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

COMP 4901B

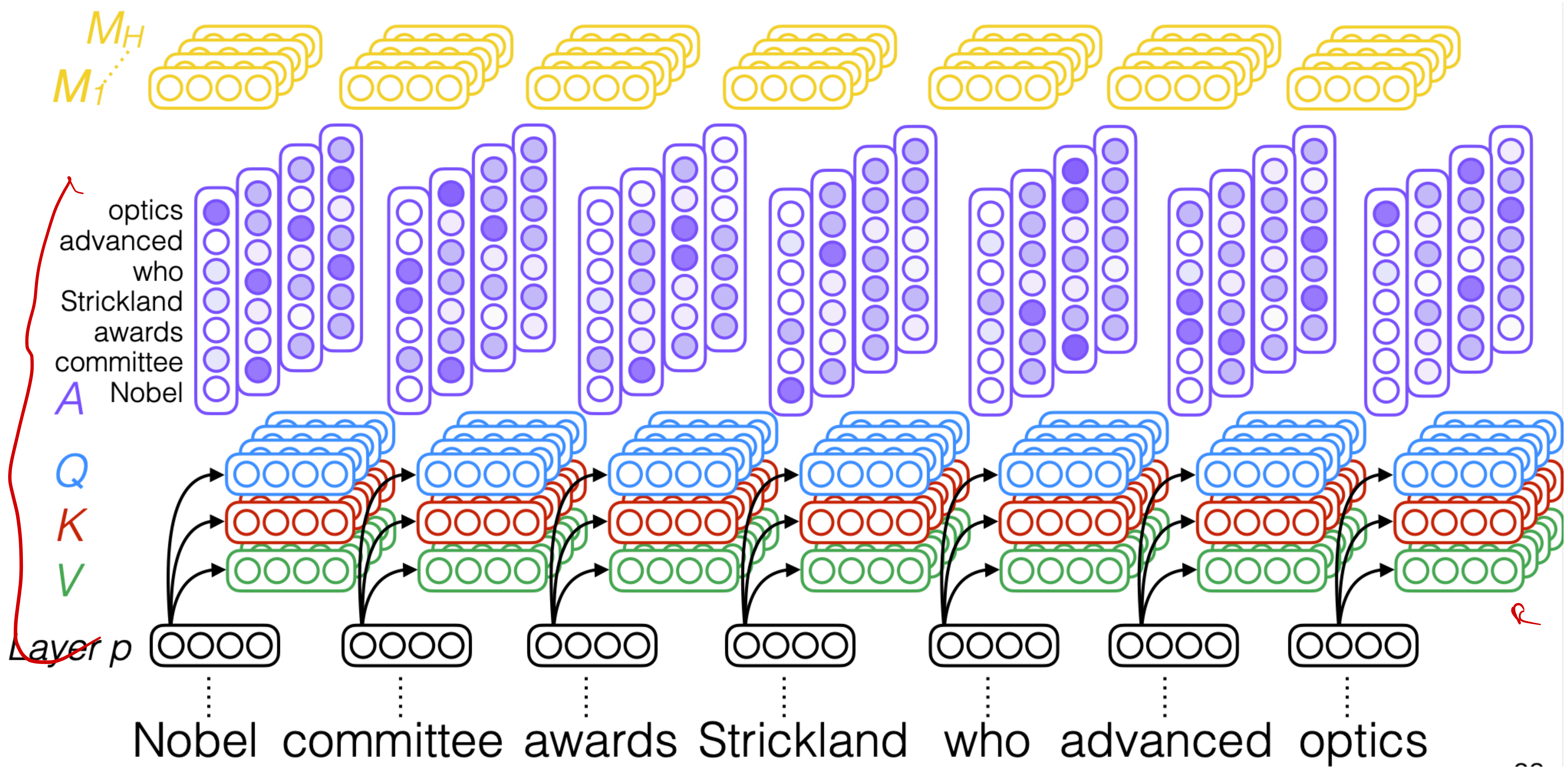
Large Language Models

Language Model Pretraining

Junxian He

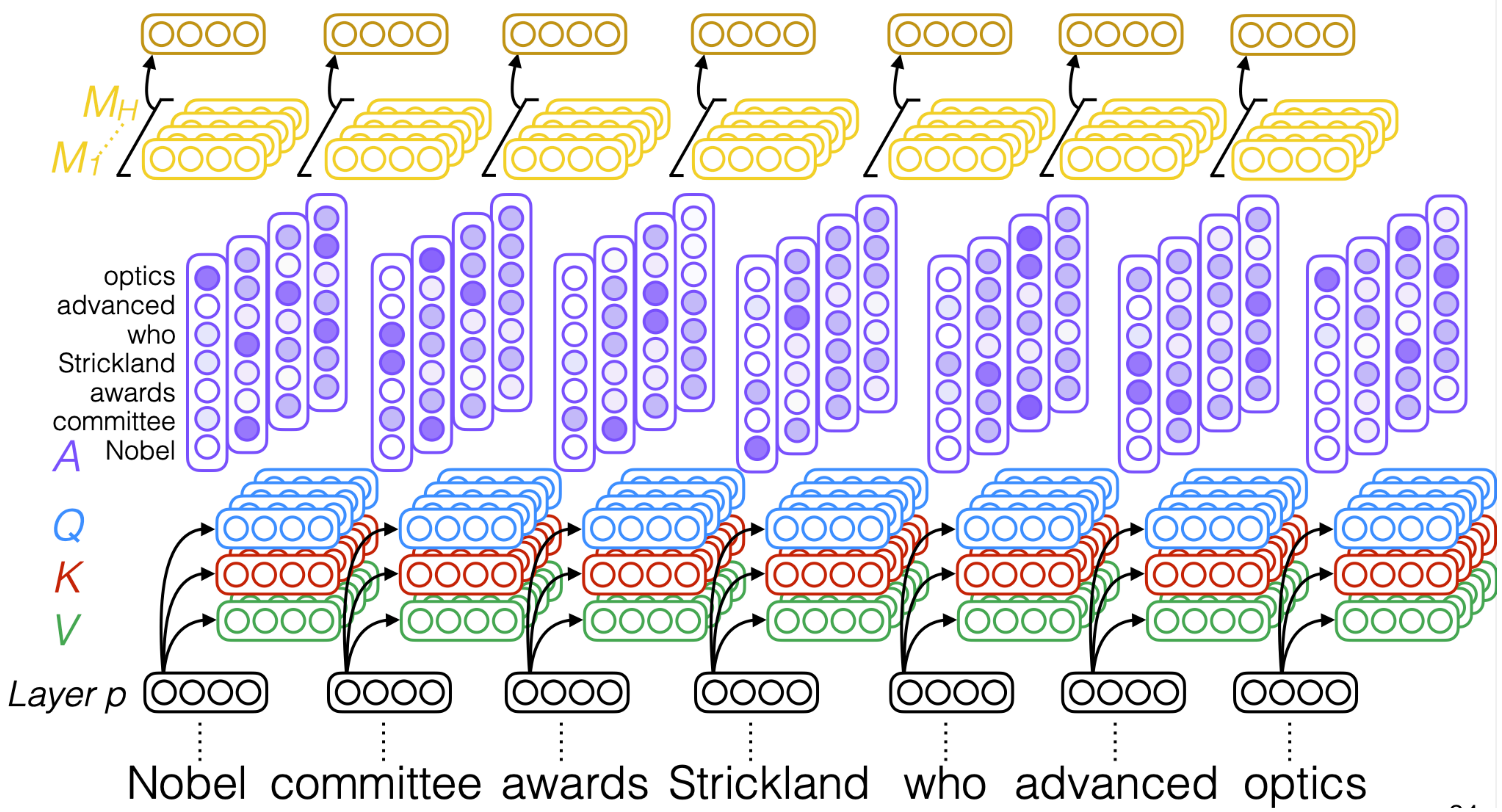
Sep 17, 2025

Recap: Multi-head Self-Attention



