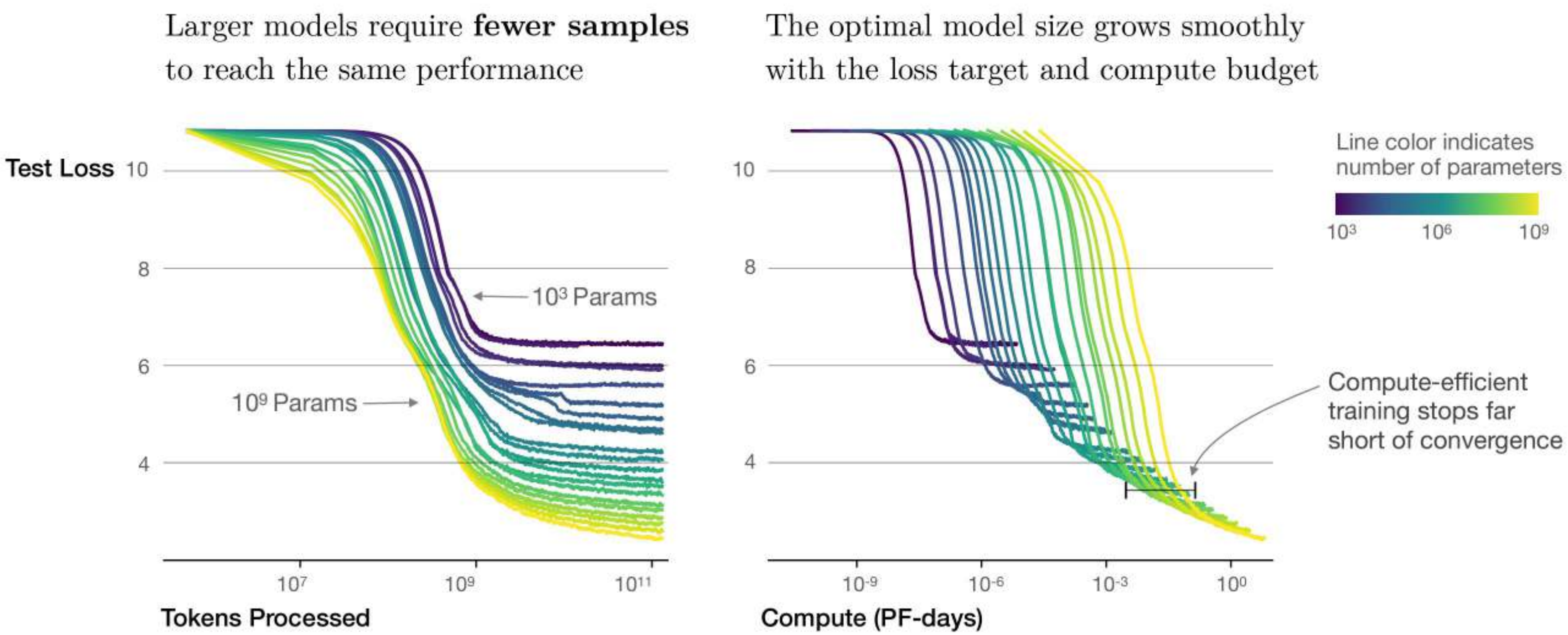


Pretraining: Scaling Unsupervised Learning on the Internet

- Key recipes for successful pretraining
  - A scalable model - Transformer
  - Massive training data – The Internet & Self-supervised learning
  - Compute - GPUs



