



香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

COMP 4901B

Large Language Models

Transformers

Junxian He

Sep 17, 2025

Please Download HKUST iLearn in Your Mobile Phone or iPad



HKUST iLearn

HKUST Learning
Designed for iPad. Not verified



Canvas

This will open the 'Canvas Student' app which provides an easy access to the online content of your courses at HKUST - watch videos, post to discussions, submit quizzes, etc.



SFQ

Allows you to complete the Student Feedback Questionnaire for all your courses at HKUST on the move.



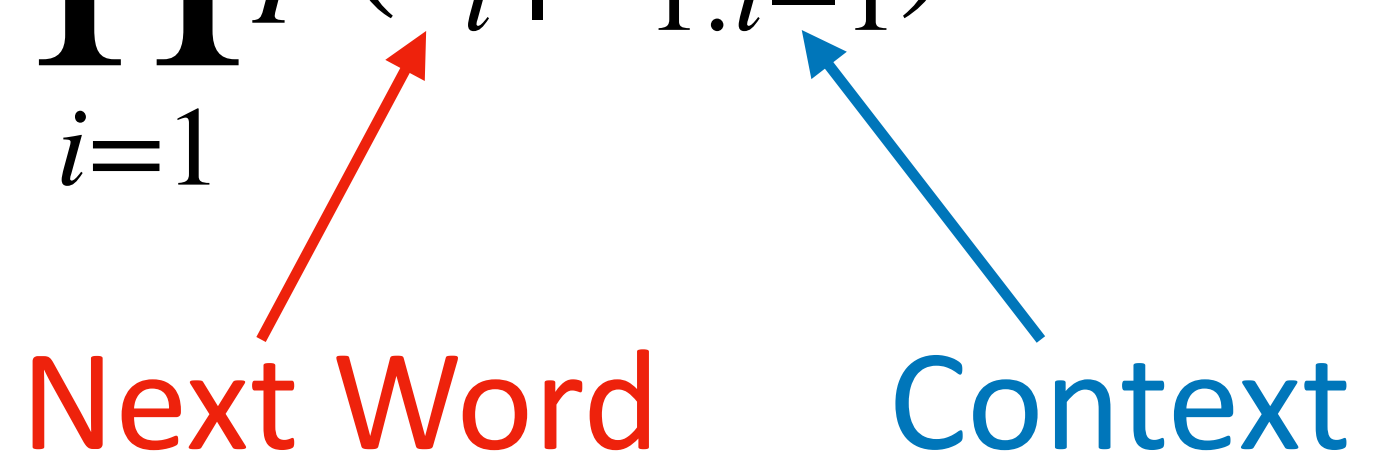
iPRS

Enables you to quickly respond to questions or polls created by your instructor in class.

We are going to use iPRS to do quizzes in the future

Recap: Autoregressive Language Models

$$\begin{aligned} p(\text{the, mouse, ate, the, cheese}) &= p(\text{the}) \\ &\quad p(\text{mouse} \mid \text{the}) \\ &\quad p(\text{ate} \mid \text{the, mouse}) \\ &\quad p(\text{the} \mid \text{the, mouse, ate}) \\ &\quad p(\text{cheese} \mid \text{the, mouse, ate, the}). \end{aligned}$$

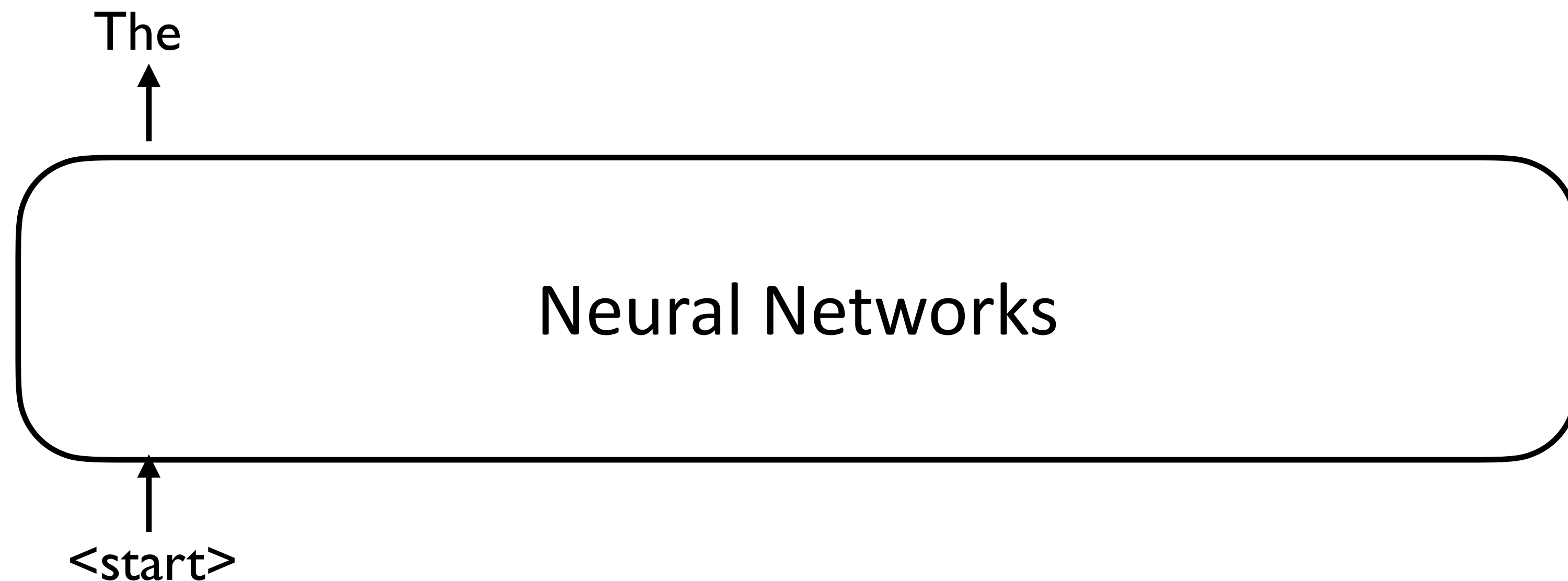
$$p(x_1, x_2, \dots, x_I) = \prod_{i=1}^I p(x_i \mid x_{1:i-1})$$


The diagram illustrates the components of the autoregressive model equation. A red arrow points from the text "Next Word" to the variable x_i in the probability term $p(x_i \mid x_{1:i-1})$. A blue arrow points from the text "Context" to the sequence $x_{1:i-1}$ in the same term.

Recap: Neural Language Models

Neural language models are typically autoregressive

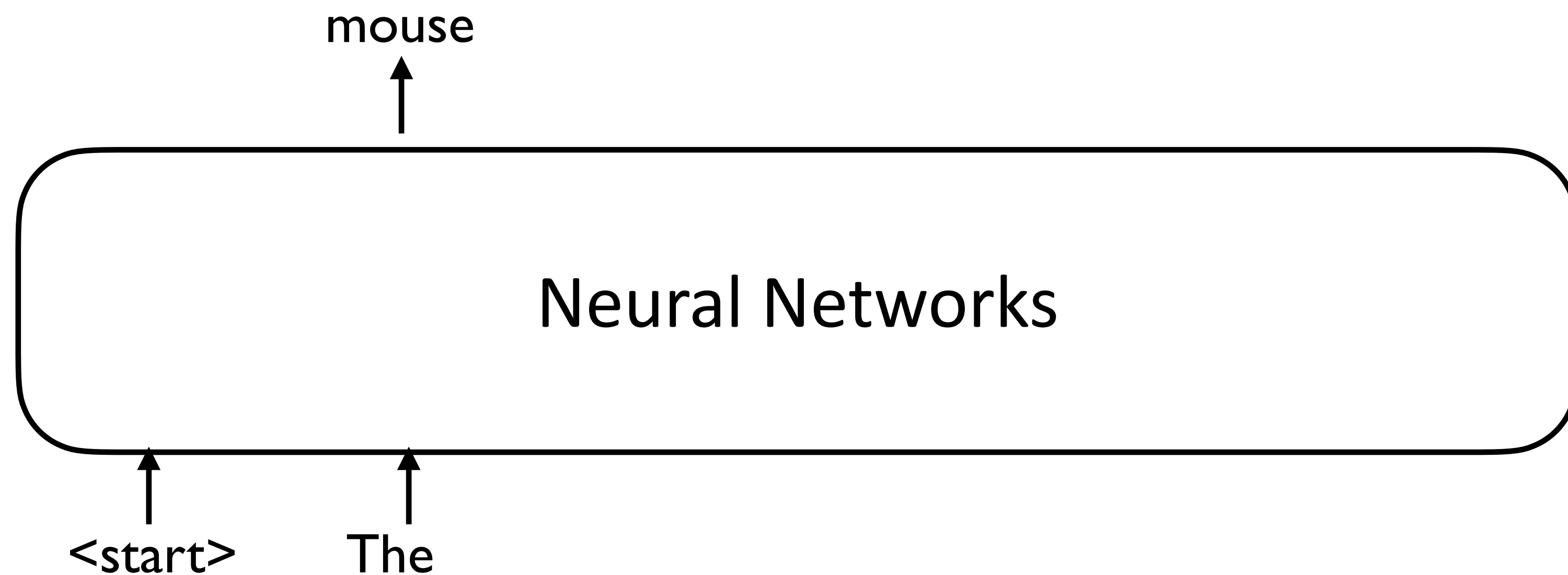
Data: “The mouse ate the cheese .”



Recap: Neural Language Models

Neural language models are typically autoregressive

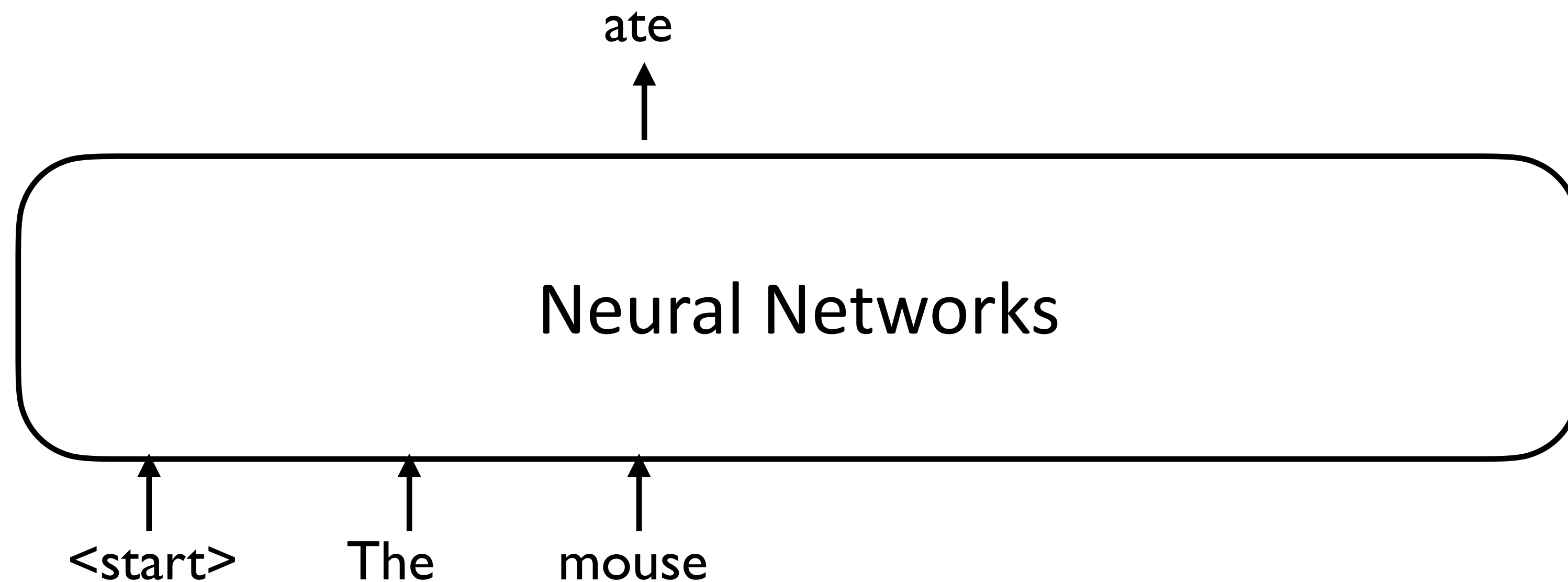
Data: “The mouse ate the cheese .”



Recap: Neural Language Models

Neural language models are typically autoregressive

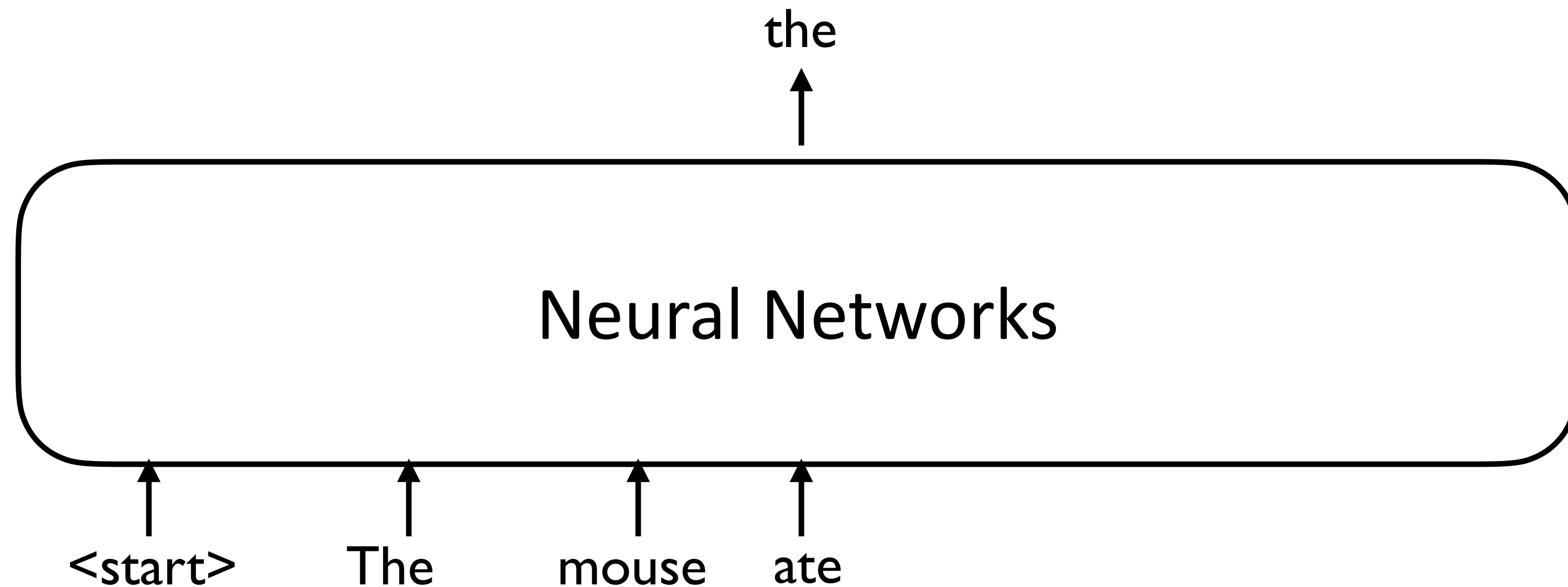
Data: “The mouse ate the cheese .”



Recap: Neural Language Models

Neural language models are typically autoregressive

Data: “The mouse ate the cheese .”