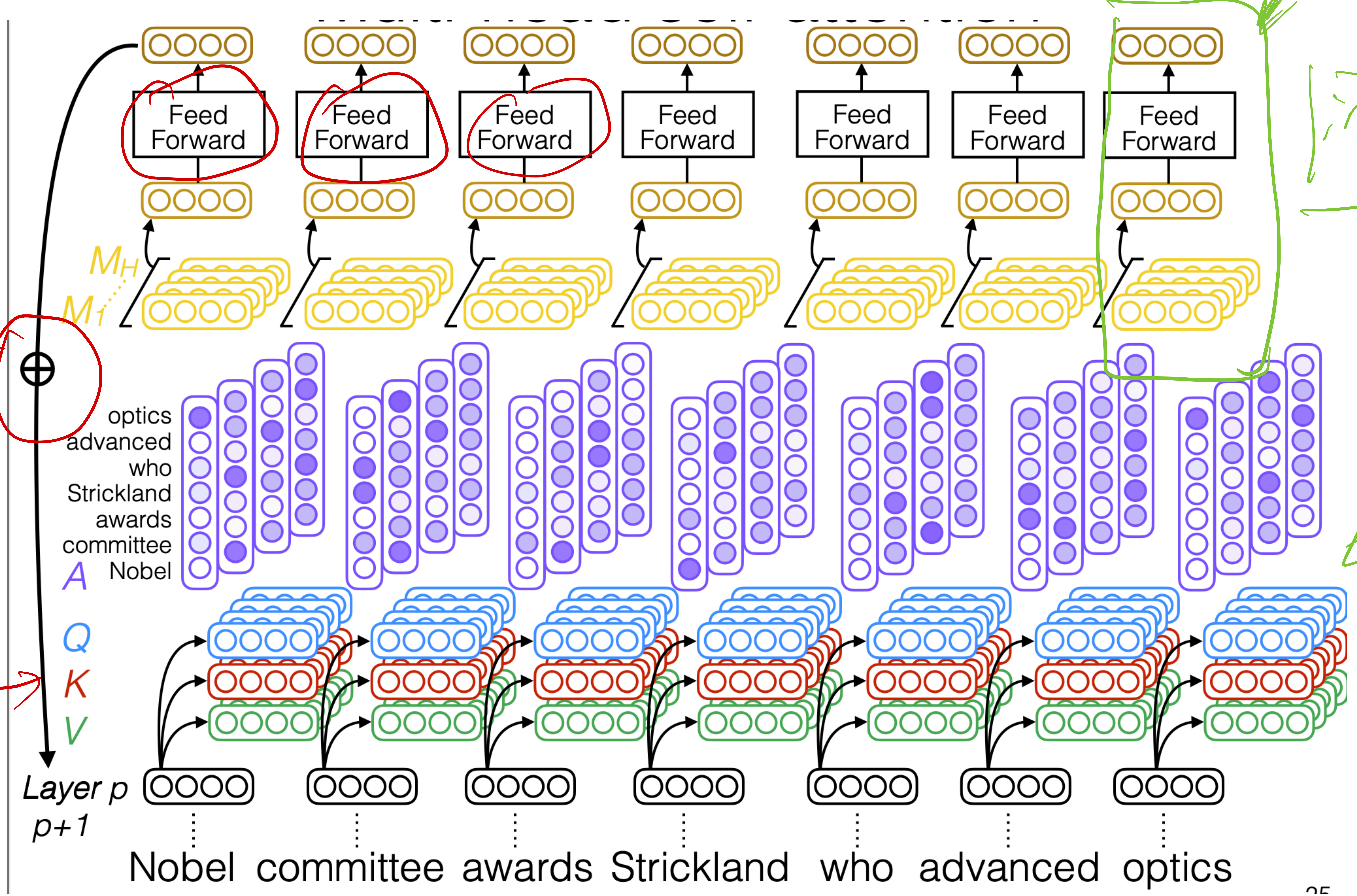
Recap: Multi-head Self-Attention

Concat and output projection



parameter sharing

Recap: Multi-head Self-Attention + FFN



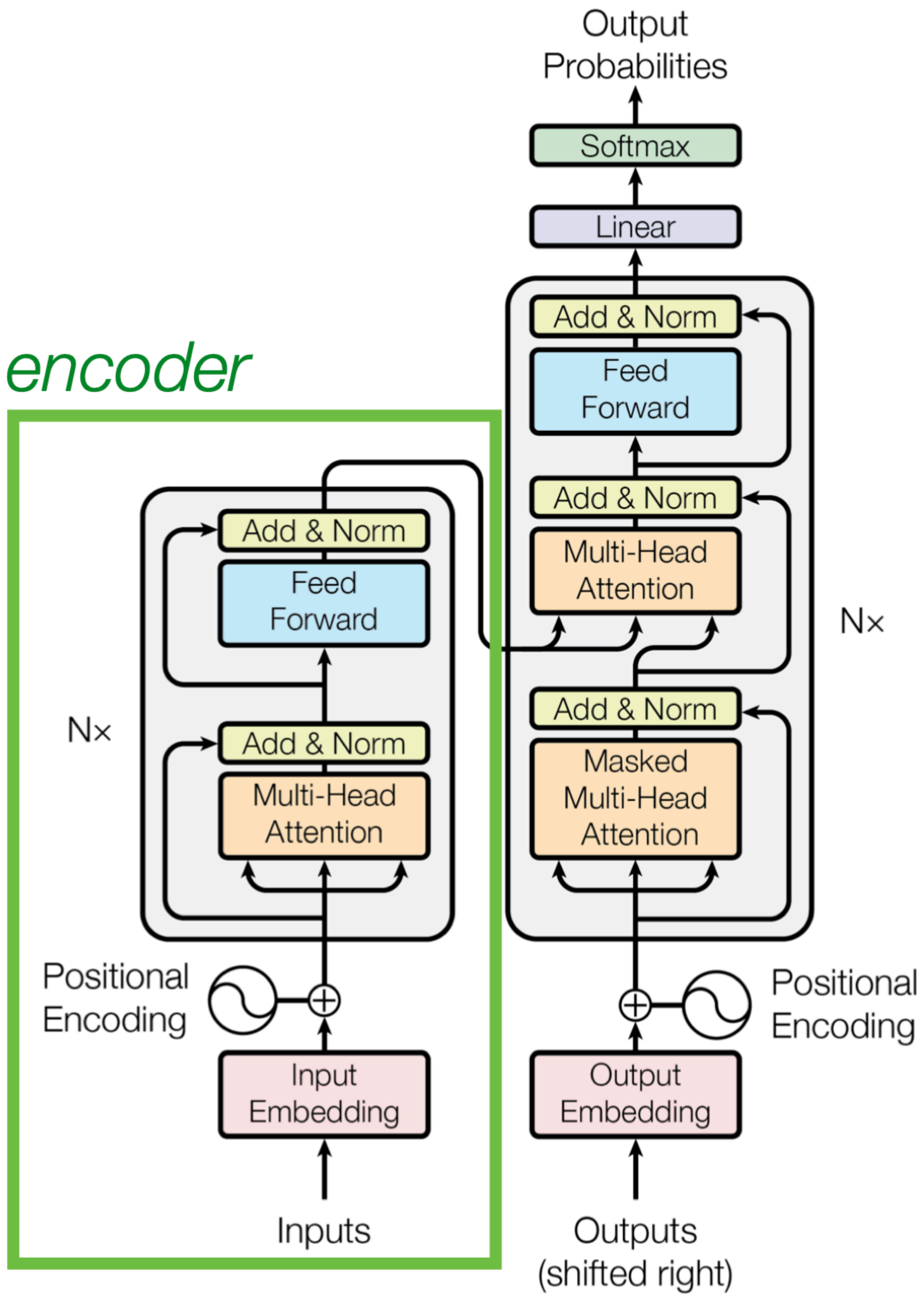
independently

q·k

← dependency between steps

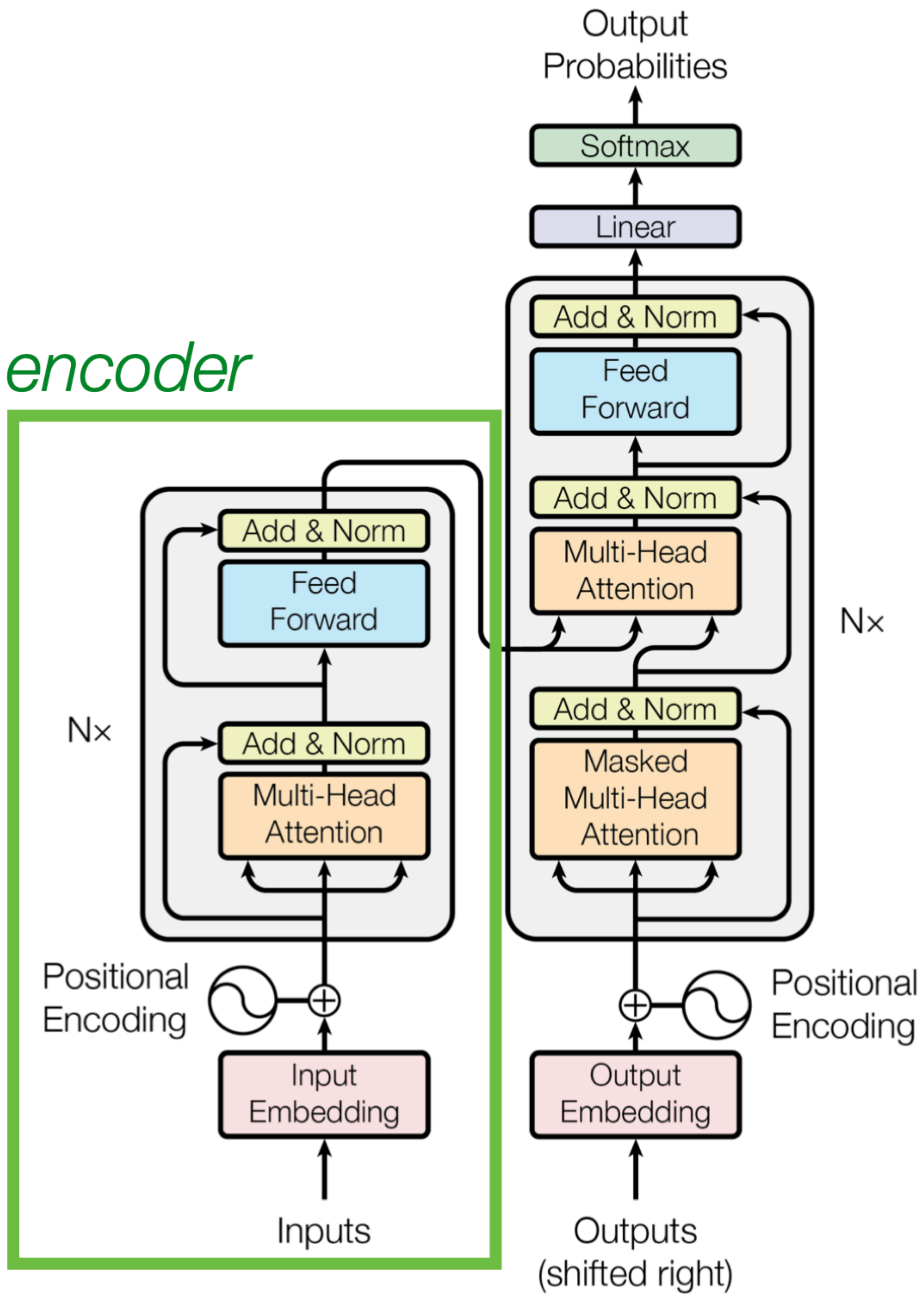