Scaling Laws



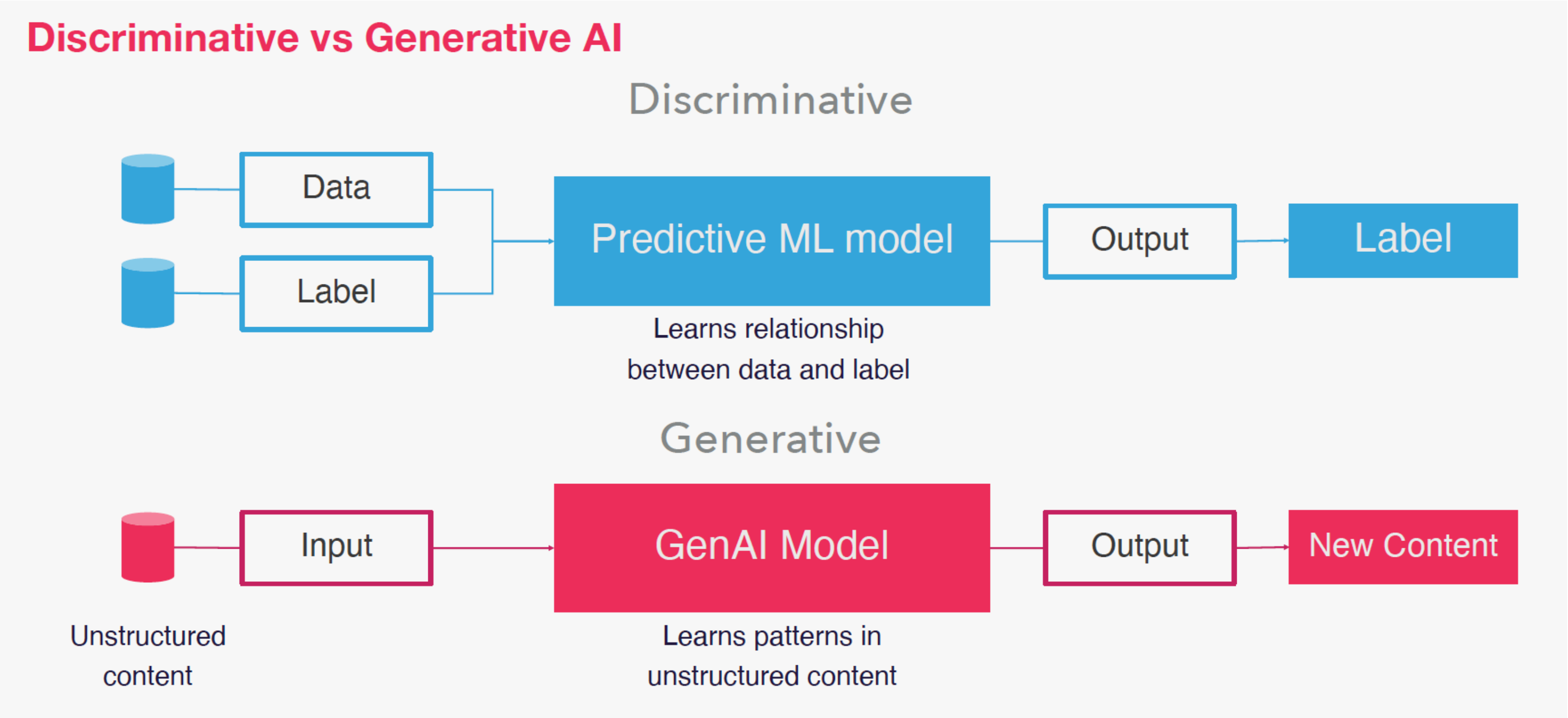
[PDF] **Scaling laws** for **neural language models**

J Kaplan, S McCandlish, T Henighan, TB Brown… - arXiv preprint arXiv …, 2020 - arxiv.org

… We study empirical **scaling laws** for **language model** performance on the cross-entropy loss. The loss scales as a power-**law** with **model** size, dataset size, and the amount of compute …

☆ Save 99 Cite Cited by 3066 Related articles All 8 versions

Discriminative vs Generative AI



Acknowledgement

The lecture note has benefited from various resources, including those listed below. Please contact Zonghao Yang (zyang99@stevens.edu) with any questions or concerns about the use of these materials.

- Lecture Notes on self-attention, sequence-to-sequence modeling, and BERT from ML 2021 Spring by Hung-Yi Lee at National Taiwan University
- Lecture Notes on Pretraining from CS224N 2025 Winter at Stanford University

STEVENS
INSTITUTE OF TECHNOLOGY
1870

THANK **YOU**

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030