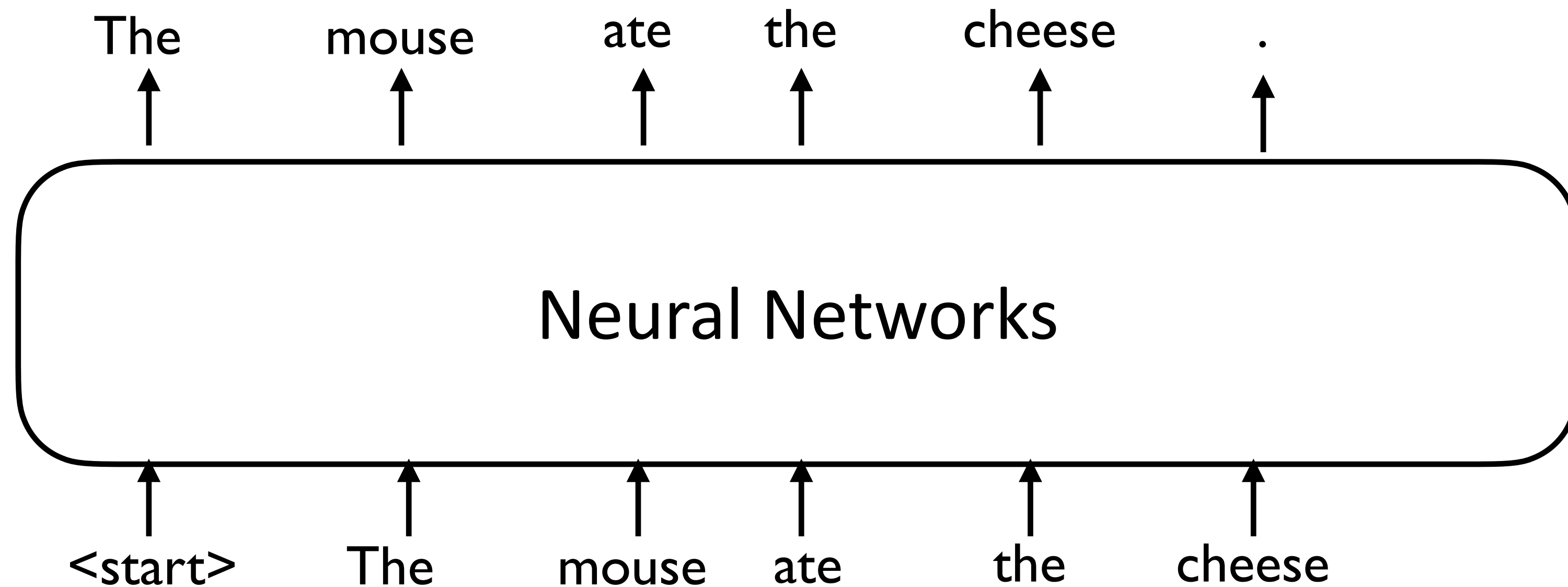


We can compute the loss on every token in parallel

Recap: Neural Language Models

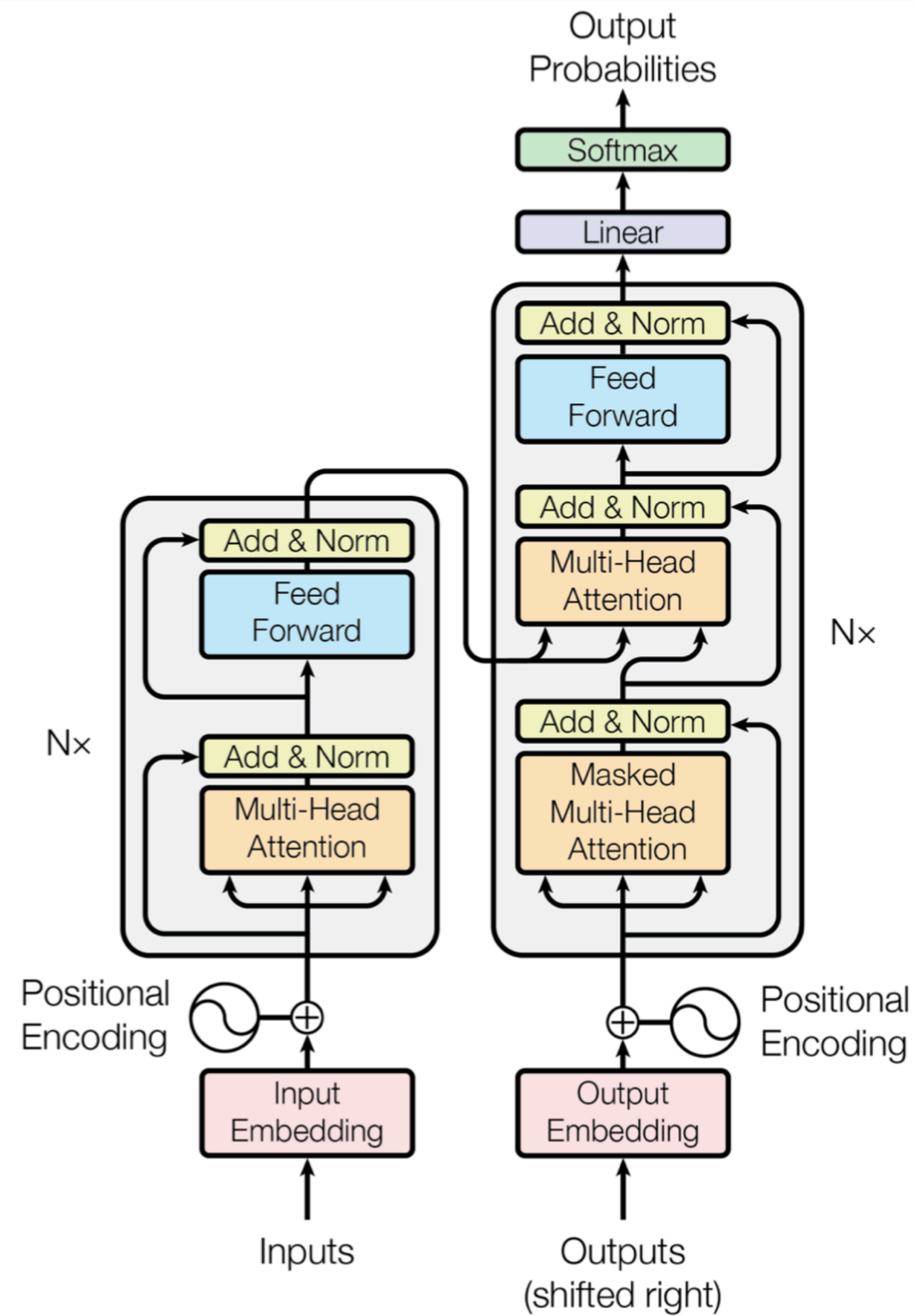
Neural language models are typically autoregressive

Data: “The mouse ate the cheese .”