W^q
W^k
W^v

Recap: Transformer Encoder

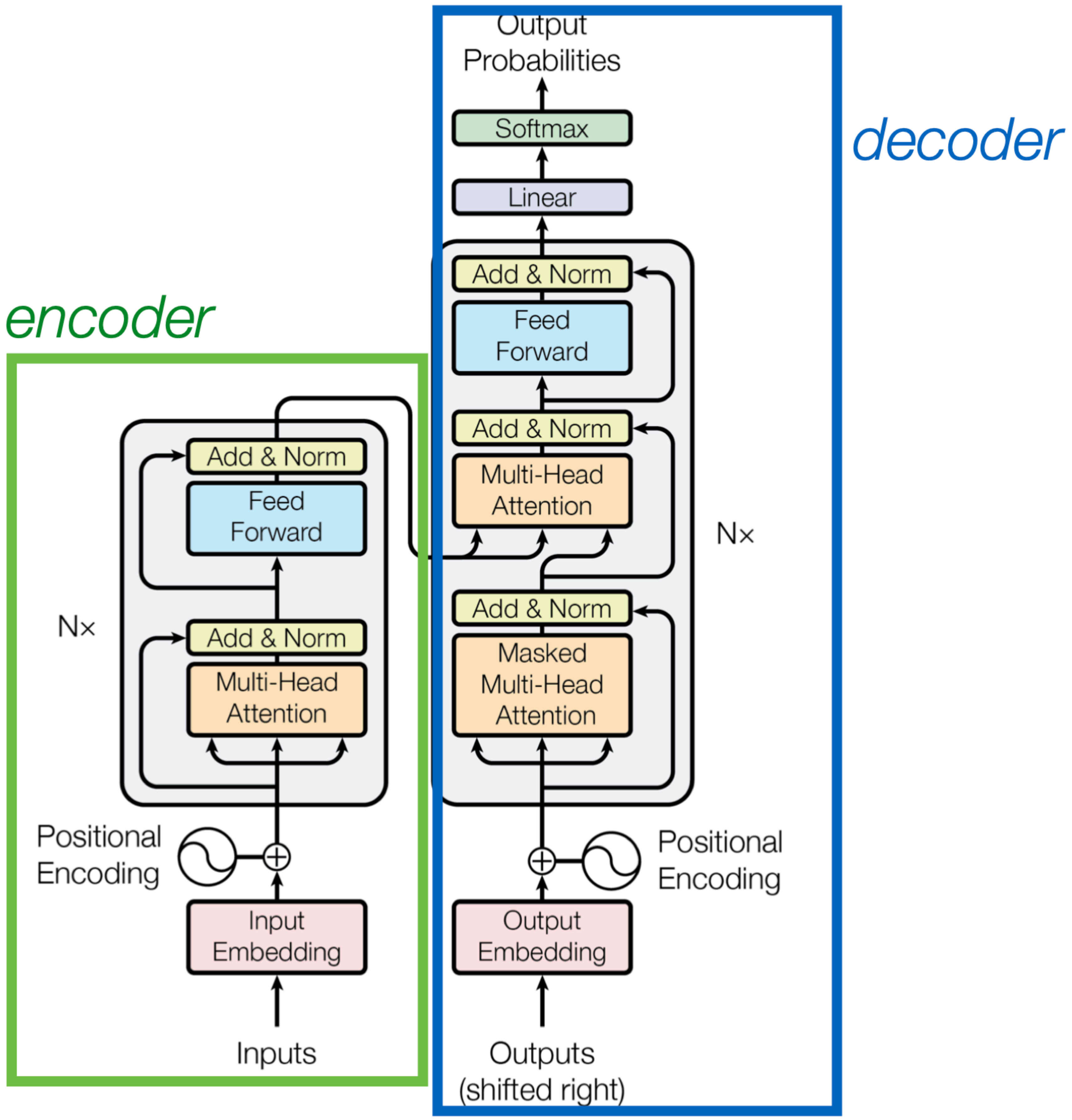


Currently we only cover the encoder side

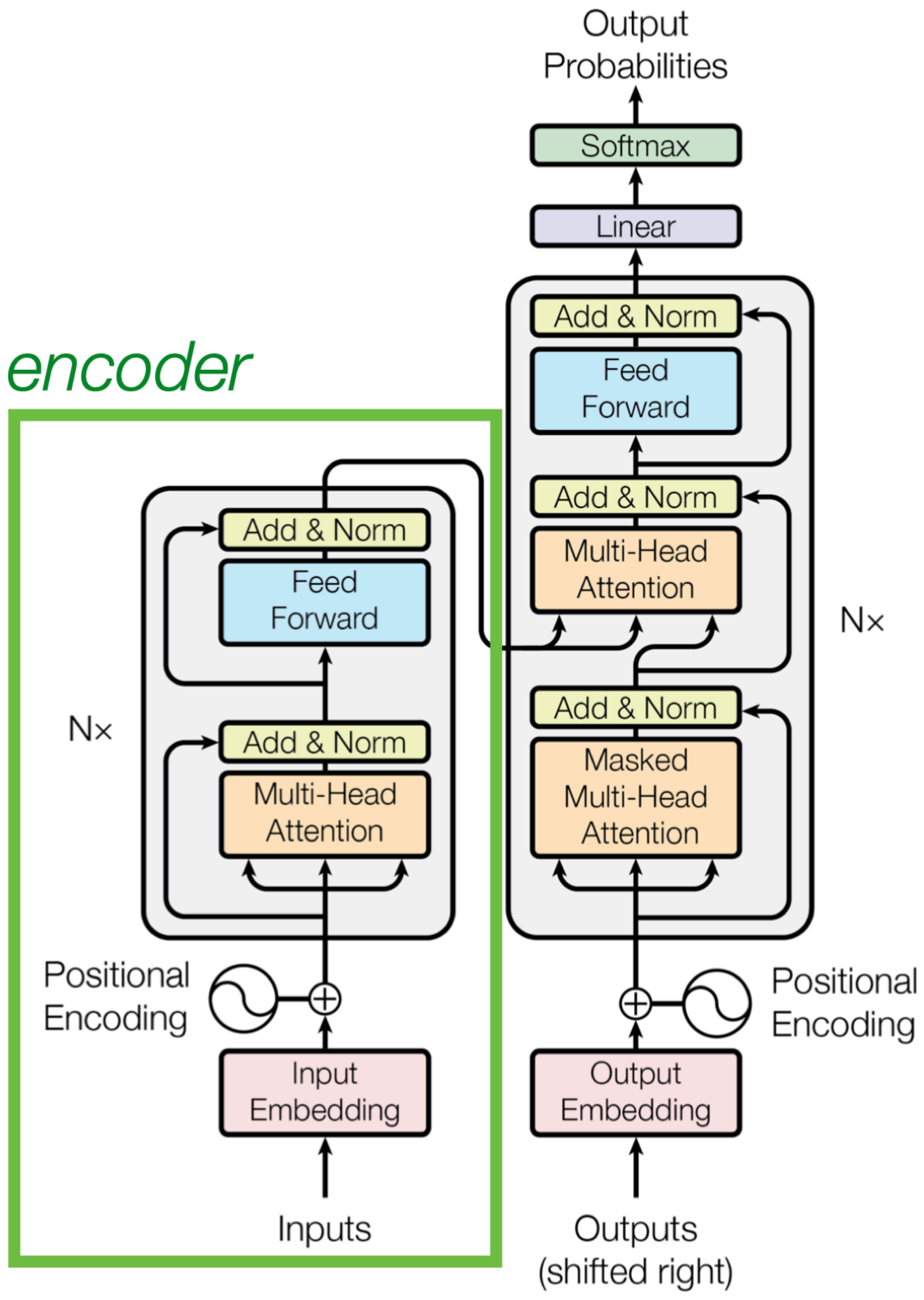
Recap: Transformer Encoder



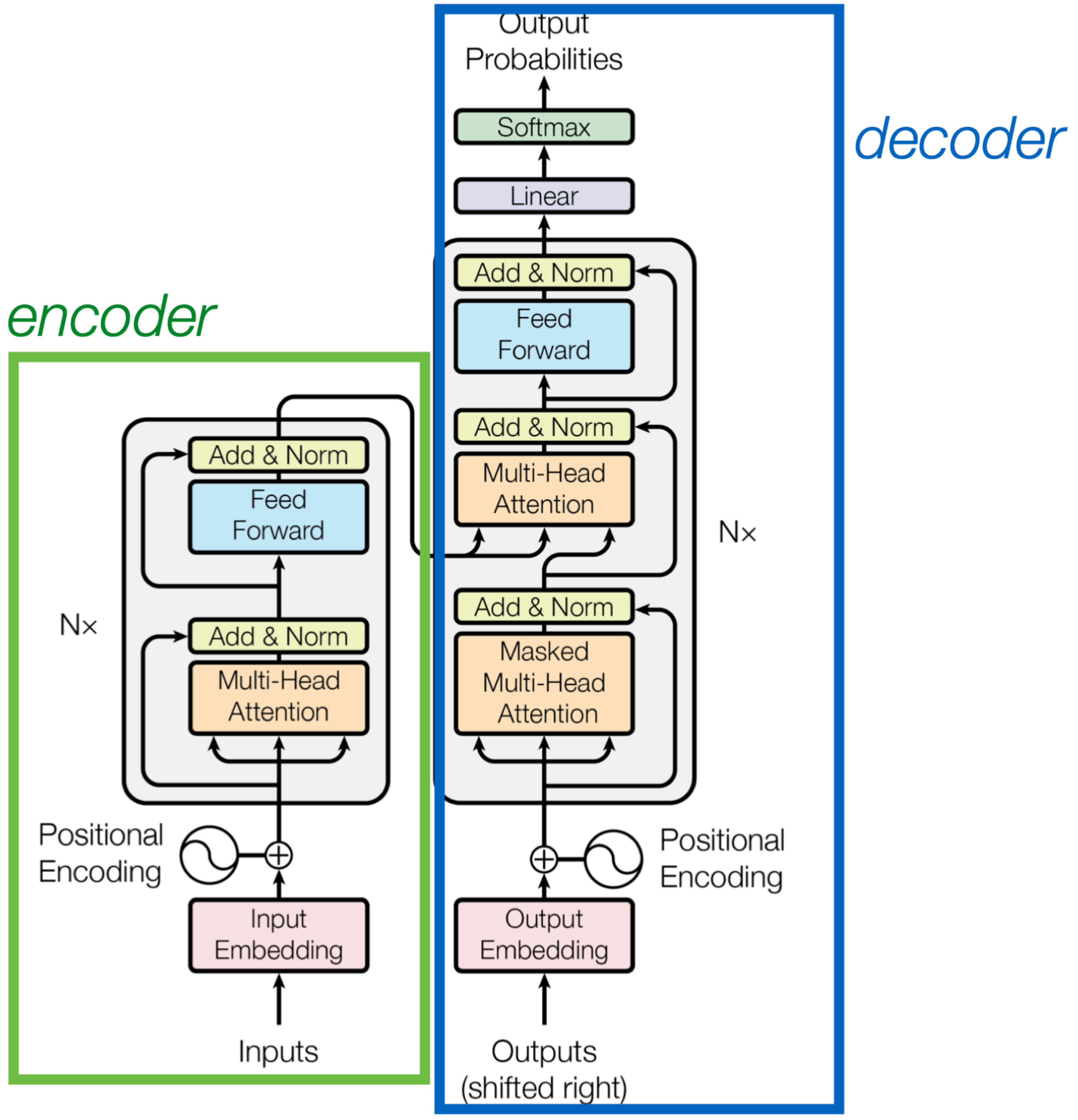
Currently we only cover the encoder side



Recap: Transformer Encoder

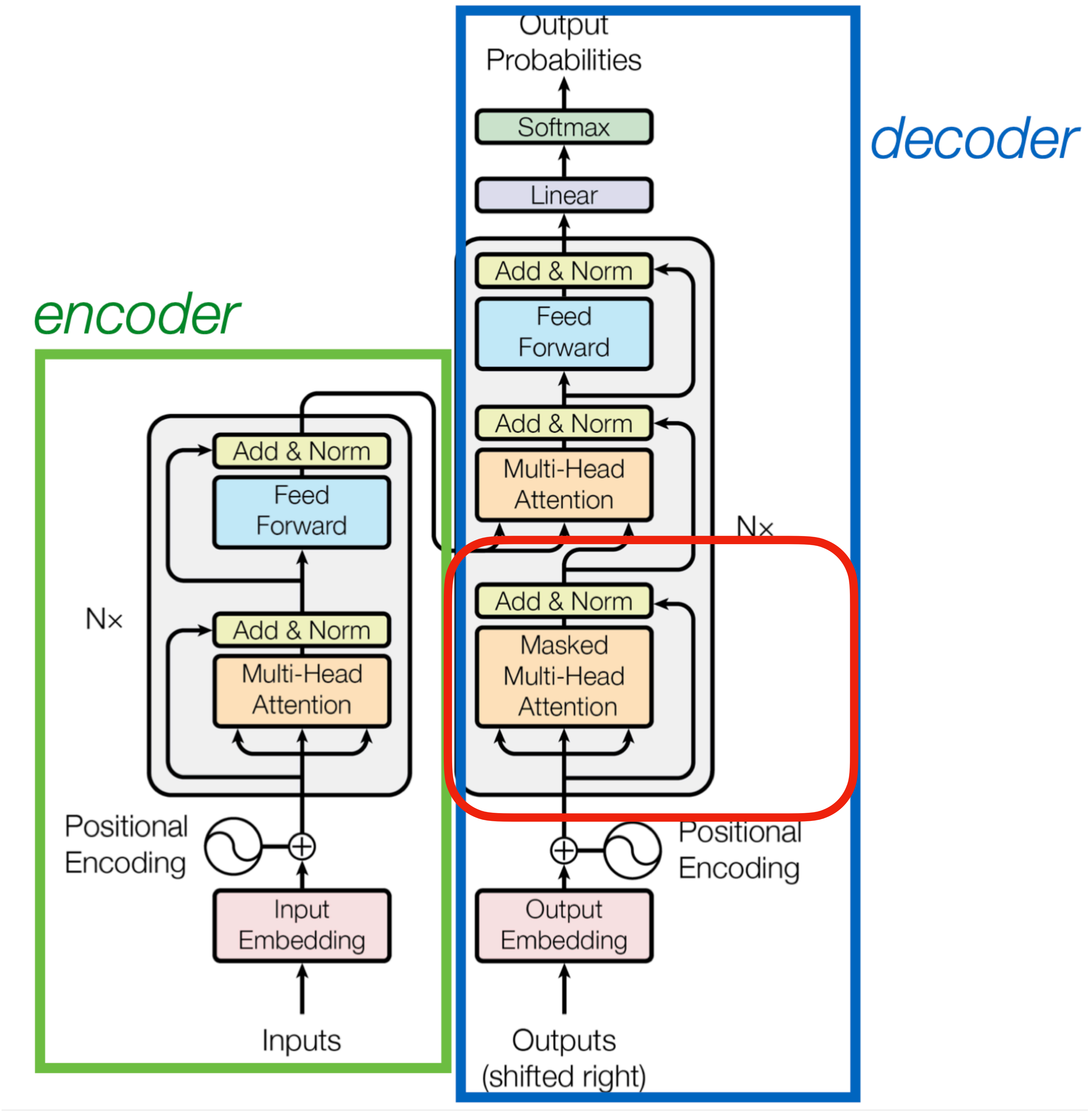


Currently we only cover the encoder side

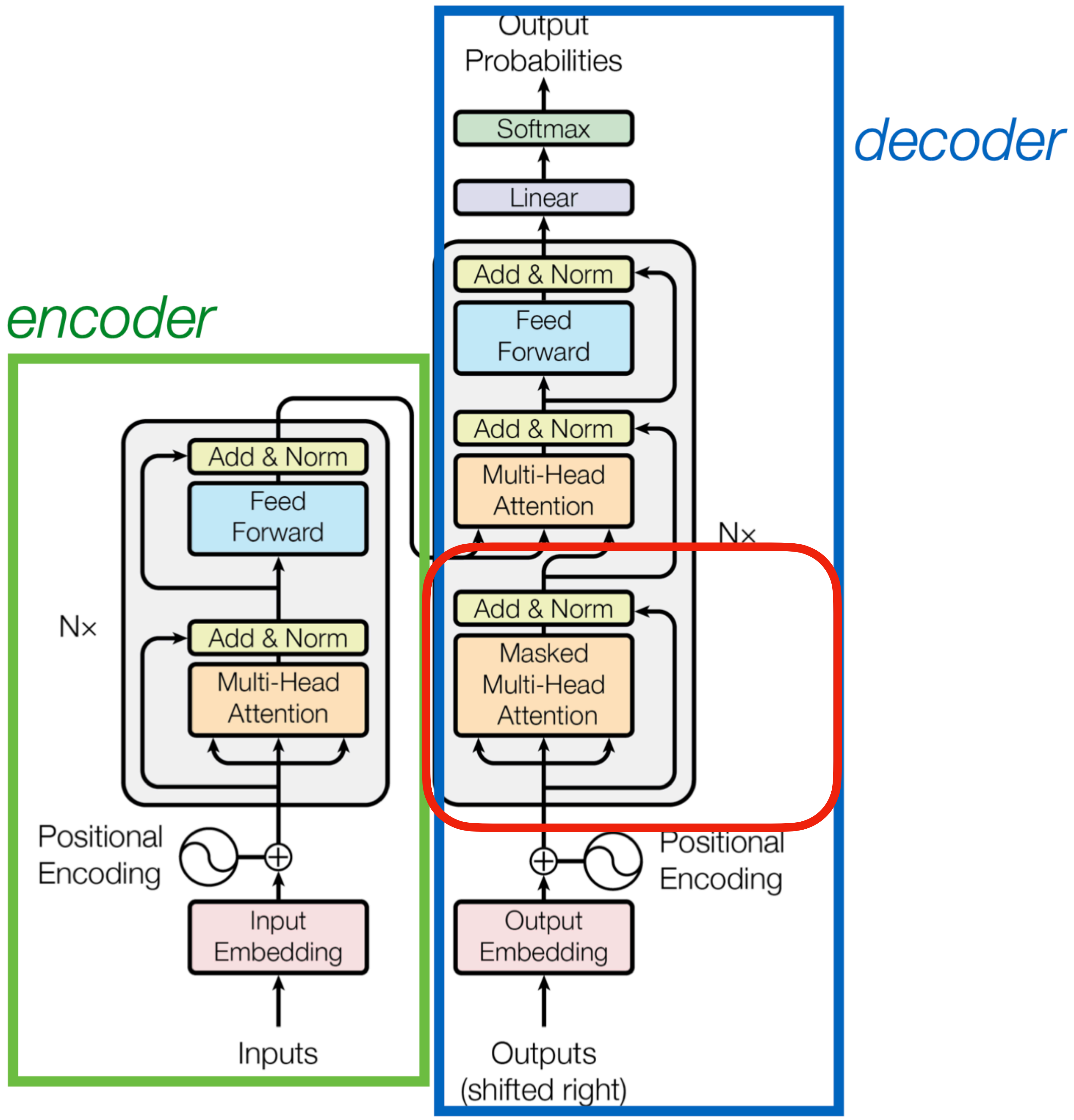


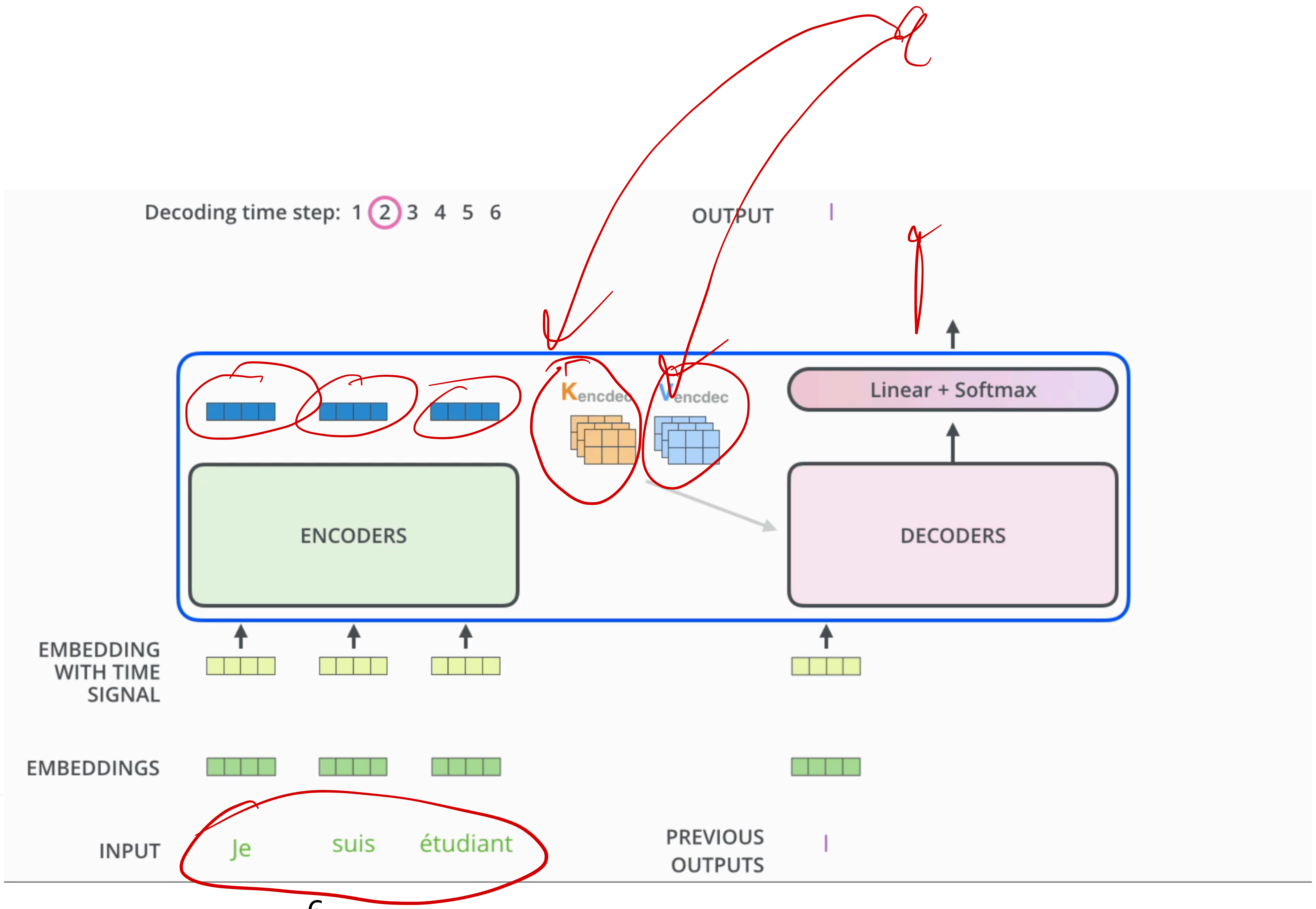
This encoder-decoder arch is originally proposed as a seq2seq arch, for classification tasks, often only encoder is used. And language models often only have a decoder