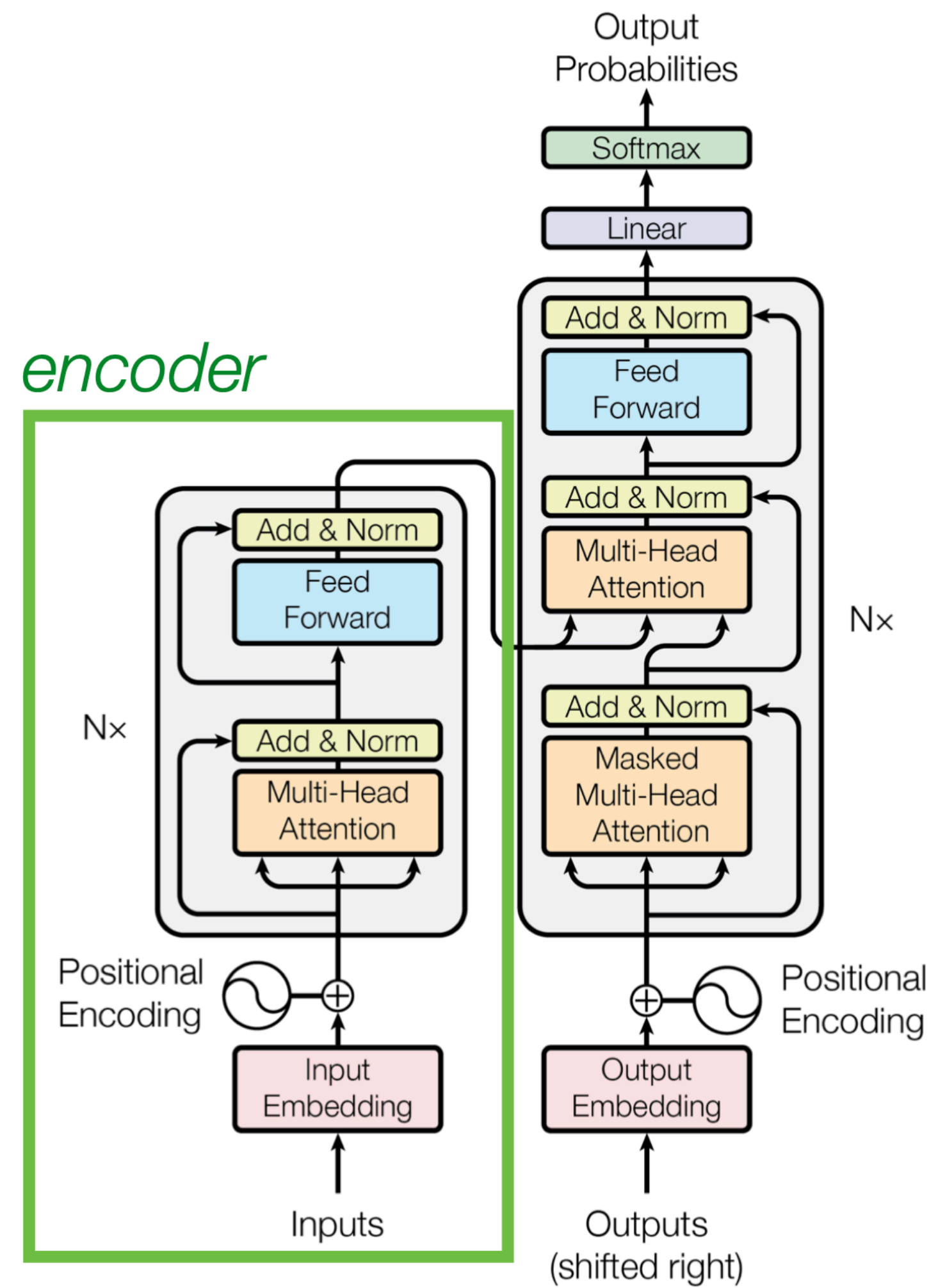


Each prediction only sees the inputs on its left

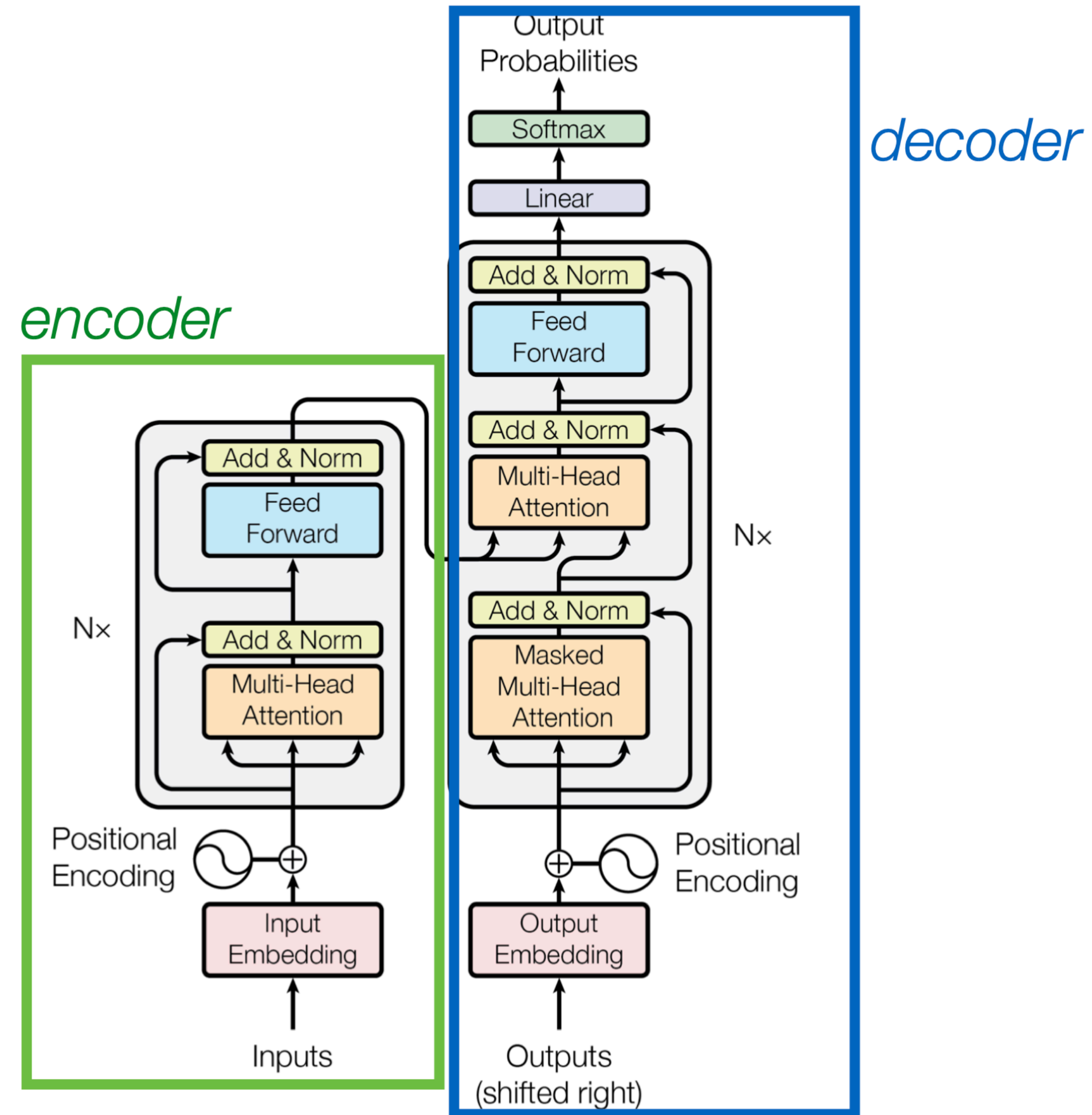
Recap: Transformer



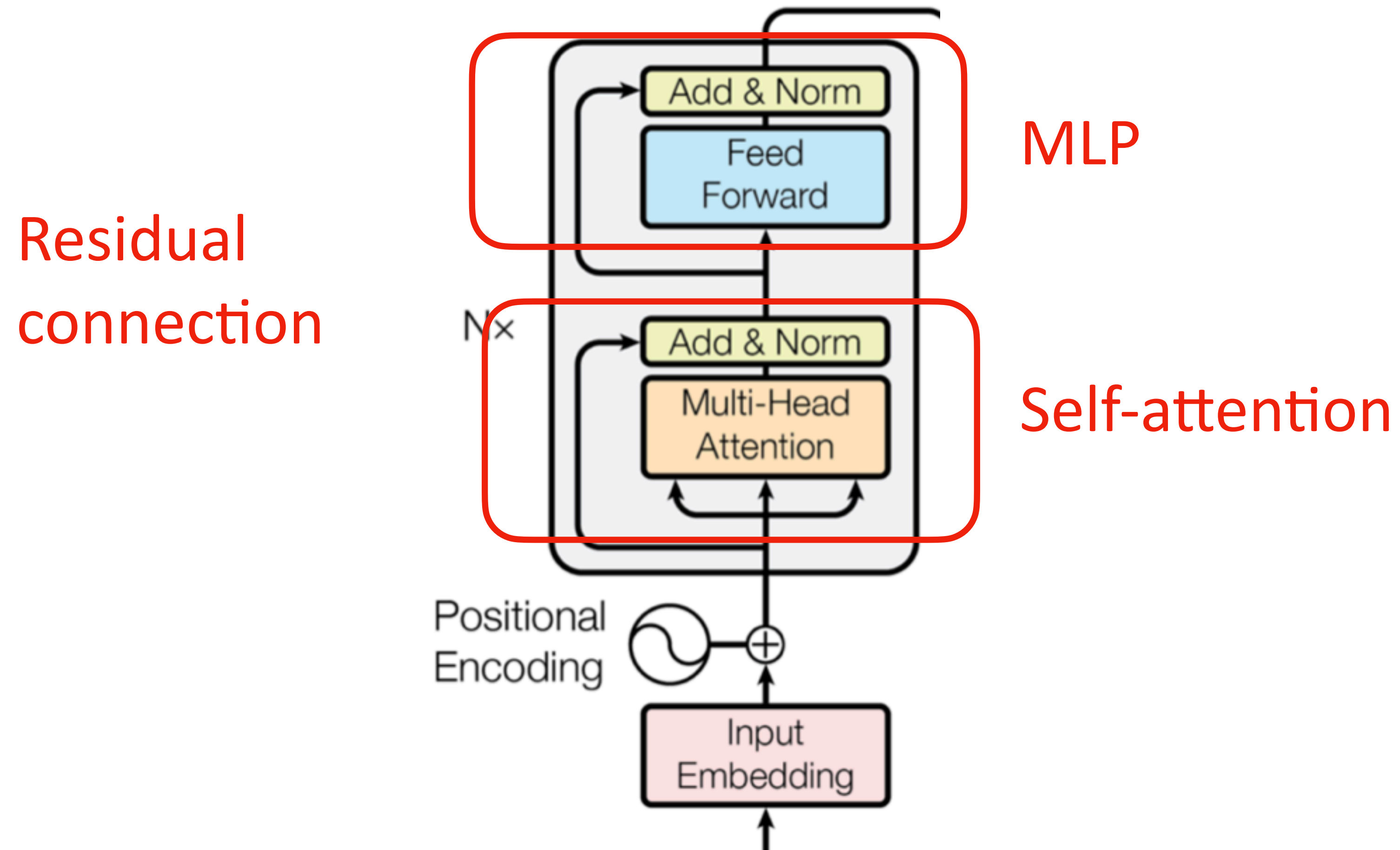
Recap: Encoder



Recap: Decoder



Recap: Transformer Encoder

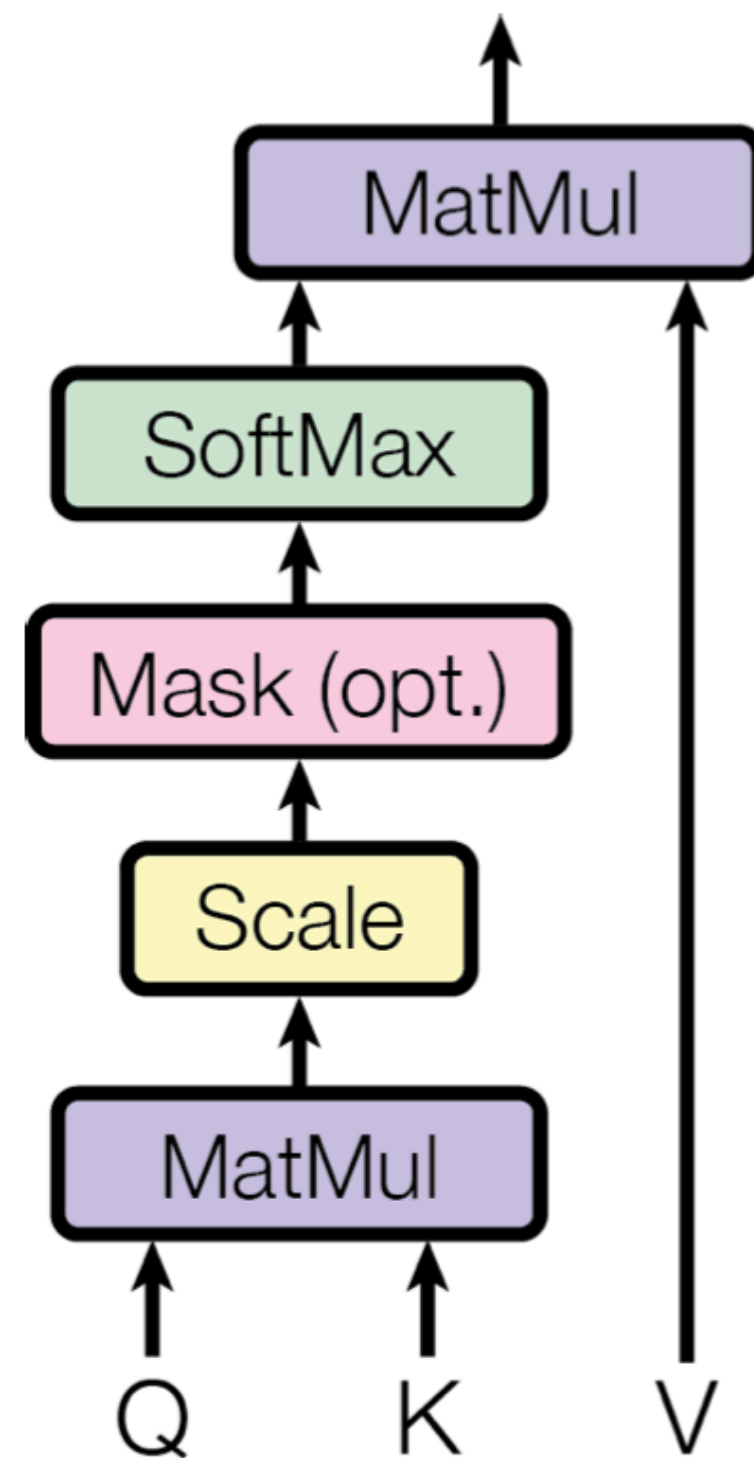


Recap: What is Attention

$$Q \in R^{n \times d} \quad K \in R^{m \times d} \quad V \in R^{m \times d}$$

We have n queries, m (key, value) pairs

Scaled Dot-Product Attention



Q: Query
K: key
V: value

$$\text{Attention weight} = \text{softmax}(QK^T)$$

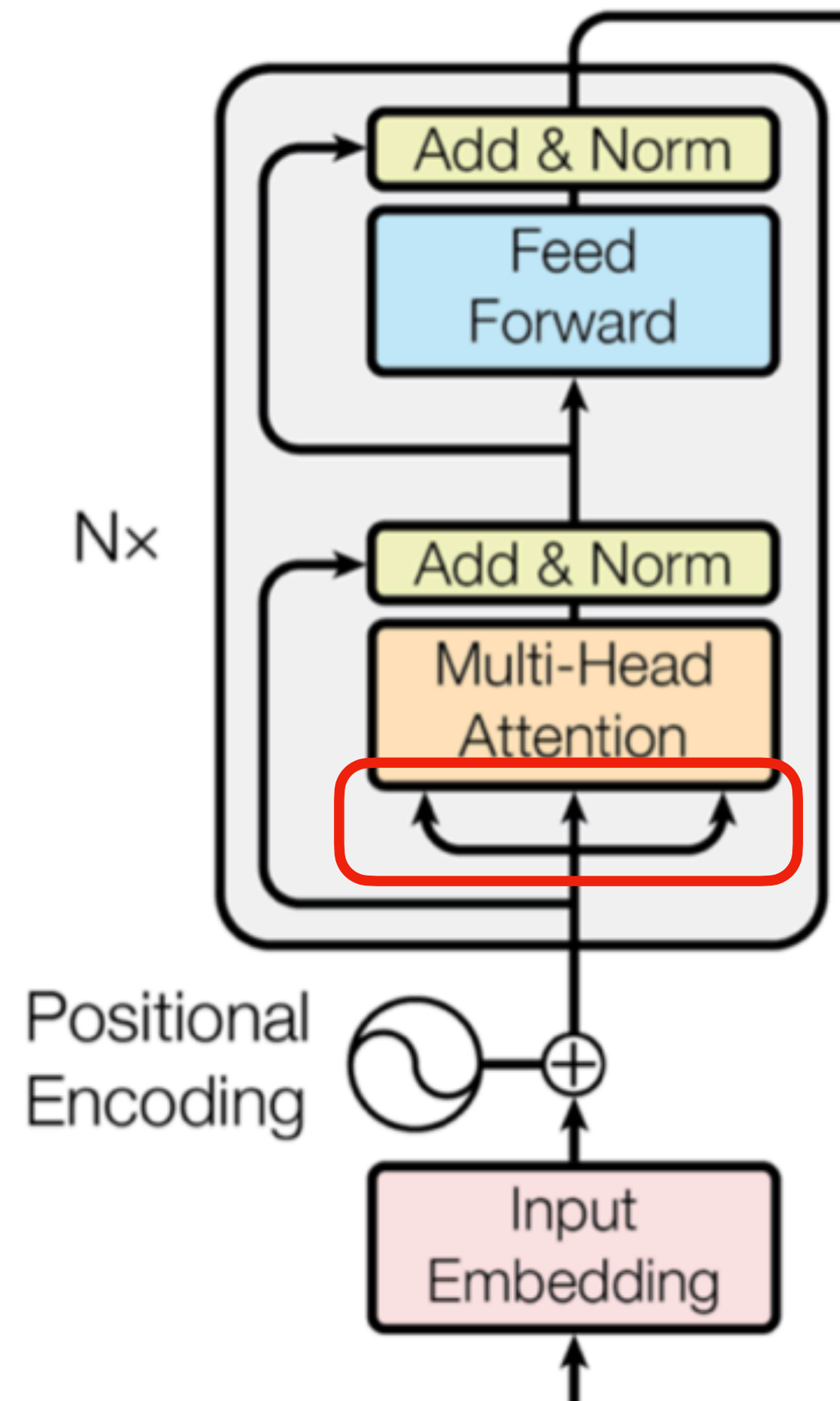
Dot-products grow large in magnitude

$$\text{Scaled Attention weight} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad \text{Shape is } m \times n$$

Attention weight represents the strength to “attend” values V

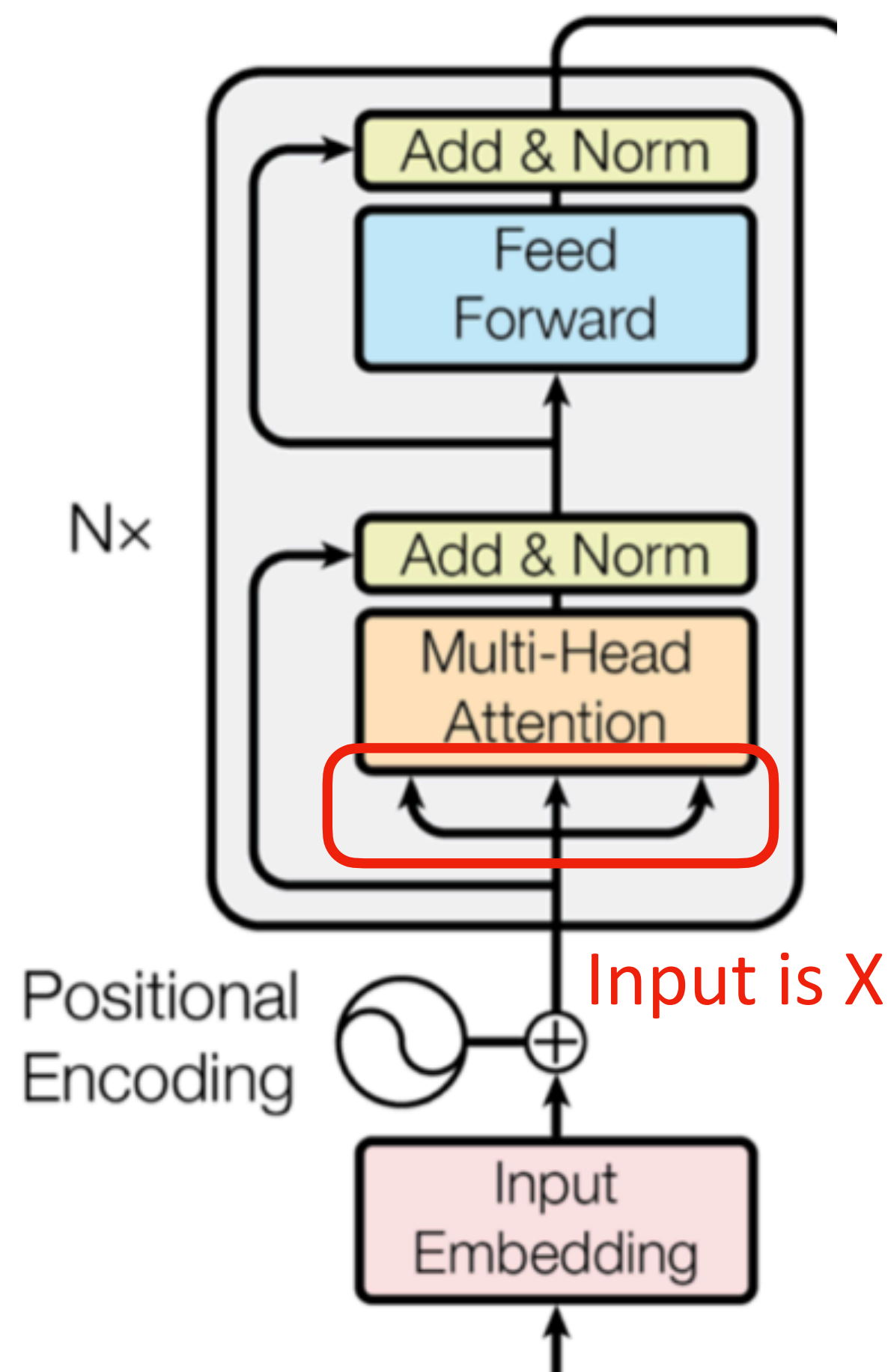