Recap: Masked Attention



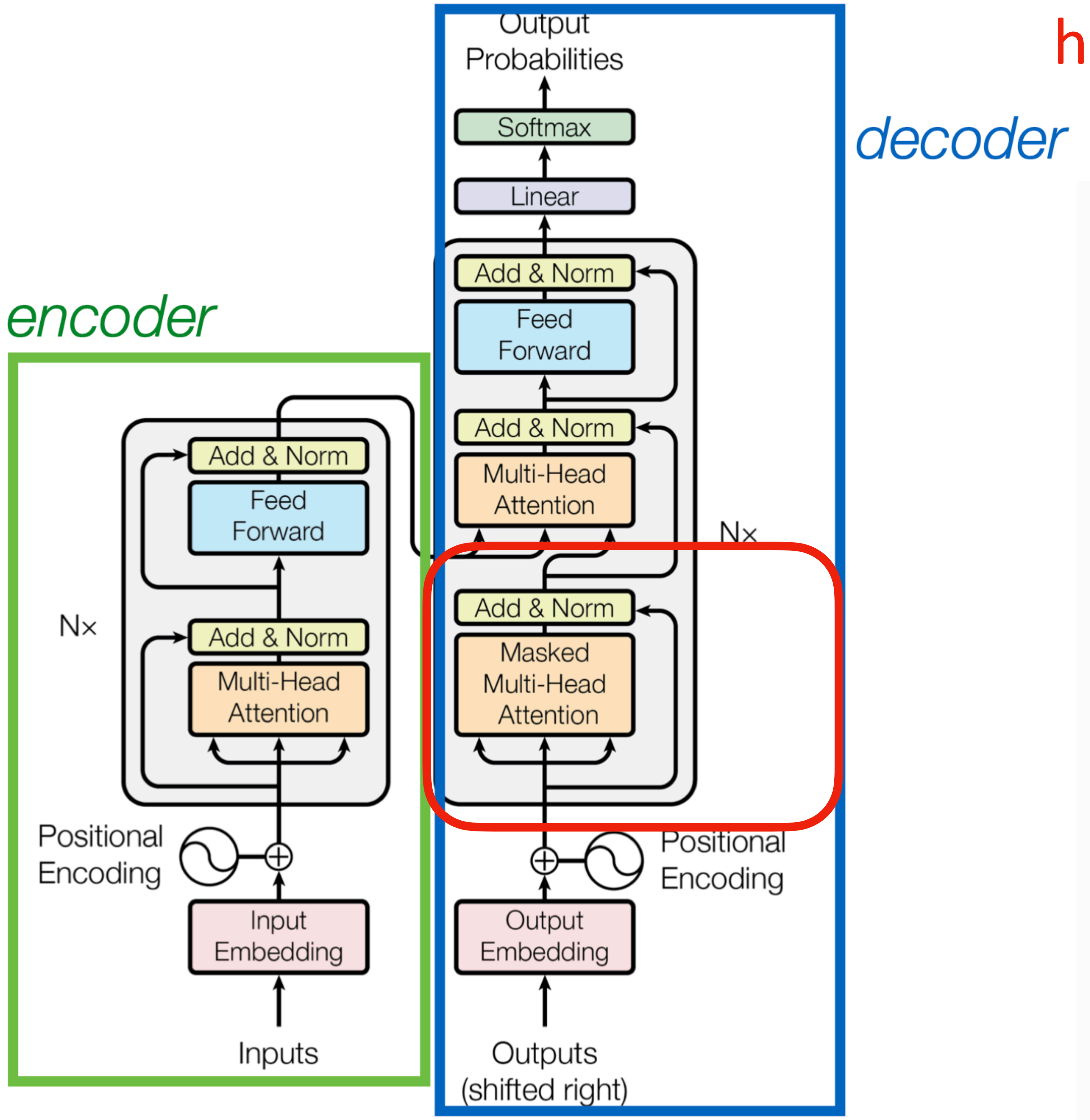
Recap: Masked Attention

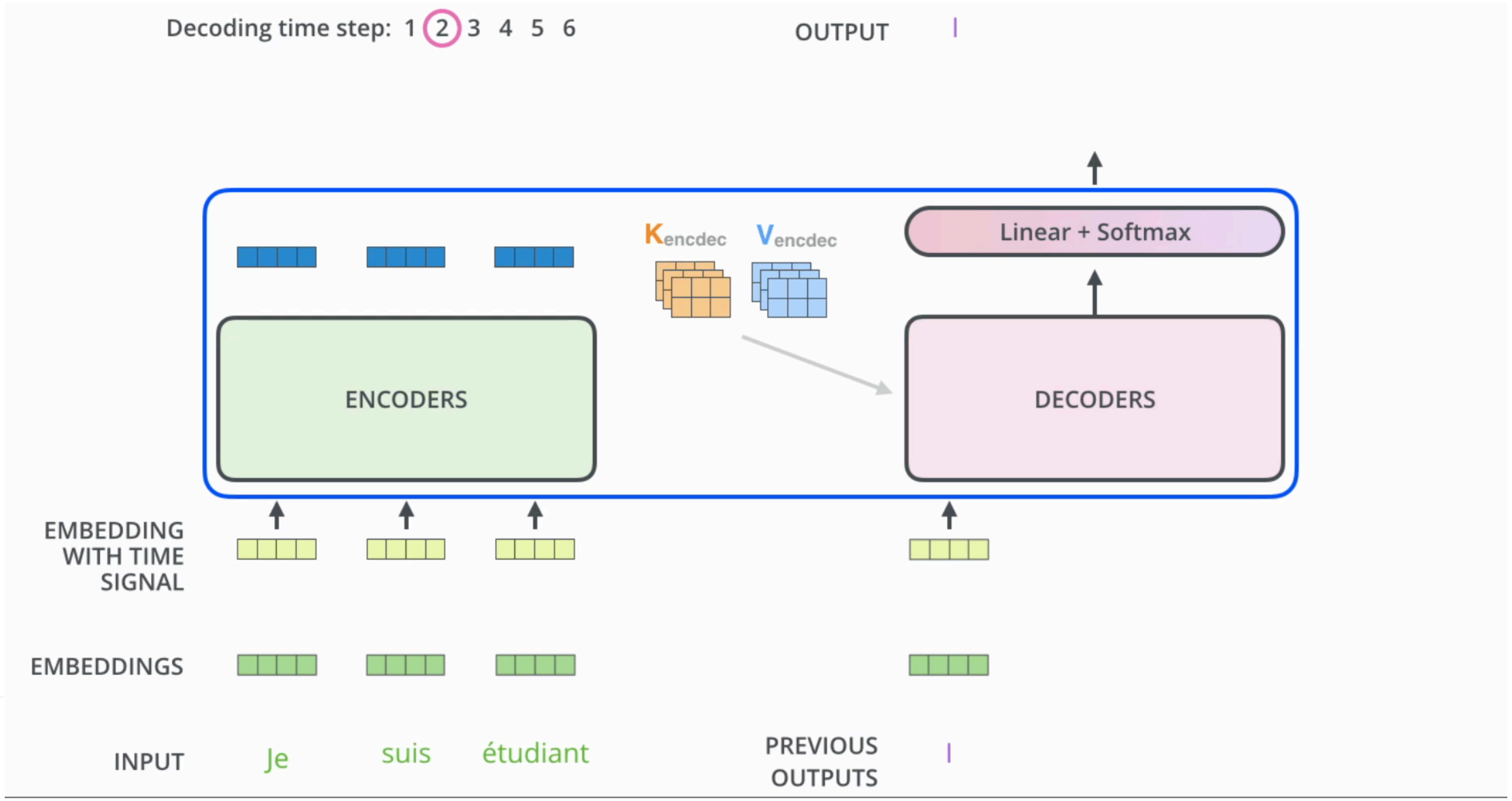




Recap: Masked Attention

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



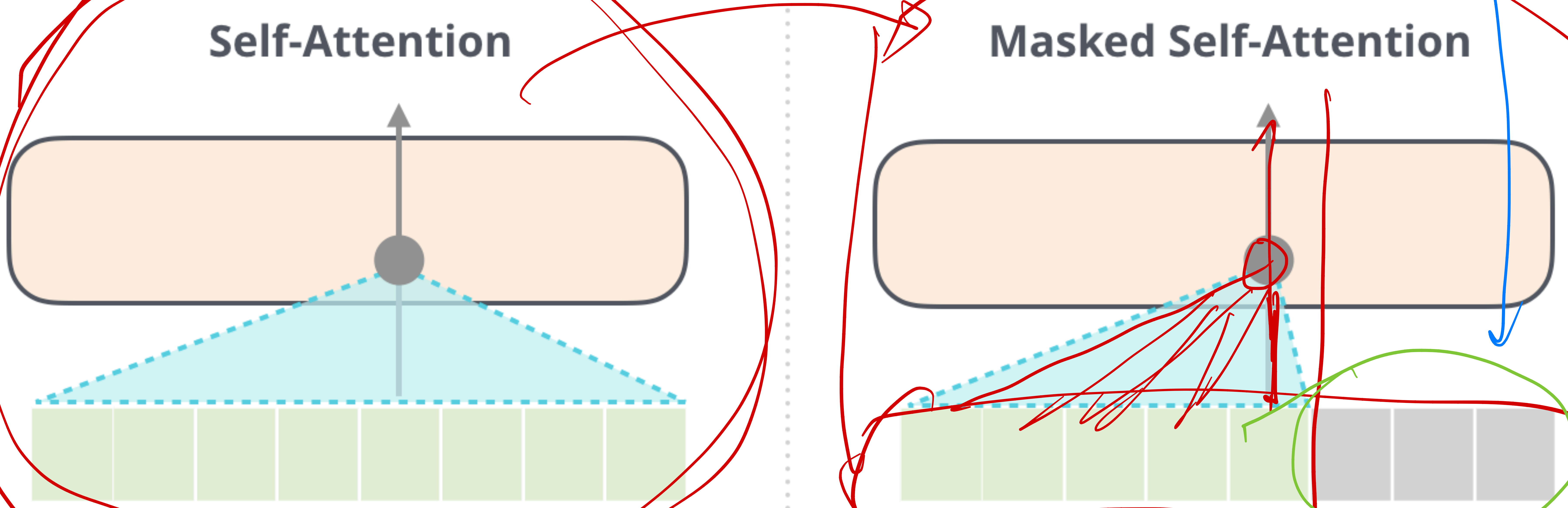


Masked Attention

Σ att-weight · Value

attention weight 0

bidirectional



Σ atten weight $\neq 1$

atten weight

$\uparrow$ softmax

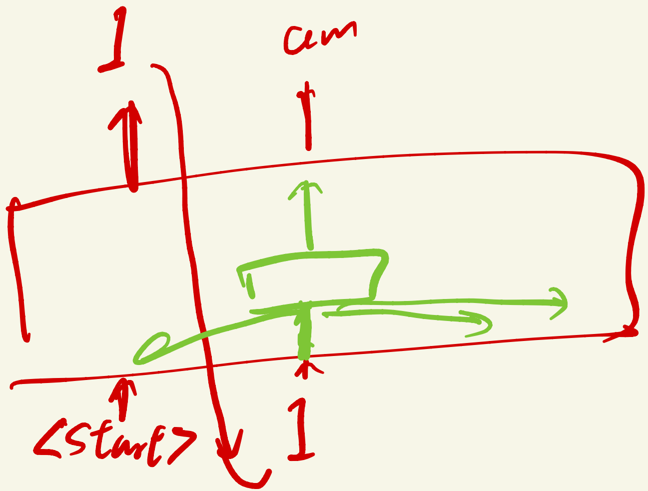
$q \cdot k$

$q \cdot k \longrightarrow -\infty$

-10000

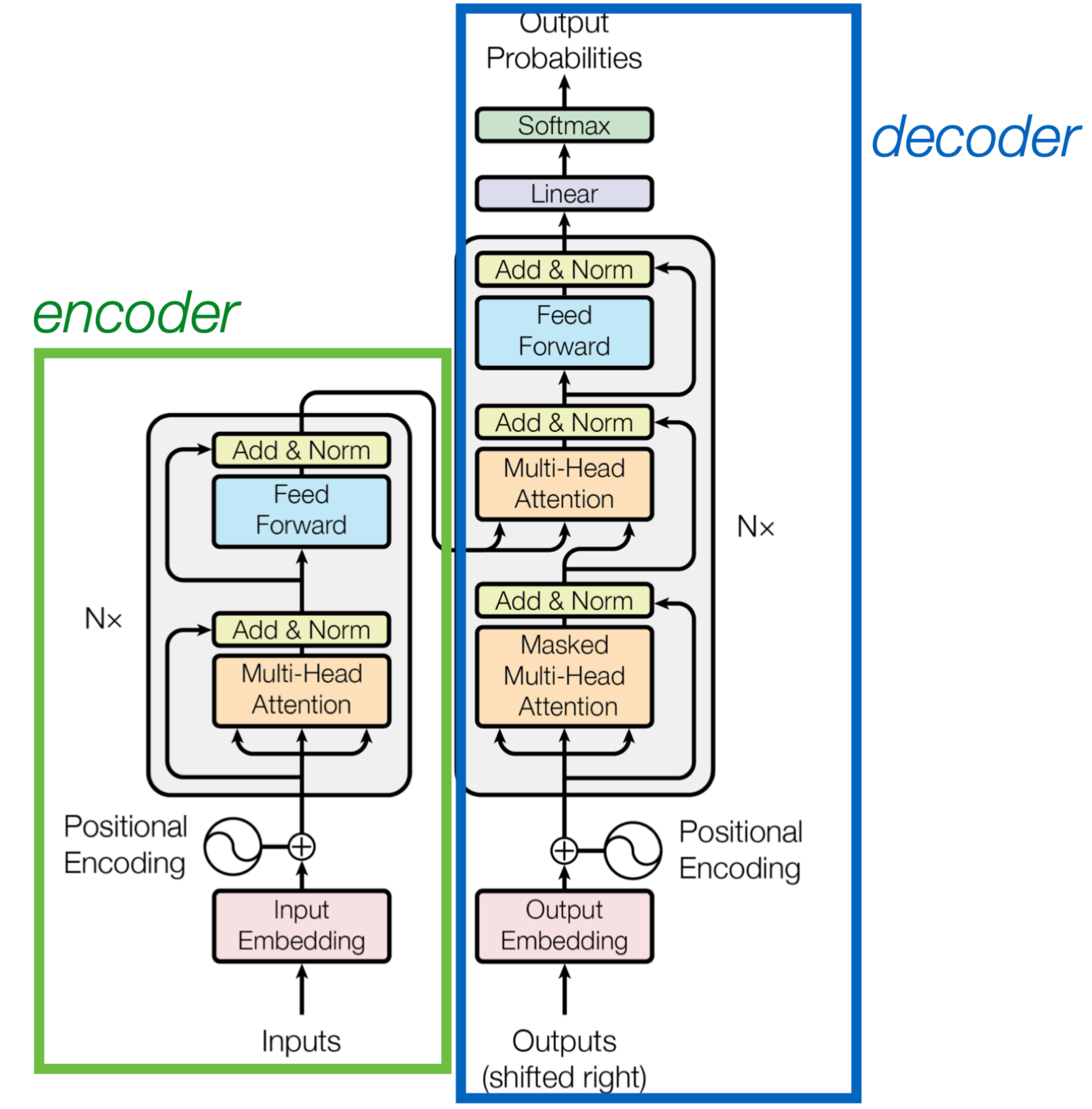
softmax

≈ 0

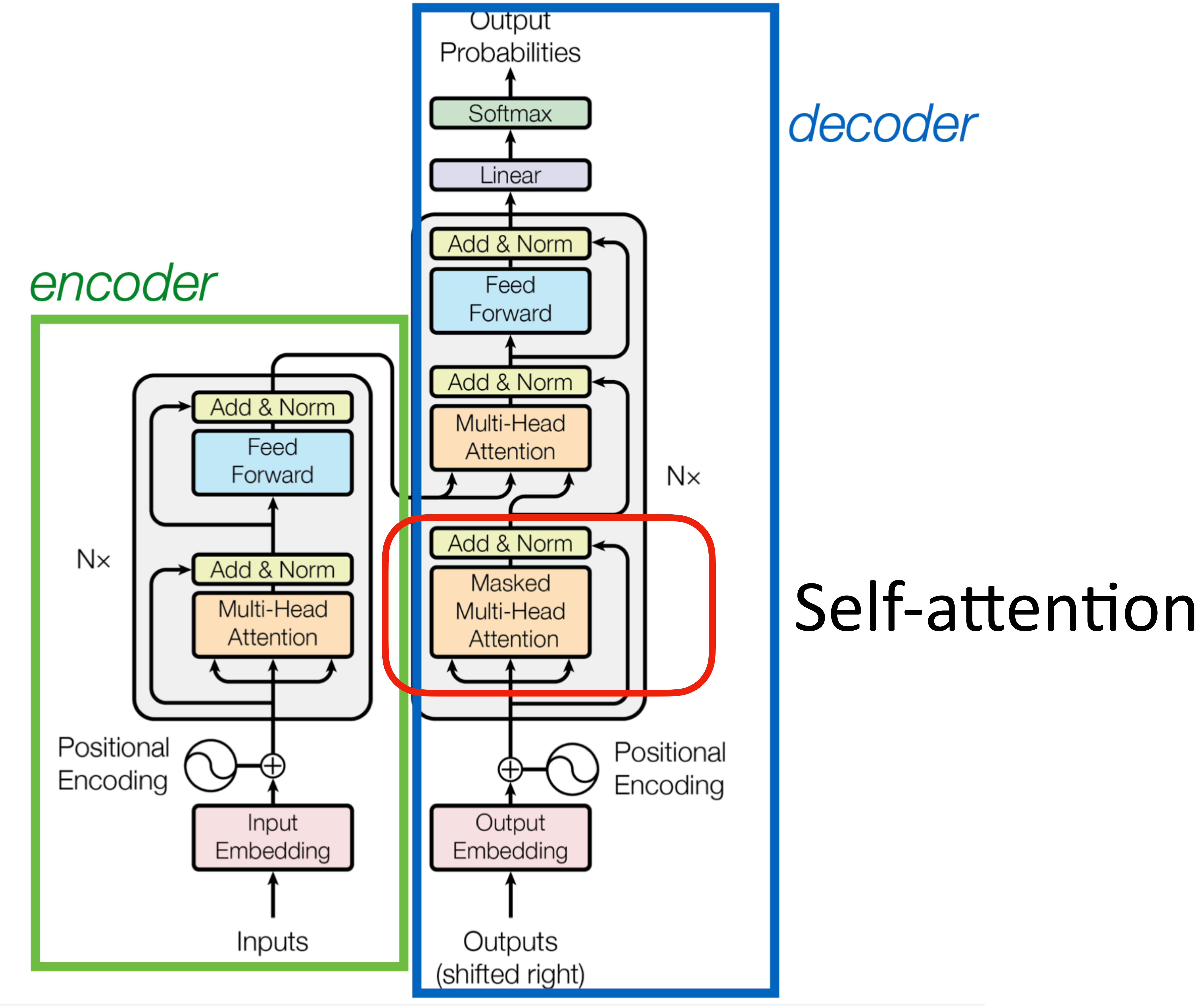


I am having a class