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Q, K, V

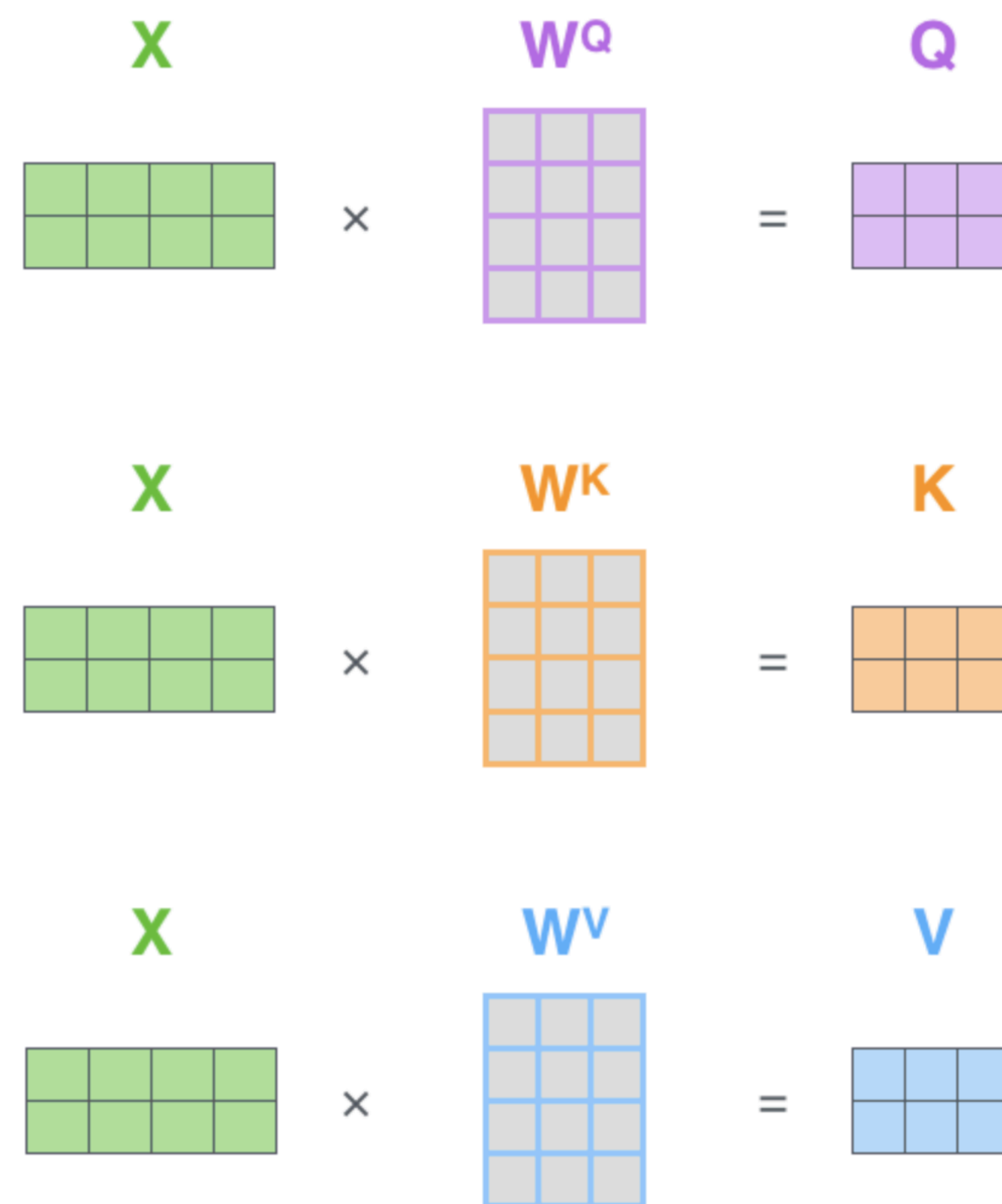


What are Q, K, V in the transformer

Self-Attention



15



15

Query, key, and value are from the same input, thus it is called “self”-attention

$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V$$

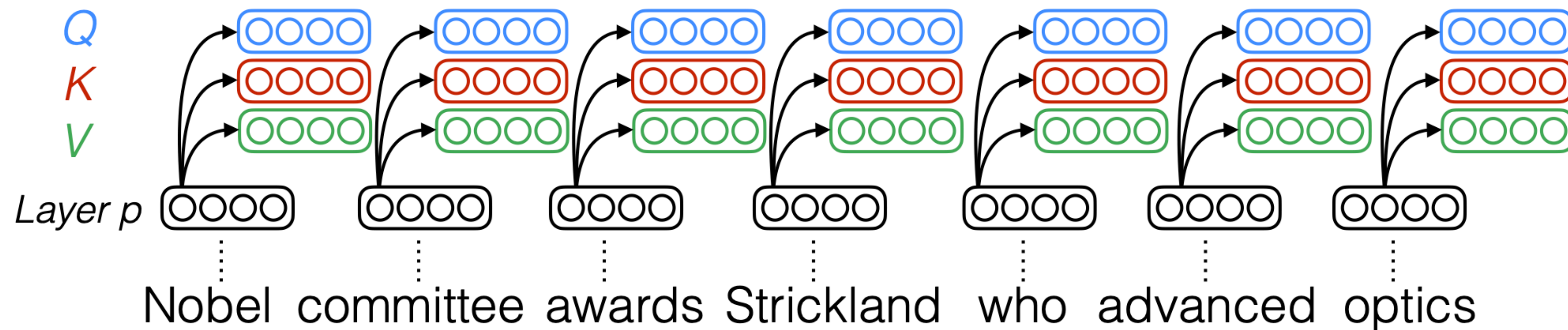
$$= Z$$

The diagram shows the calculation of the self-attention output Z (pink 2x3 grid). It involves the dot product of the Query matrix Q (purple 2x3 grid) and the transpose of the Key matrix K^T (orange 3x2 grid), scaled by $\sqrt{d_k}$, followed by a softmax operation and multiplication by the Value matrix V (blue 2x3 grid).

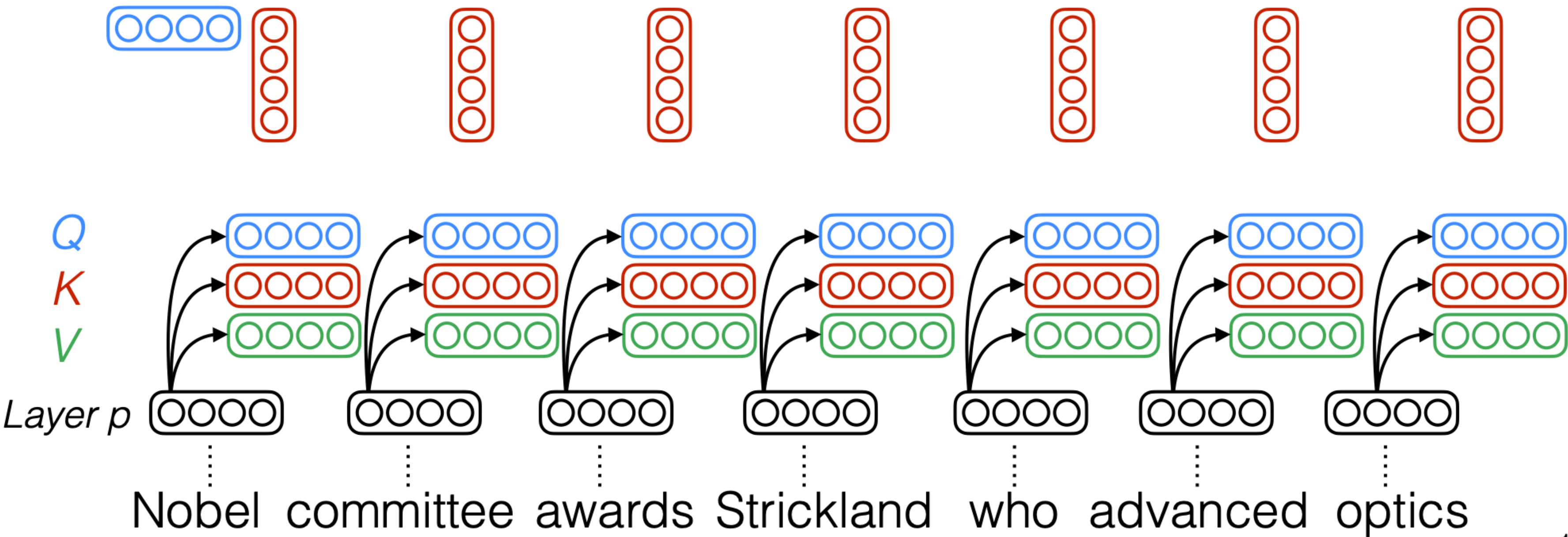
Jay Alammar. The Illustrated Transformer.

Self-Attention

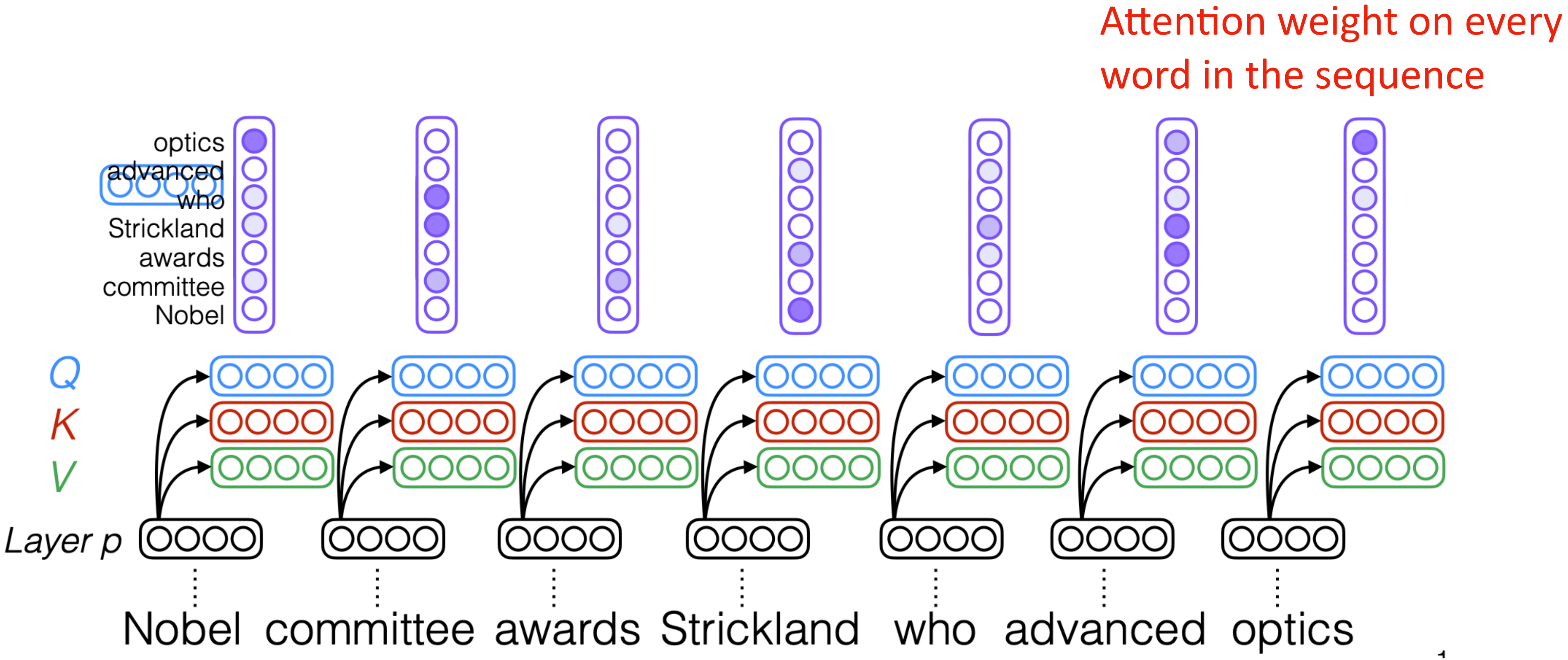
At each step, the attention computation attends to all steps in the input example