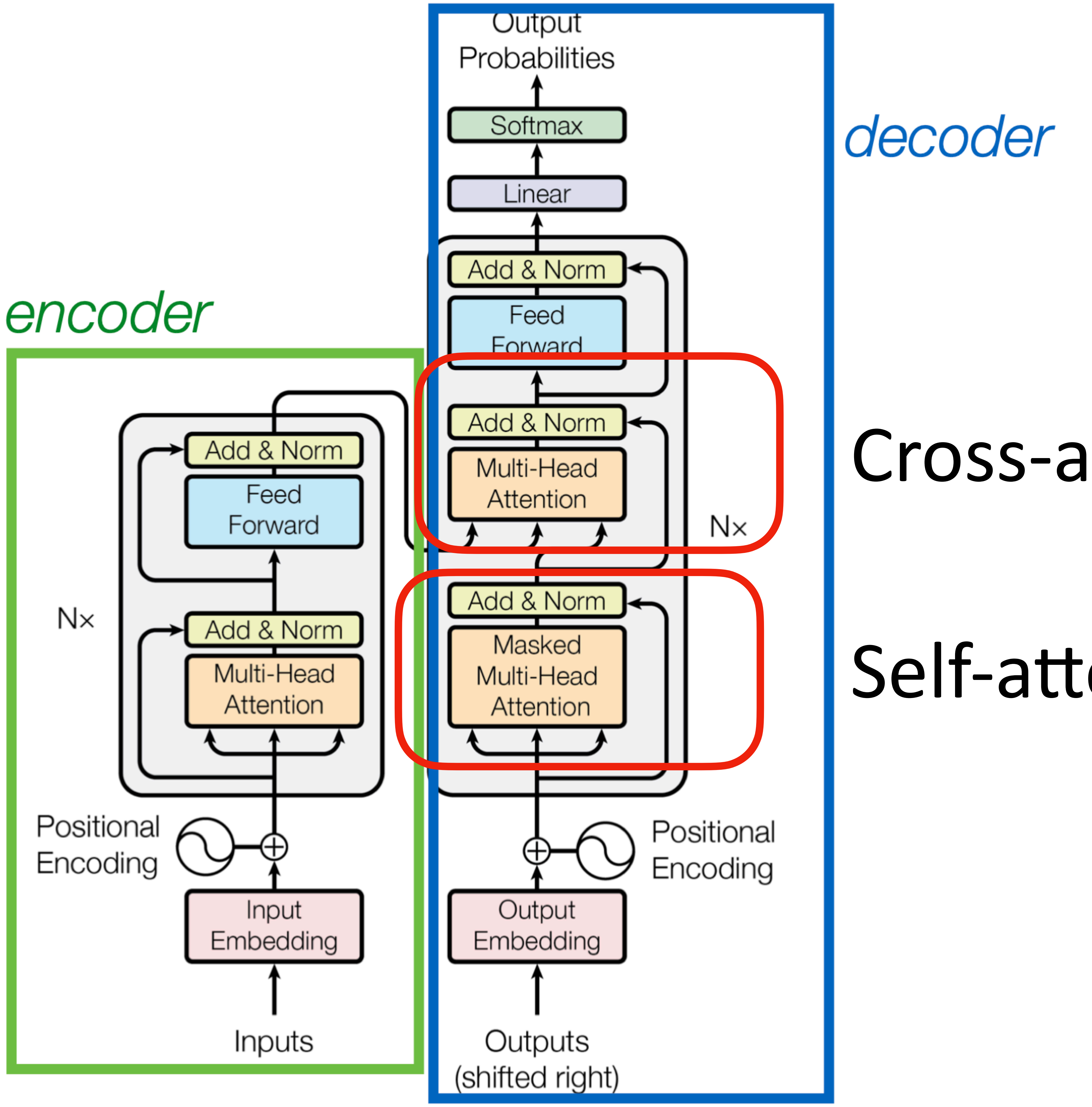
Transformer Decoder in Seq2Seq



Transformer Decoder in Seq2Seq



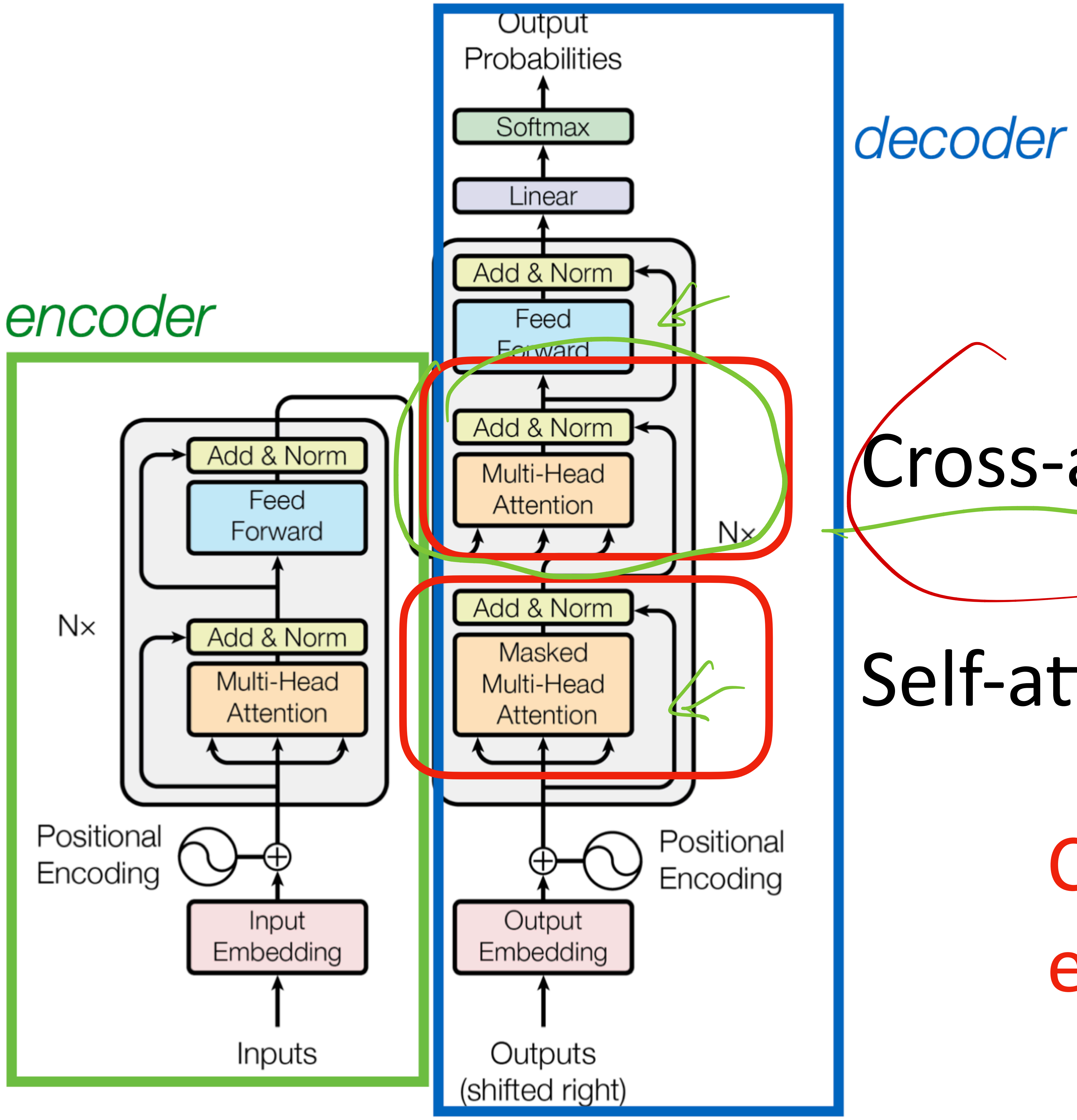
Transformer Decoder in Seq2Seq



Cross-attention

Self-attention

Transformer Decoder in Seq2Seq



Cross-attention

Self-attention

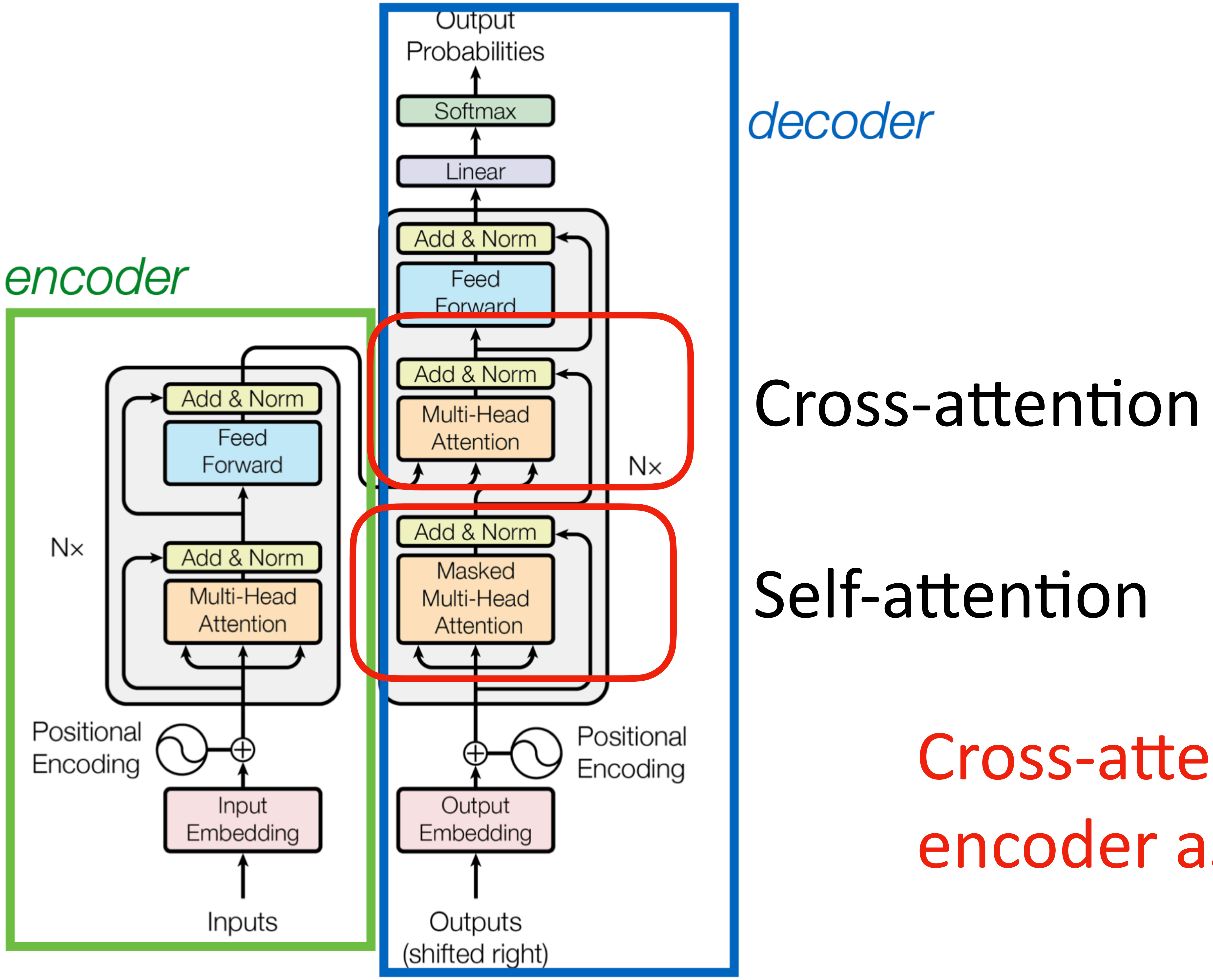