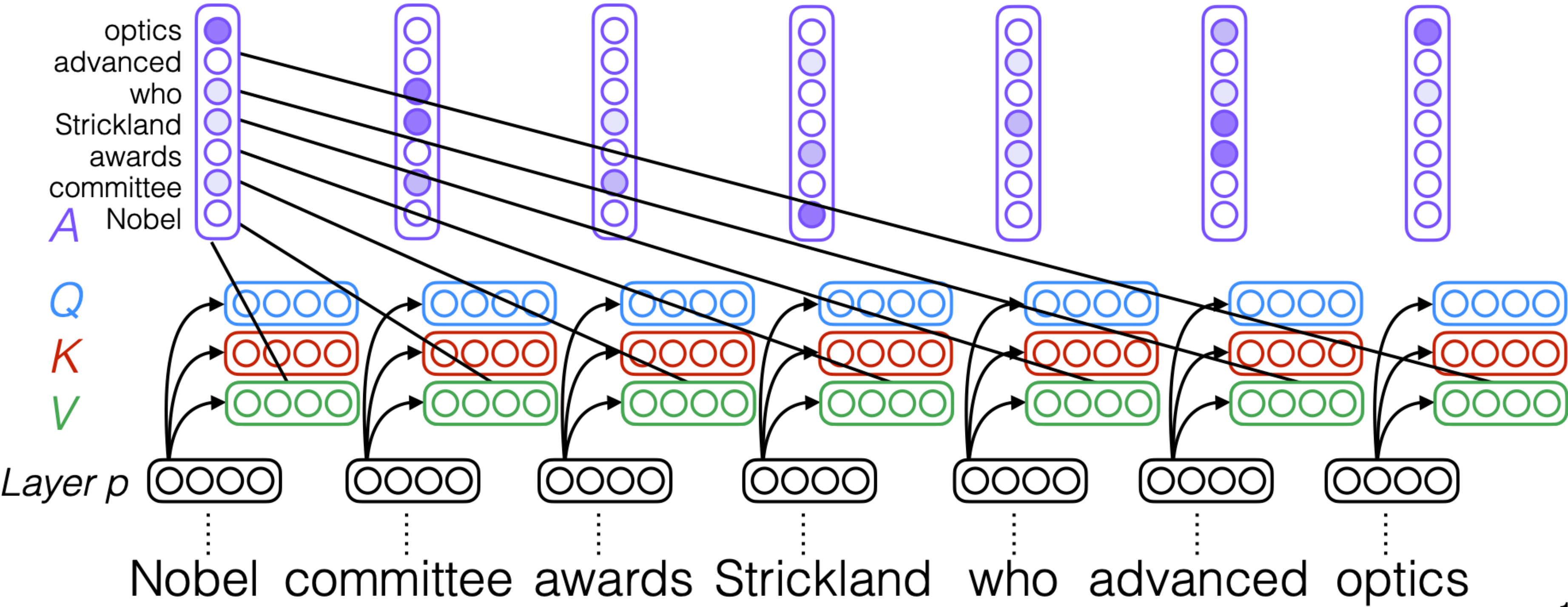
Self-Attention



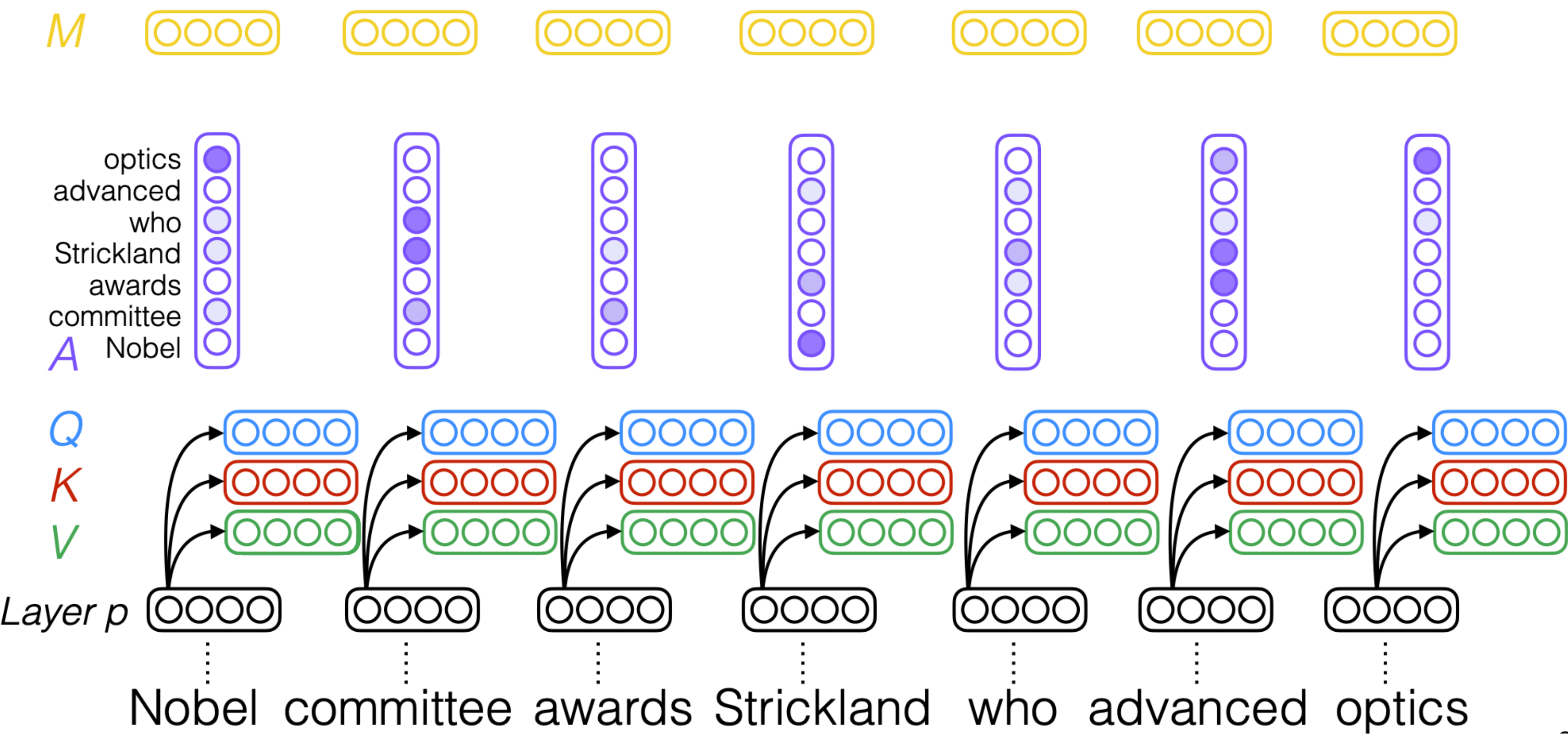
Self-Attention



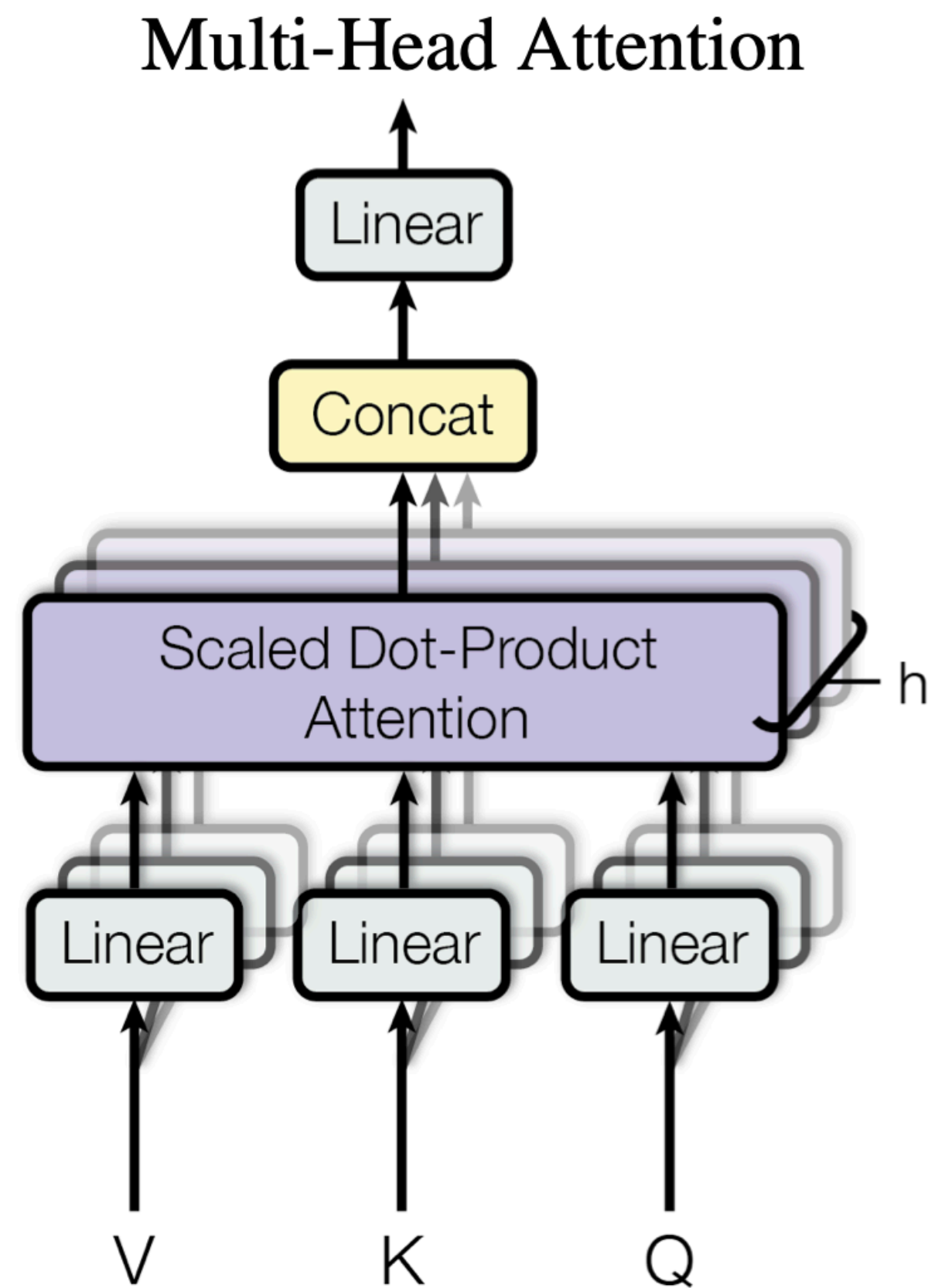
Self-Attention



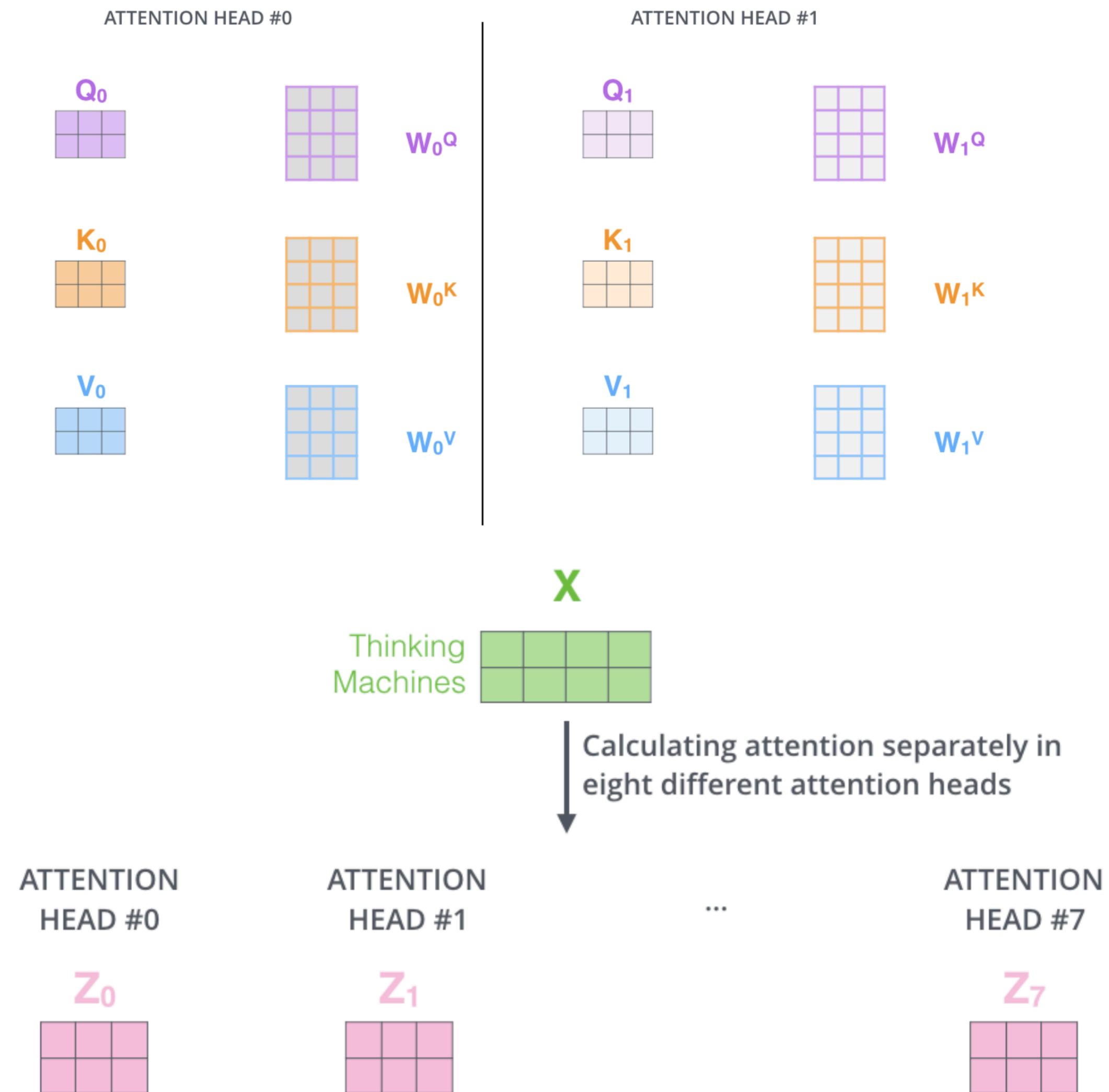
Self-Attention



Multi-Head Attention



Multi-Head Self-Attention

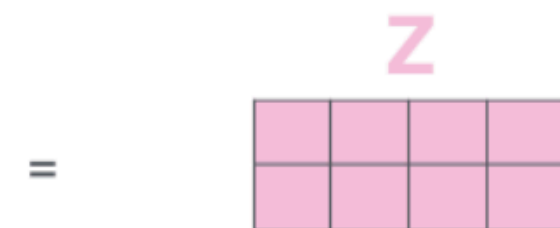


Multi-Head Self-Attention

1) Concatenate all the attention heads



3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

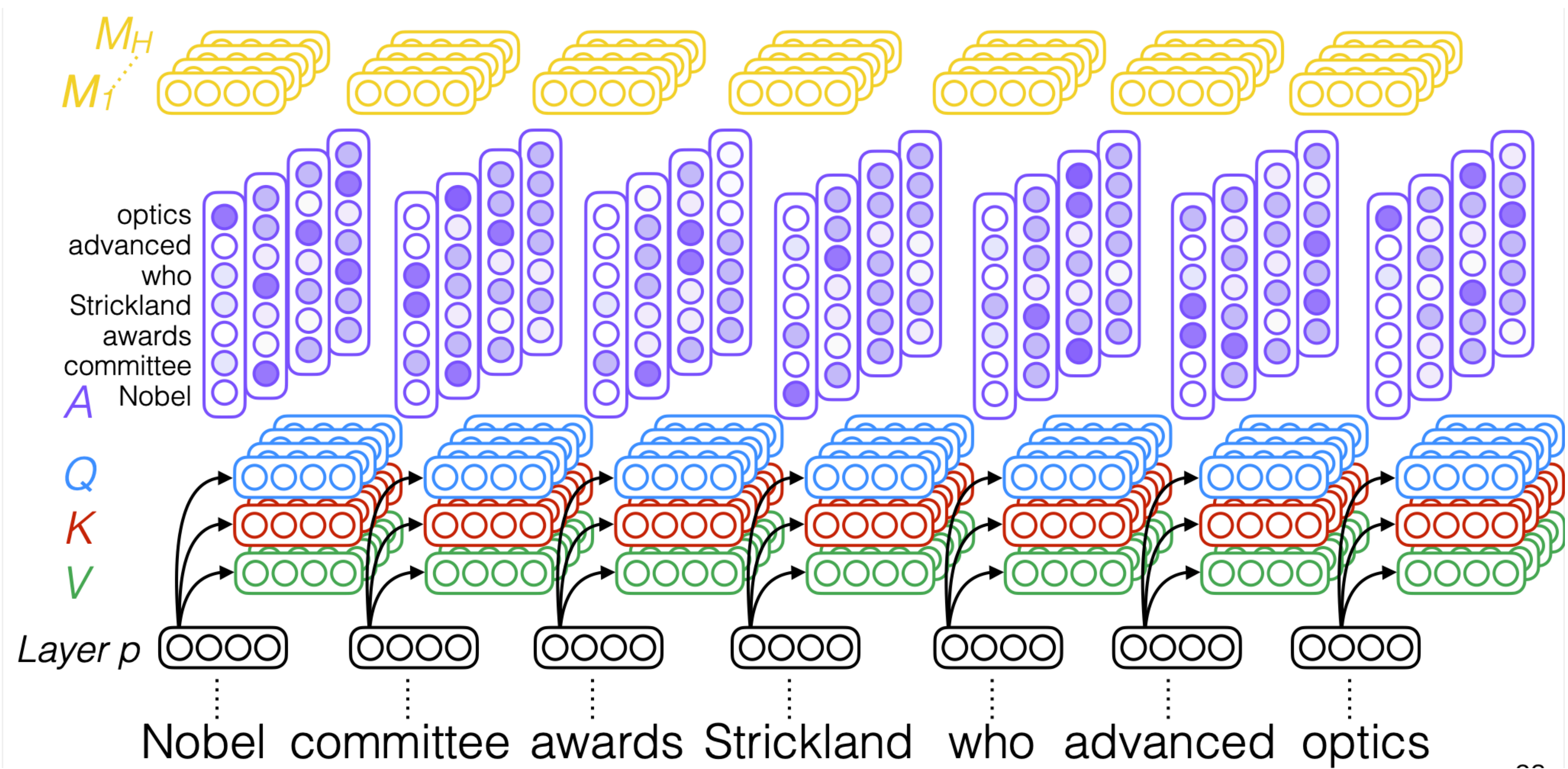


2) Multiply with a weight matrix W^O that was trained jointly with the model

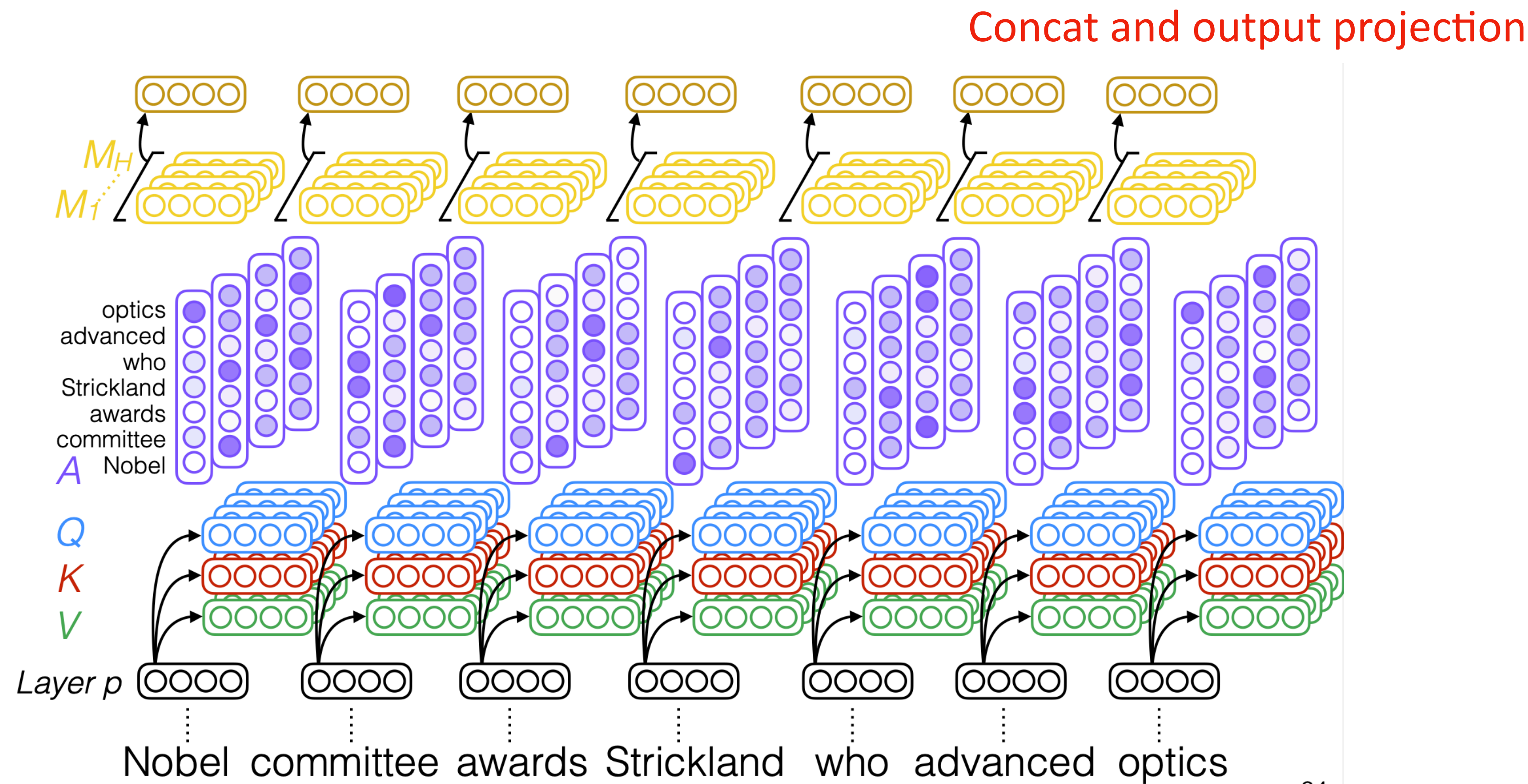
X



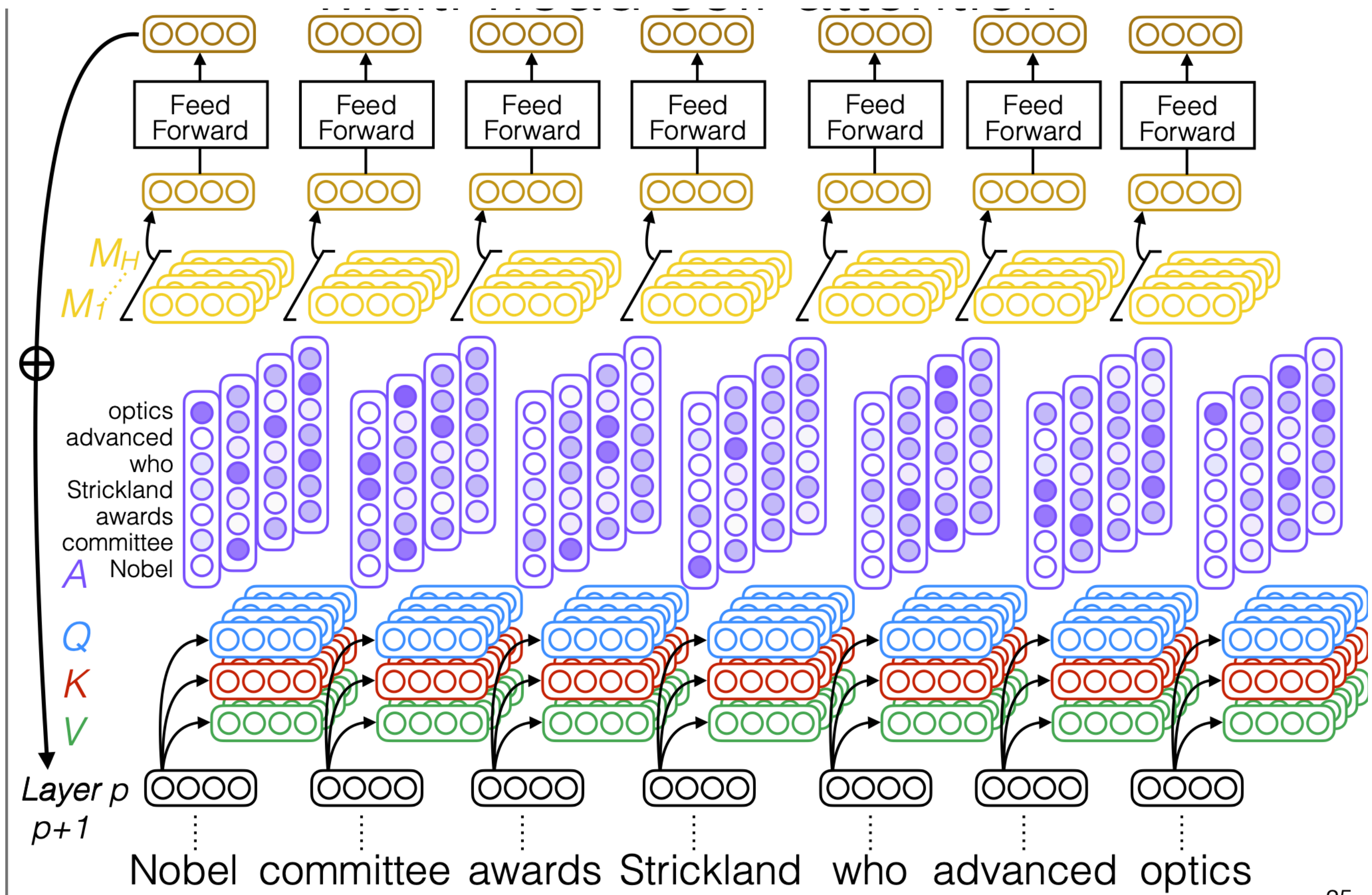
Multi-head Self-Attention



Multi-head Self-Attention

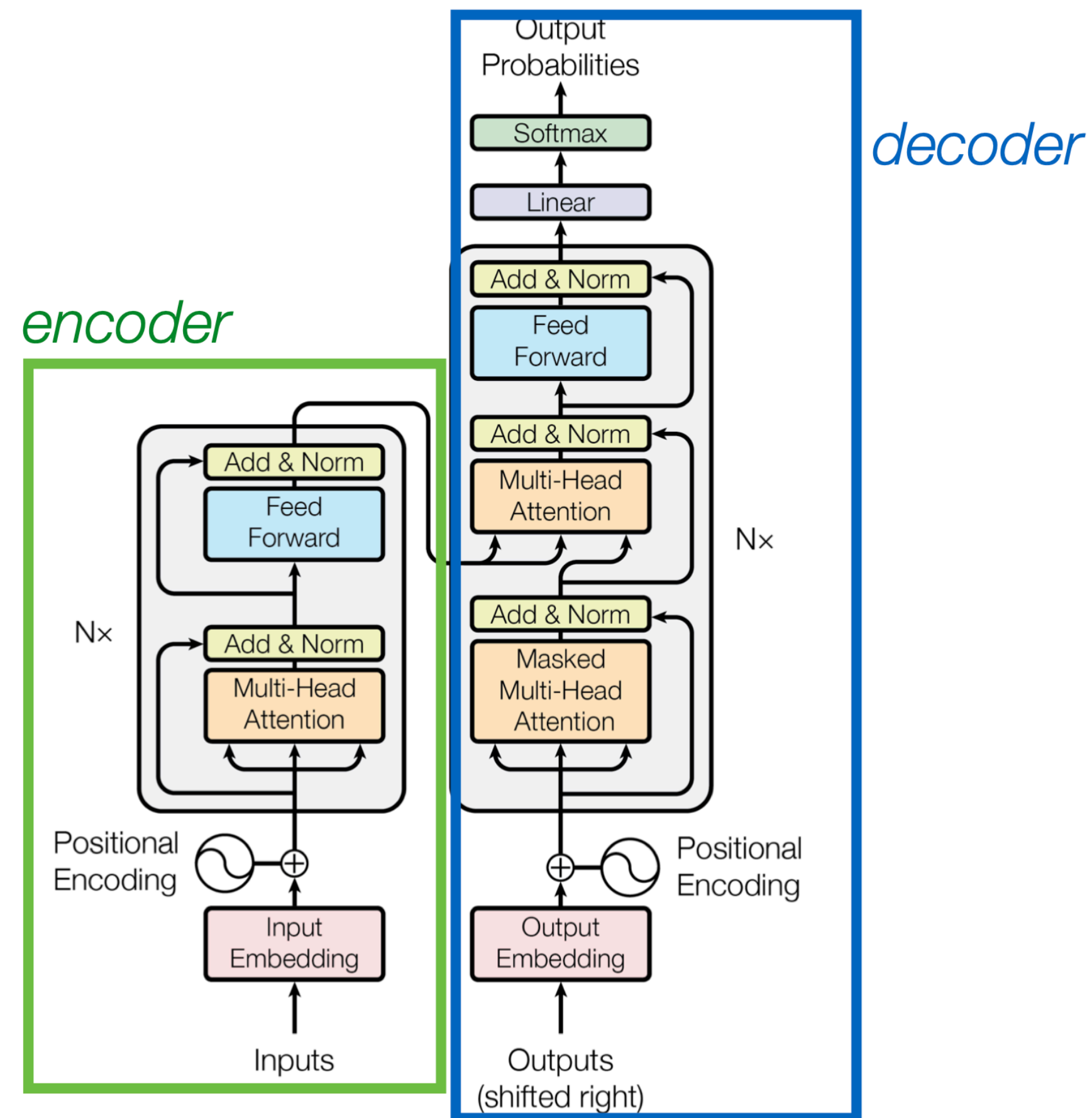
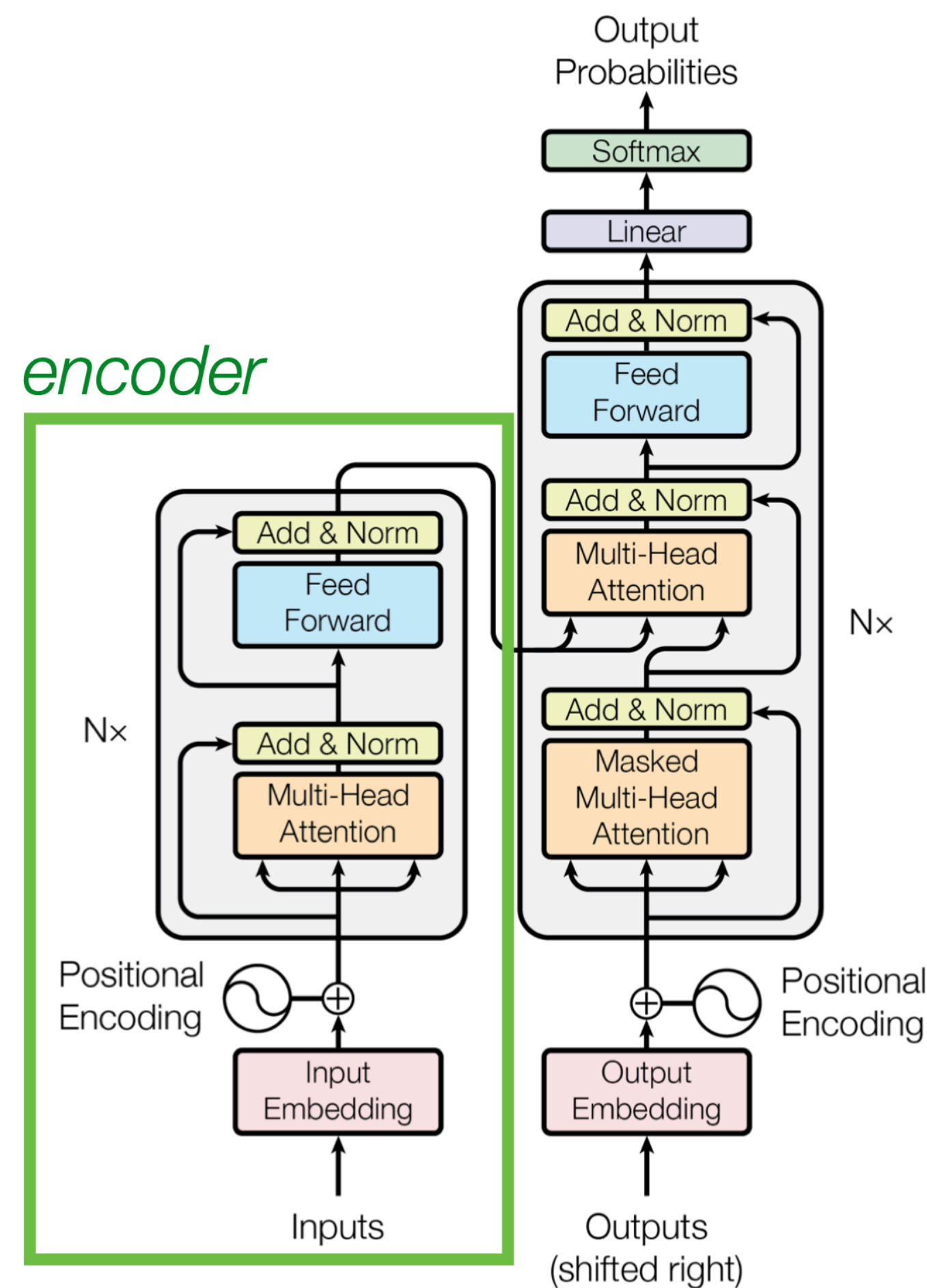


Multi-head Self-Attention + FFN



Transformer Encoder

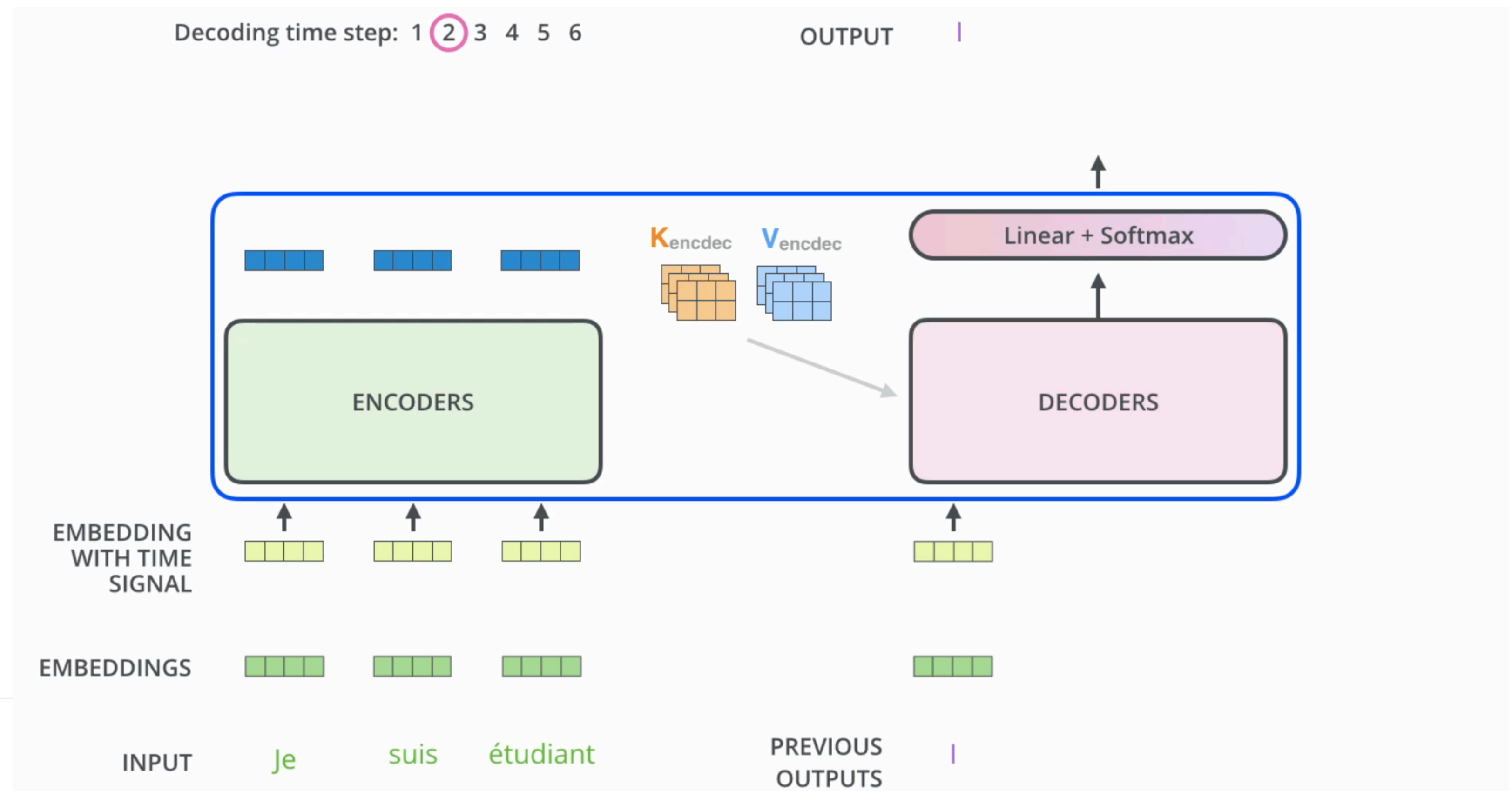
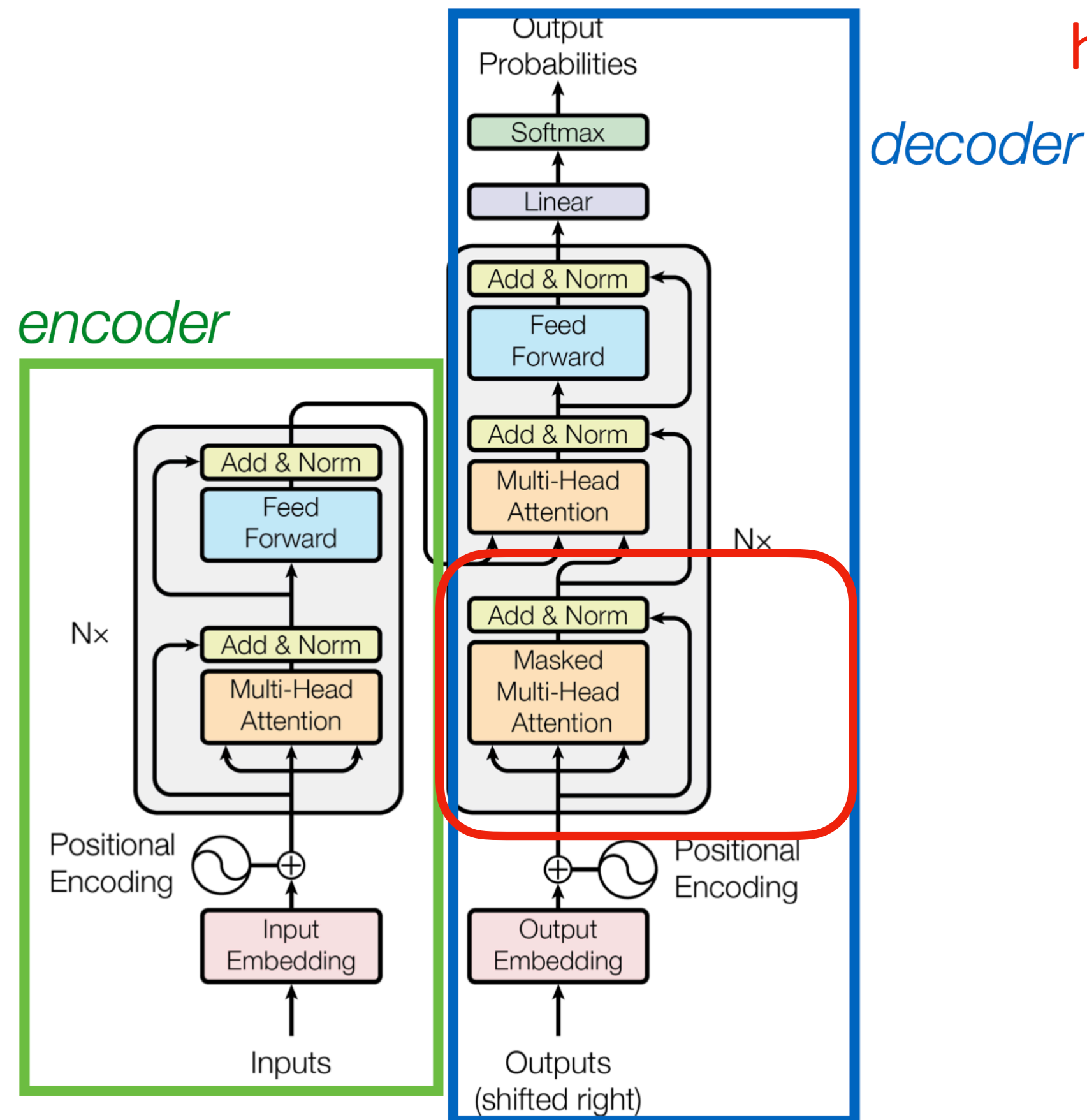
Currently we only cover the encoder side



This encoder-decoder arch is originally proposed as a seq2seq arch, for classification tasks, often only encoder is used. And language models often only have a decoder

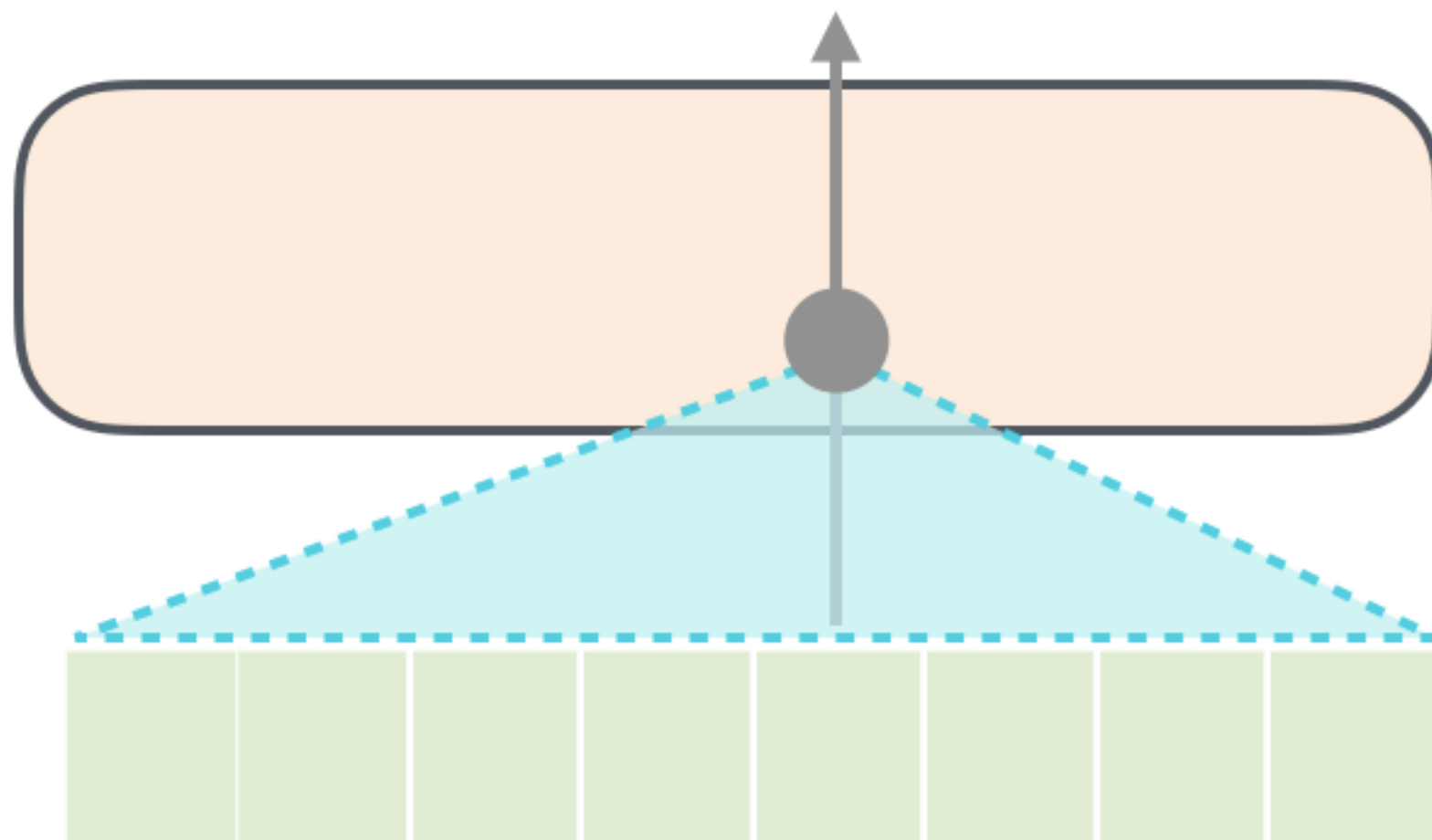
Masked Attention

Typical attention attends to the entire sequence, while masked attention only attends to the ones on the left because future words have not been generated

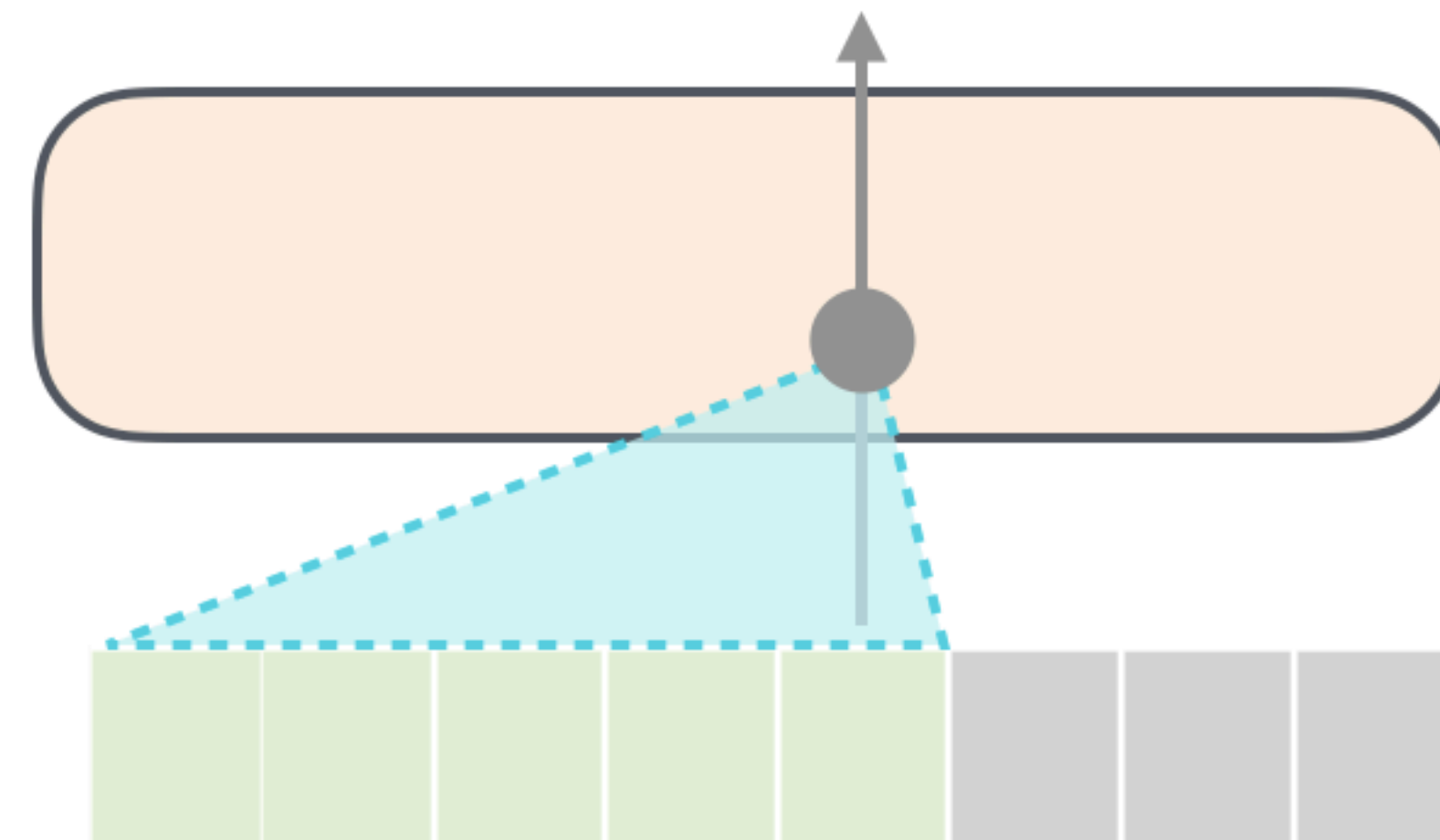


Masked Attention

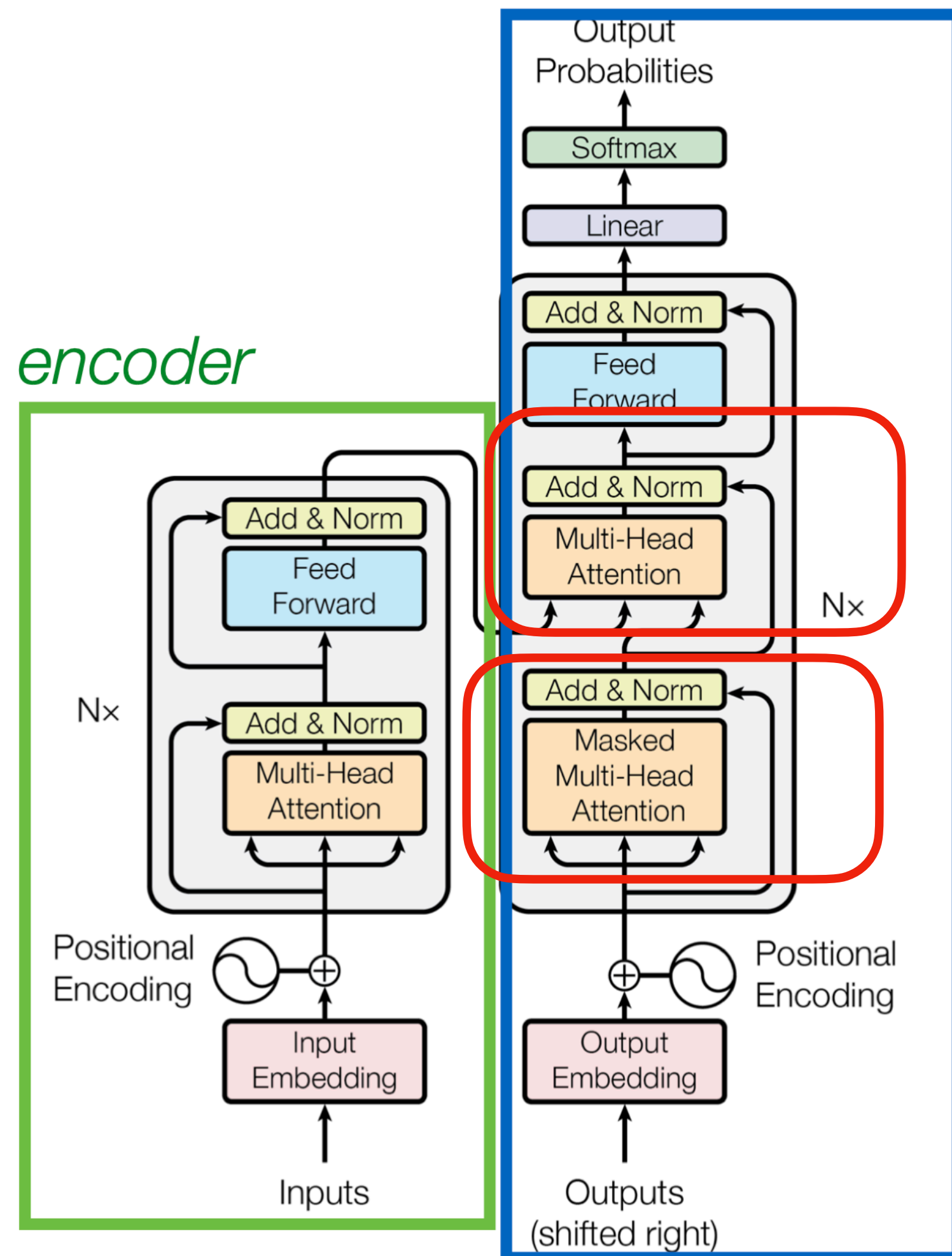
Self-Attention



Masked Self-Attention



Transformer Decoder in Seq2Seq

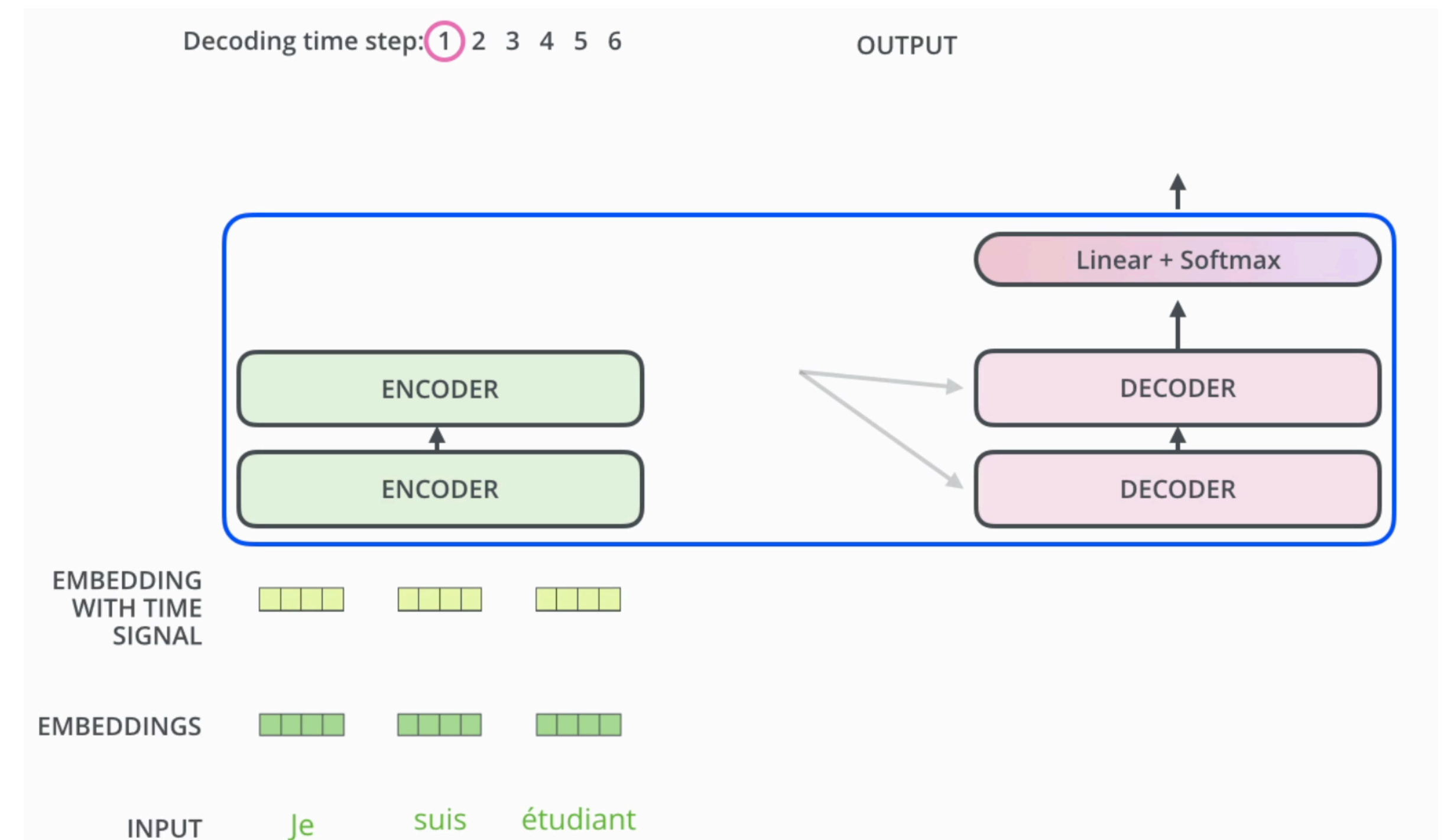


decoder

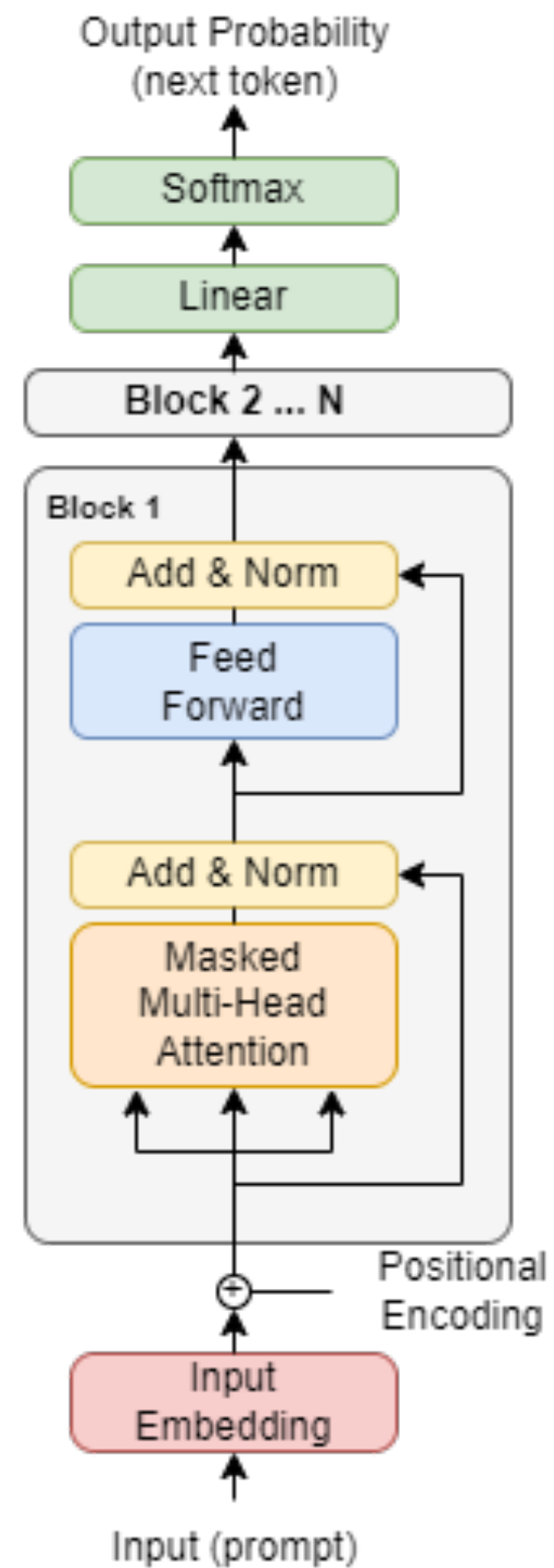
Cross-attention

Self-attention

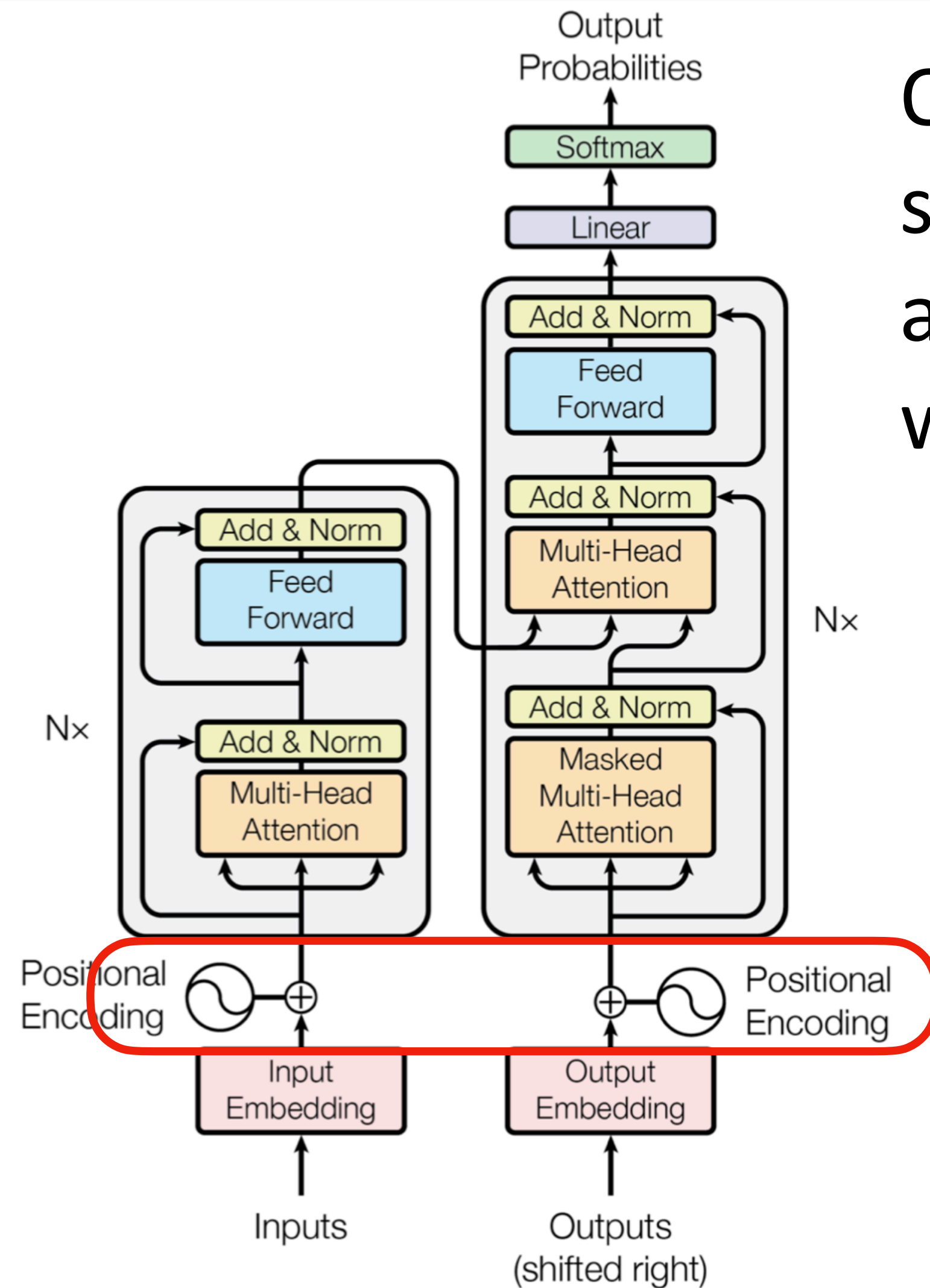
Cross-attention uses the output of encoder as input



Transformer Language Model (e.g., ChatGPT)



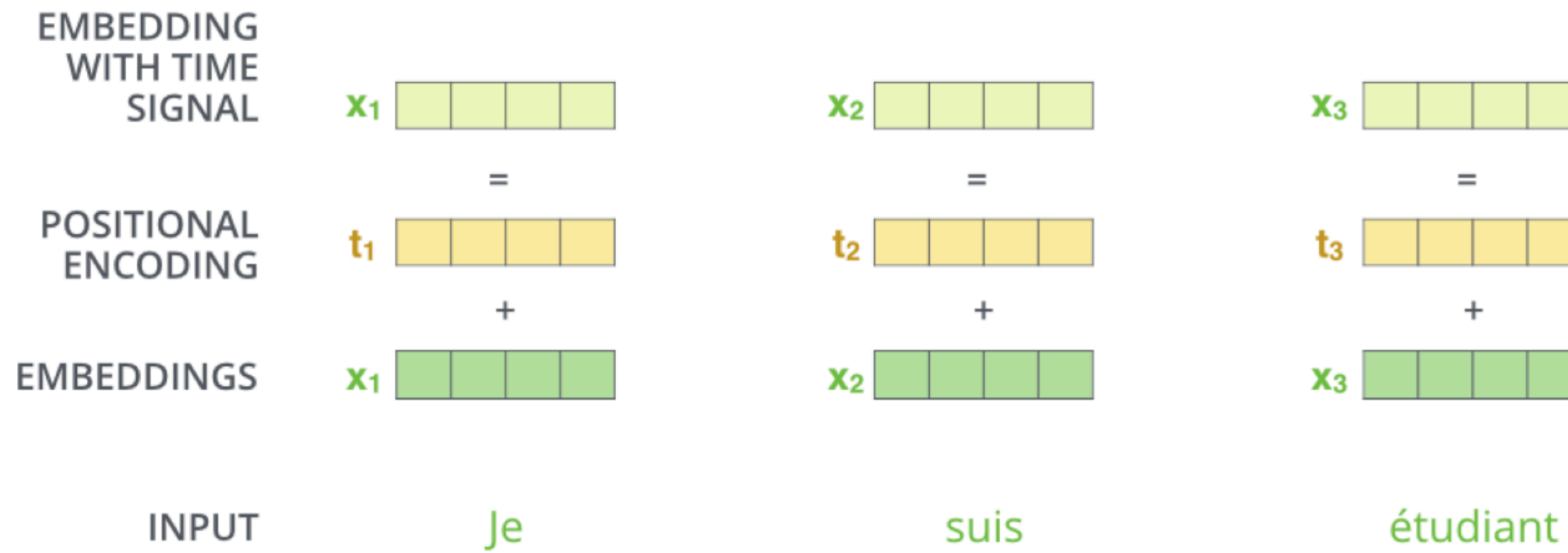
Position Embeddings



Question: If we shuffle the order of words in the sequence, will that change the attention output and feed forward output of the corresponding word?

Position embeddings are added to each word embedding, otherwise our model is unaware of the position of a word

Positional Encoding



Transformer Positional Encoding

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

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

Positional encoding is a 512d vector
 i = a particular dimension of this vector
 pos = dimension of the word
 $d_{model} = 512$

Complexity

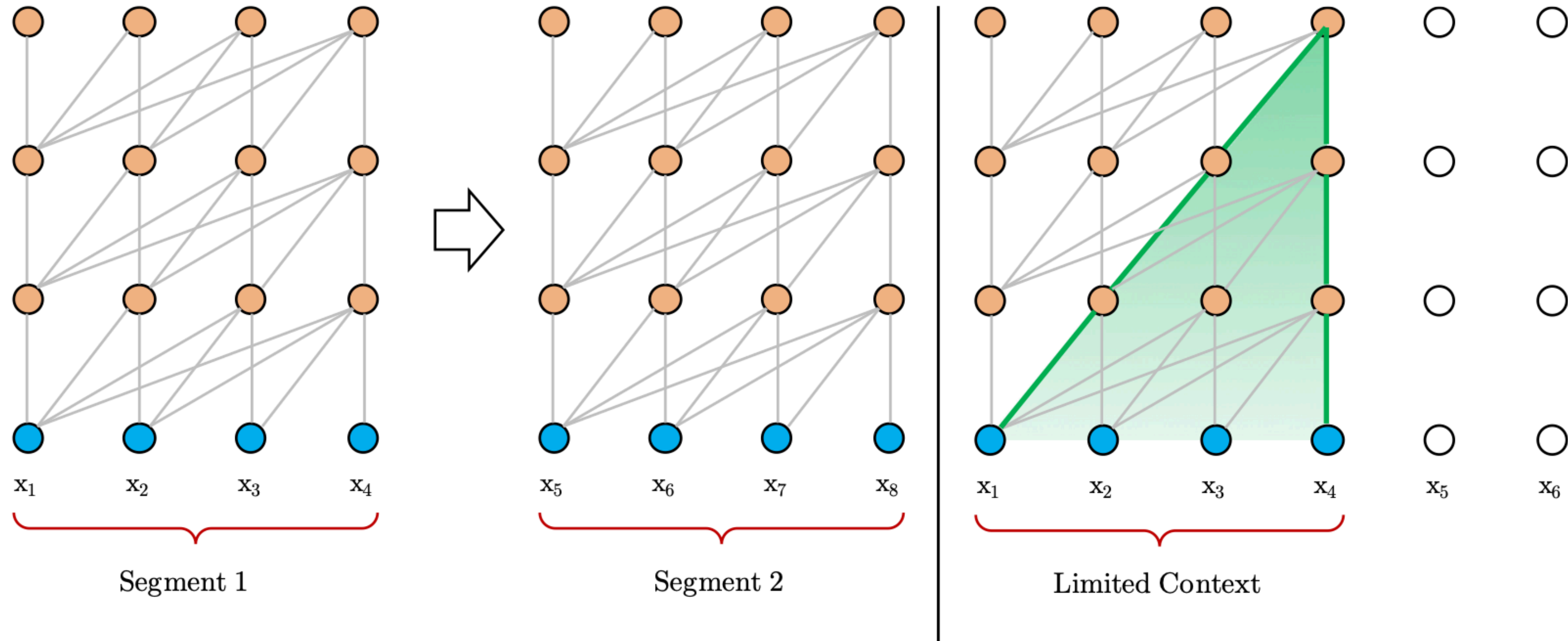
Layer Type	Complexity per Layer	Sequential Operations
Self-Attention	$O(n^2 \cdot d)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$

n is sequence length, d is embedding dimension.

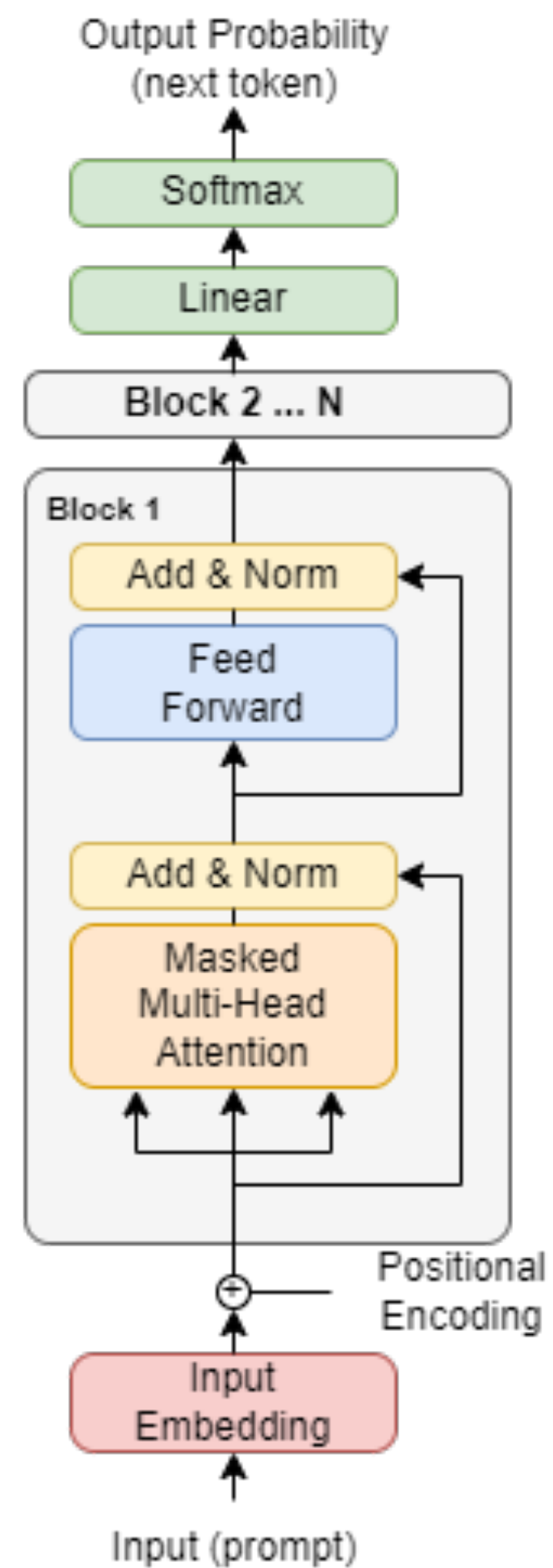
Restricted self-attention means not attending all words in the sequence, but only a restricted field

Square complexity of sequence length is a major issue for transformers to deal with long sequence

Language Model Training with Limited Context



Transformer Language Model (e.g., ChatGPT)





香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

COMP 4901B

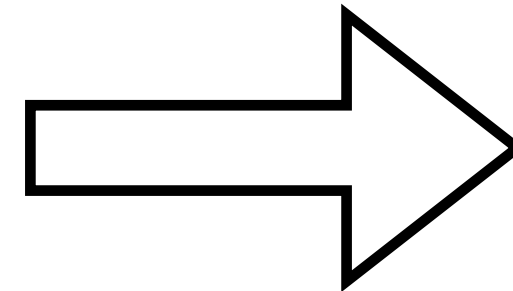
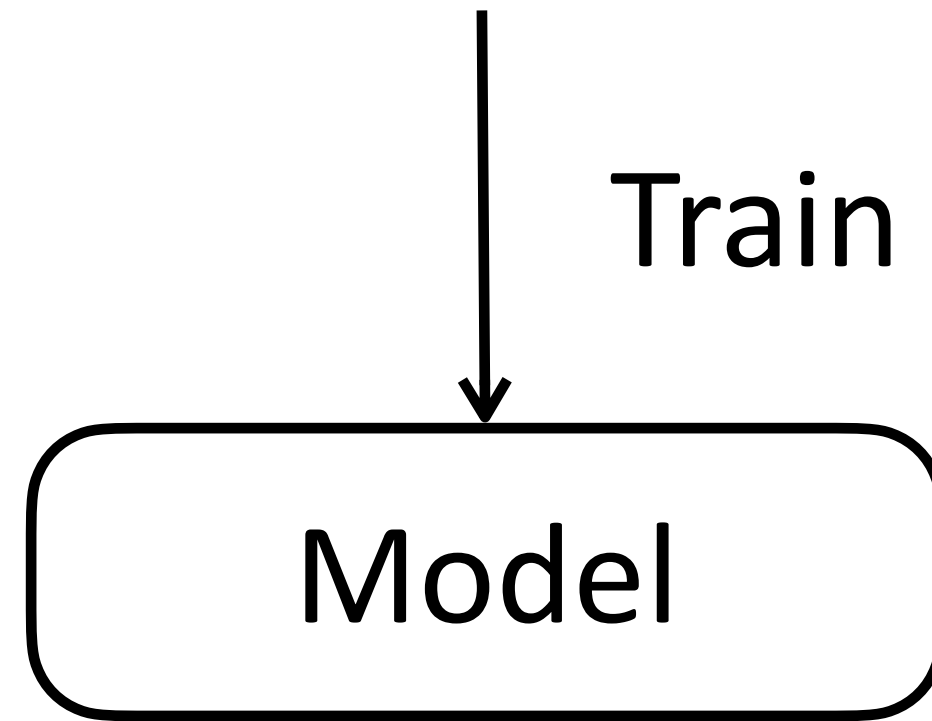
Large Language Models

Language Model Pretraining

Pretraining

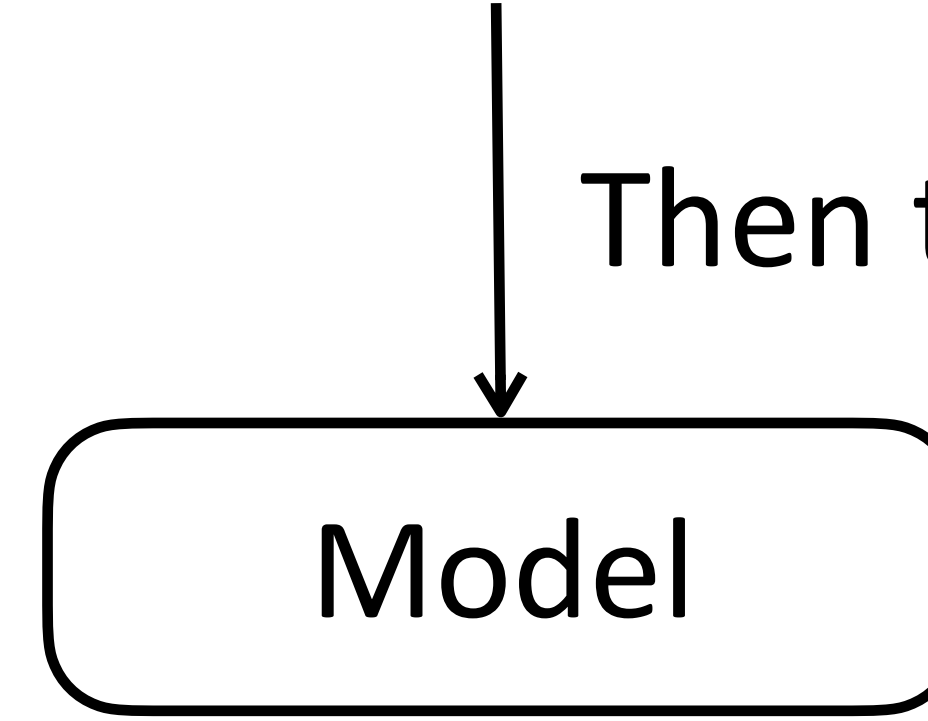
Source Data A (maybe a different task)

Train on data A first



Target Data B

Then train on data B



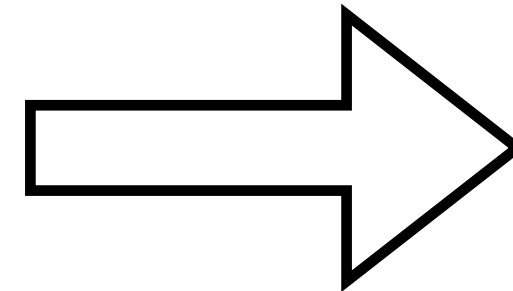
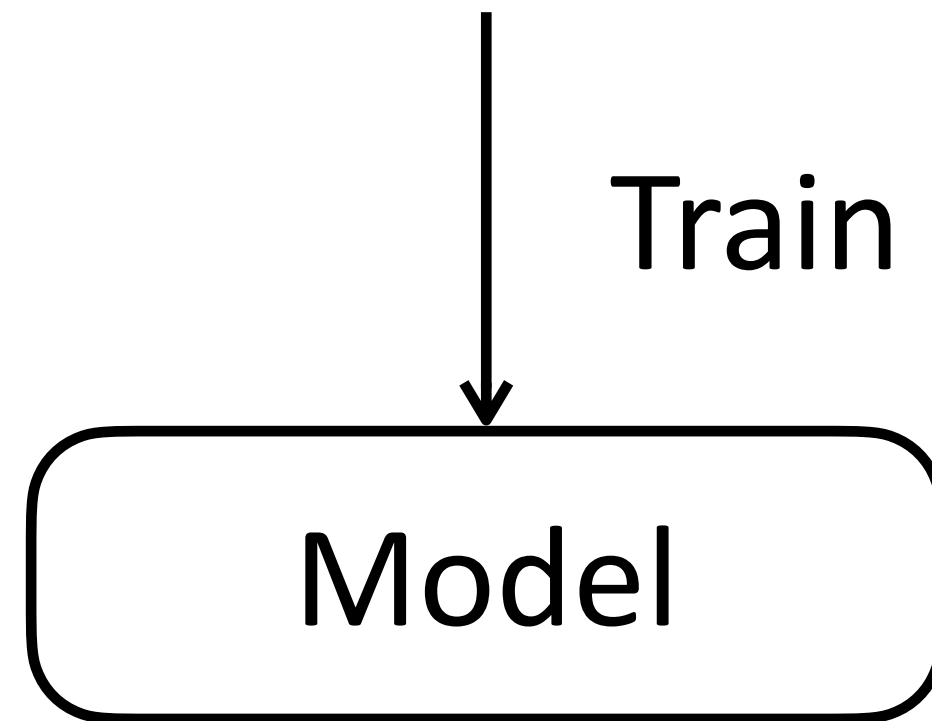
Classically, this is transfer Learning

It is now called pretraining because of the scale of A

Pretraining

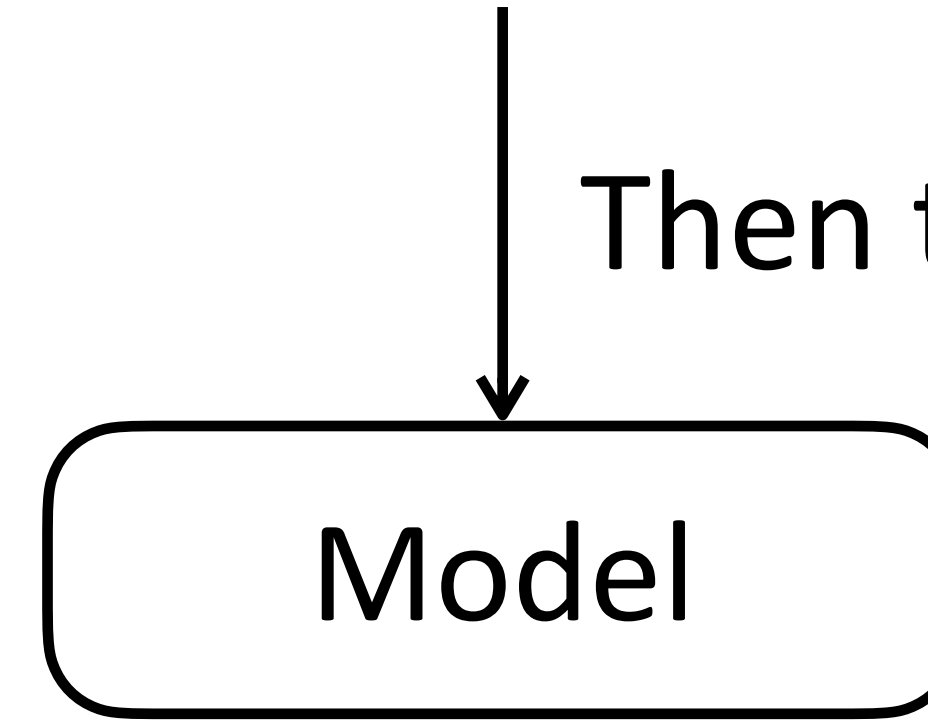
Source Data A (maybe a different task)

Train on data A first



Target Data B

Then train on data B

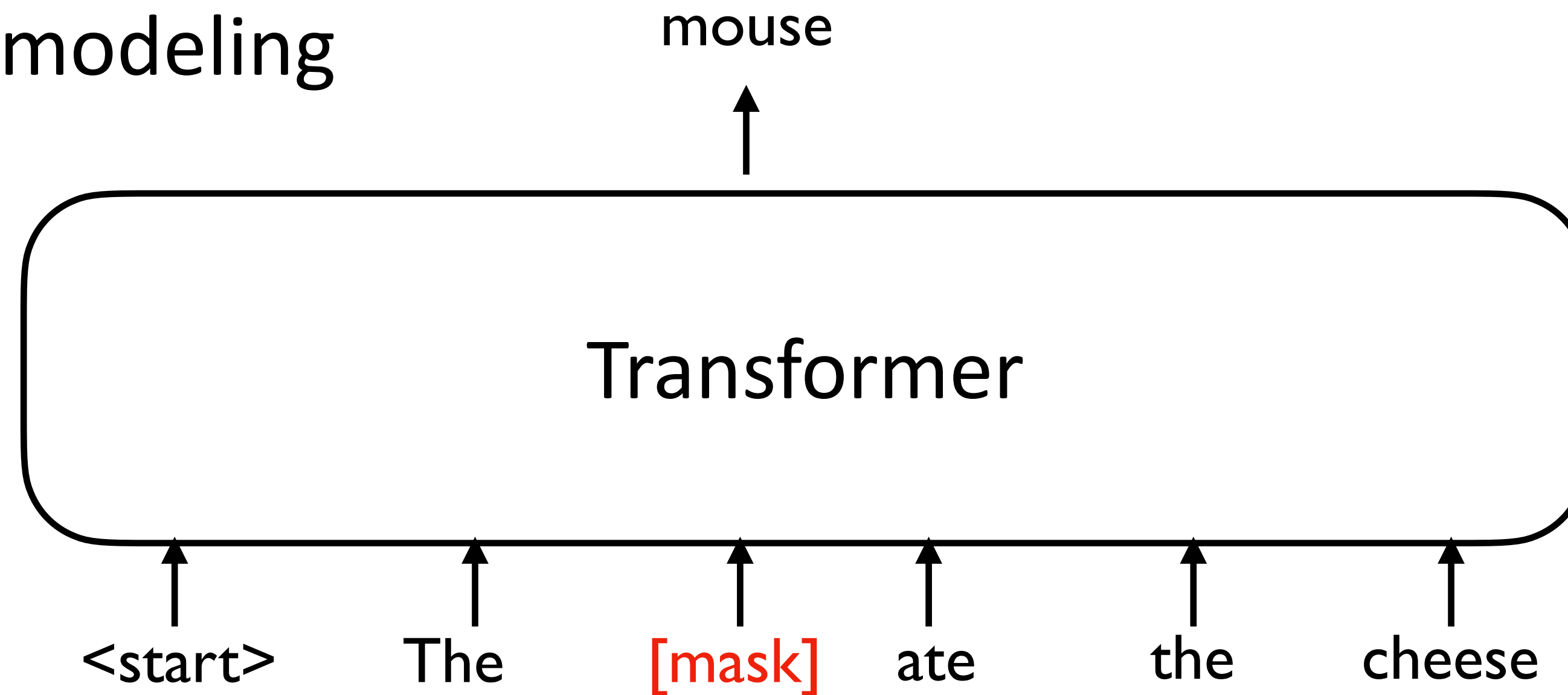


For supervised training, data A is often limited

How can we find large-scale data A to train?

BERT

Mask language modeling



Self-supervised Learning

Construct a synthetic task from raw text only

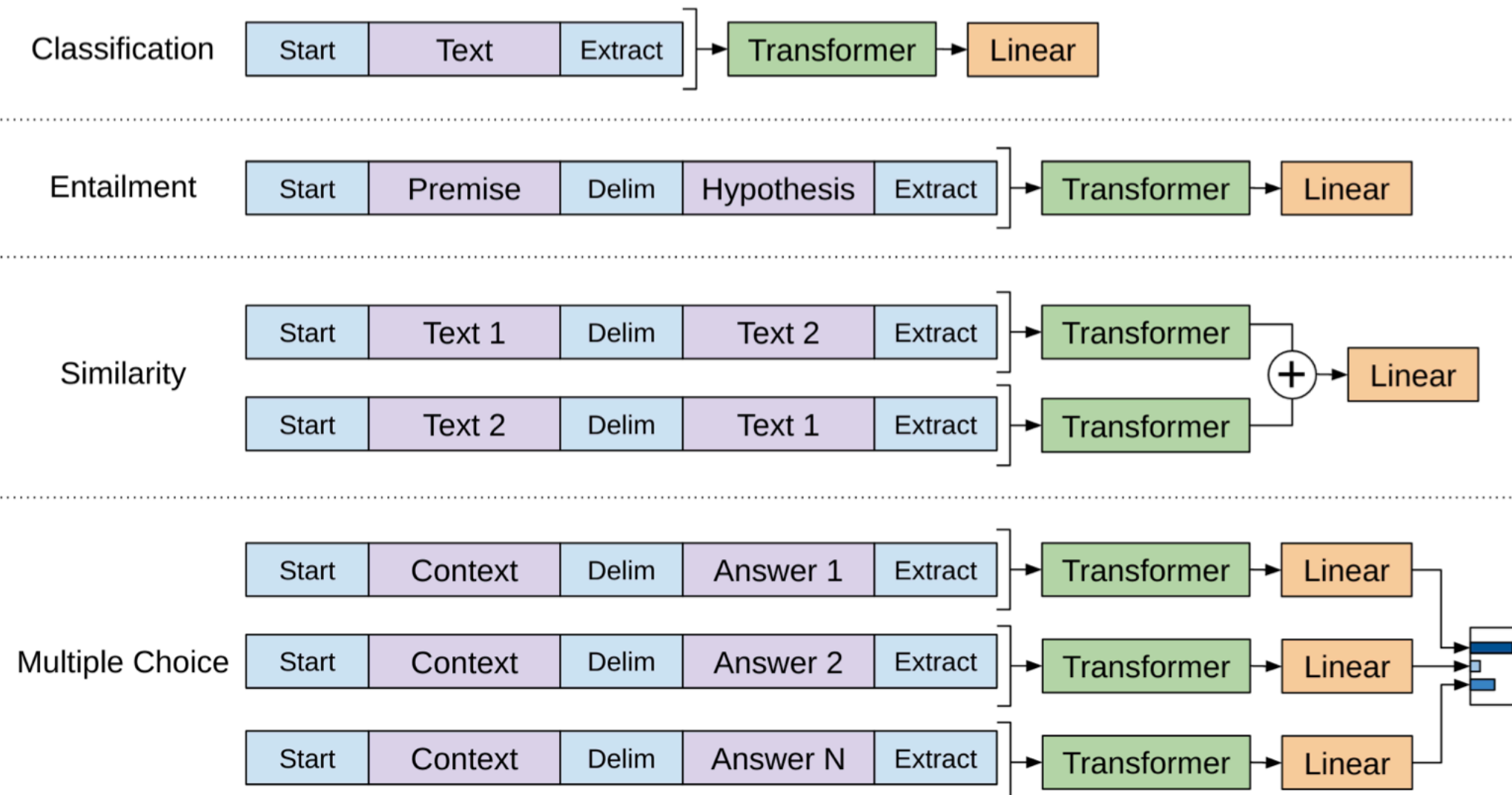
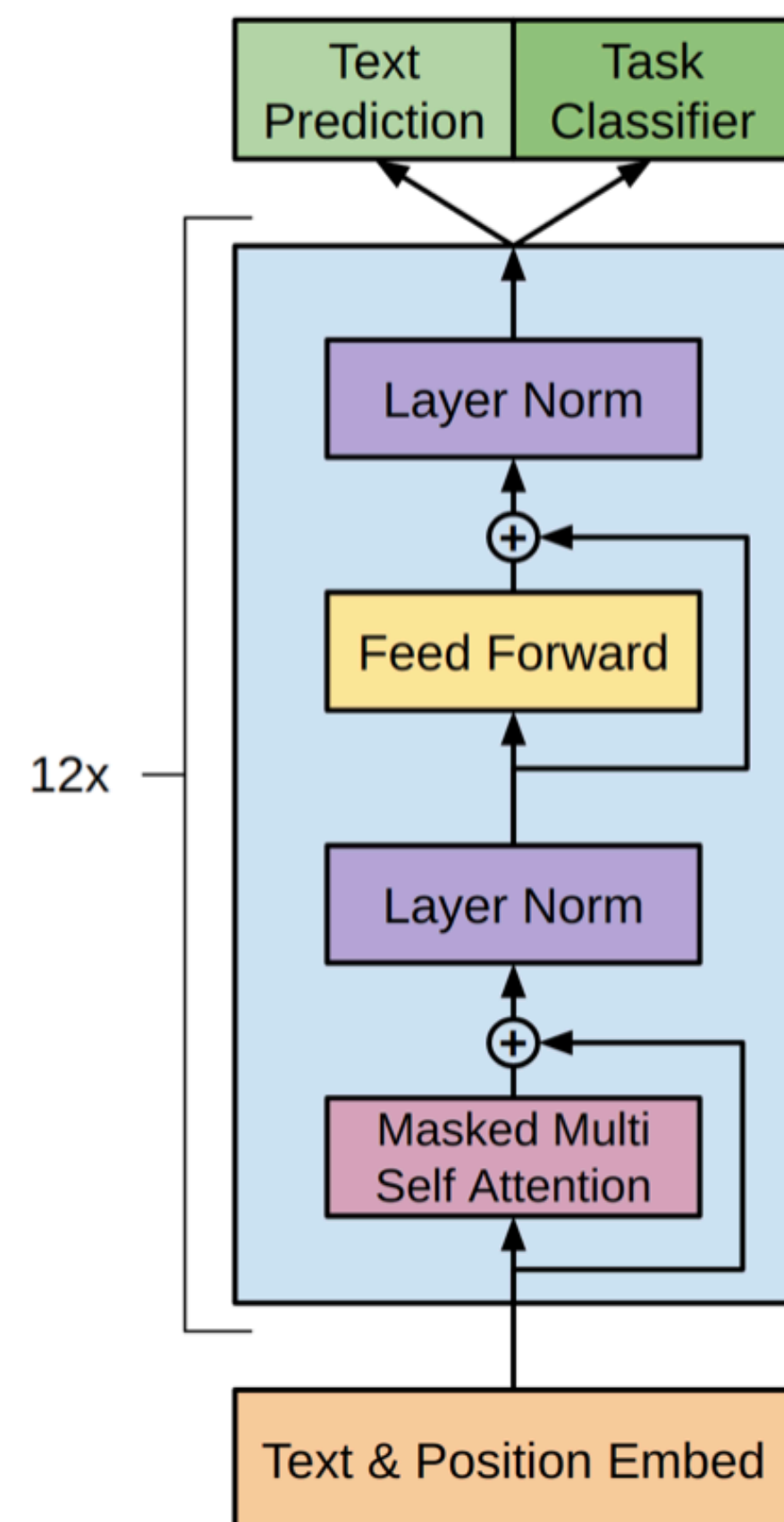
Can be made very large-scale

Is Bert a language model? Is it a generative model?

Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019.



Generative Pre-Training (GPT)



Radford et al. Improving Language Understanding by Generative Pre-Training. 2018

Thank You!