query   k, v

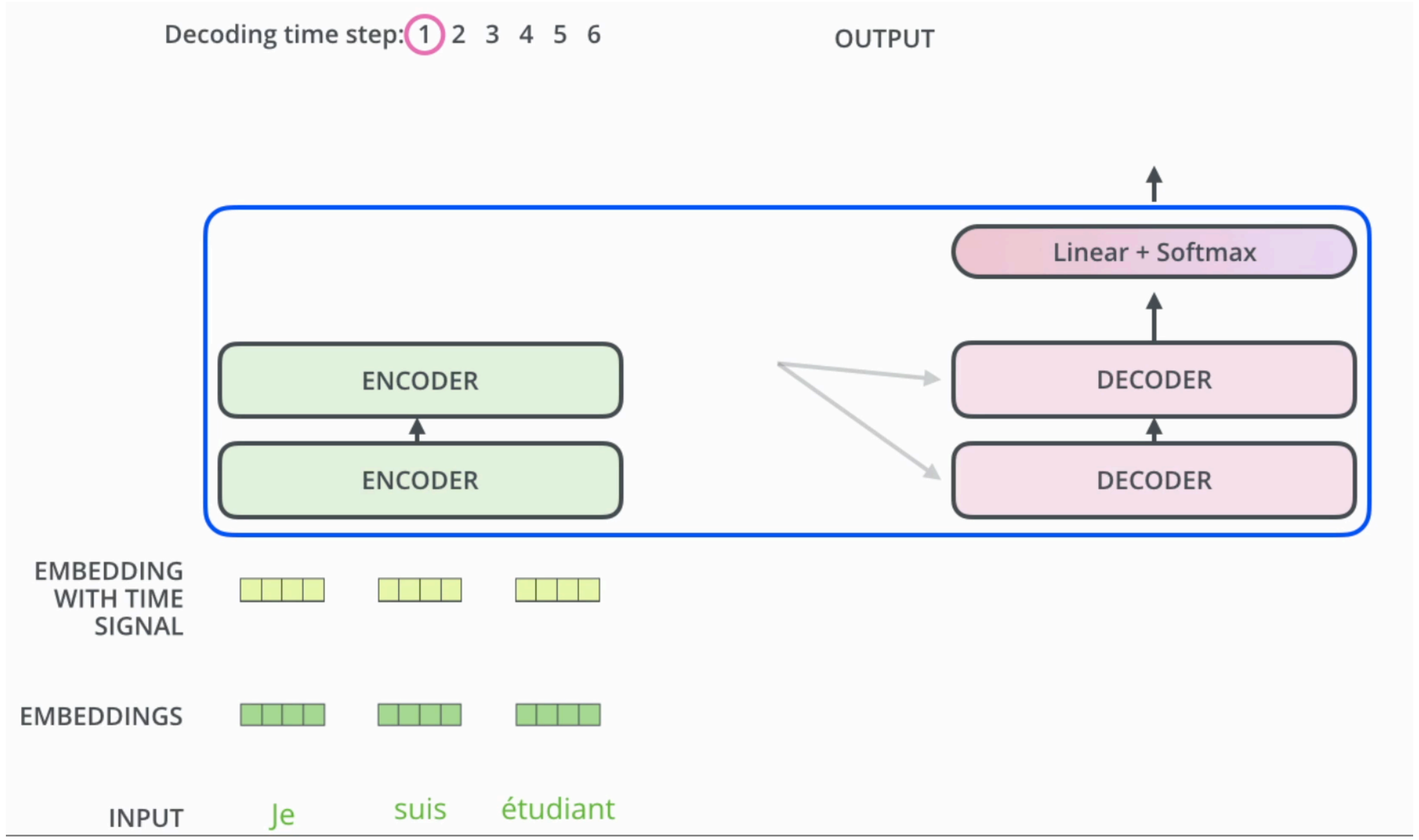
q

decoder input   encoder output

Cross-attention uses the output of encoder as input

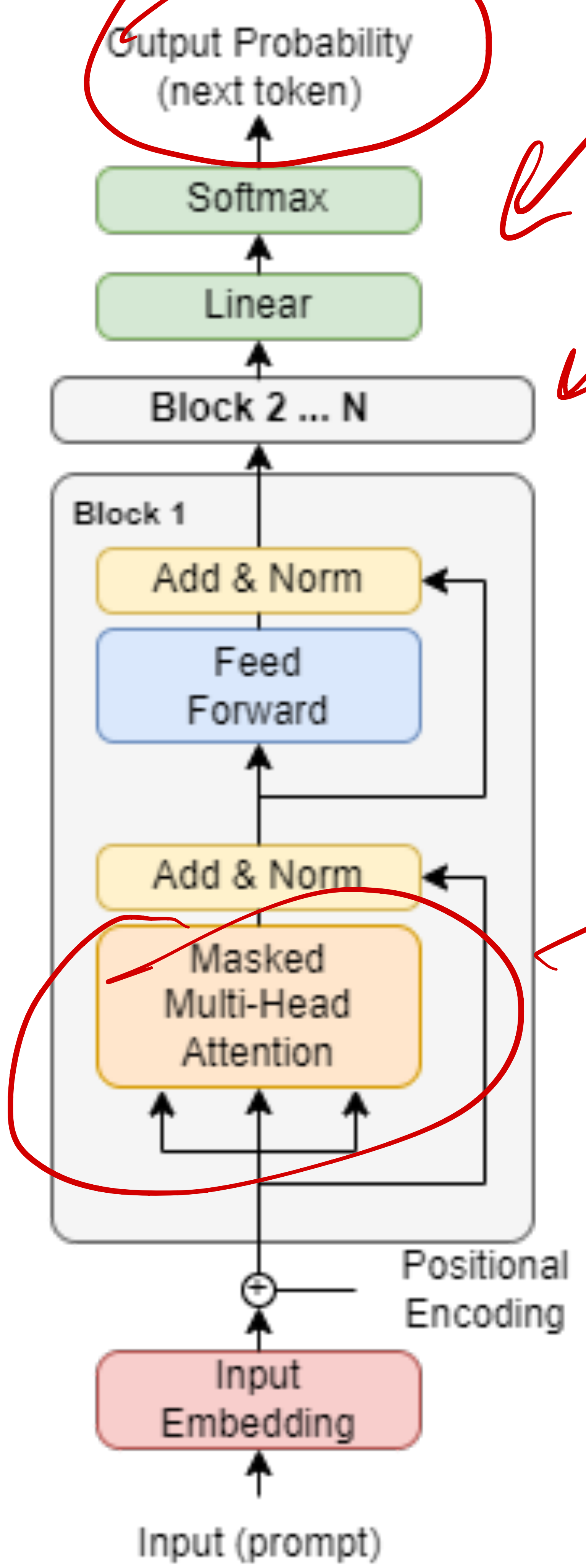
Transformer Decoder in Seq2Seq





Cross-attention uses the output of encoder as input

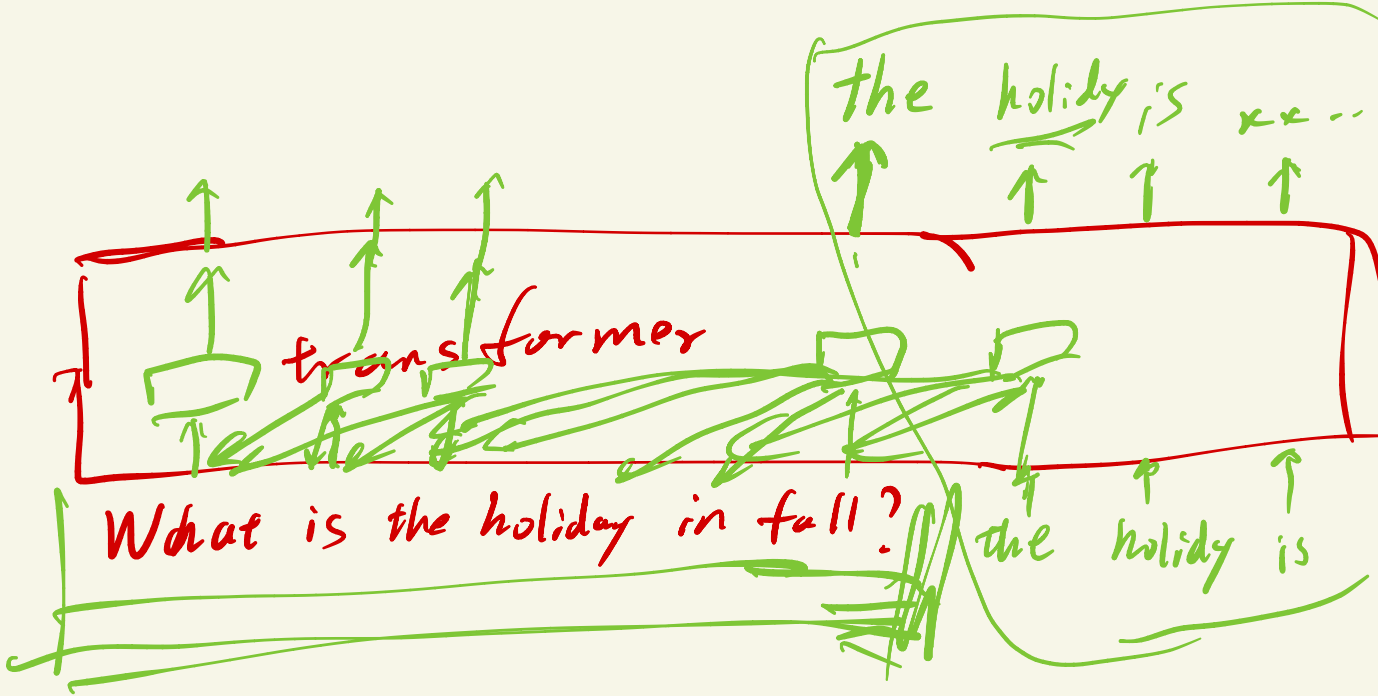
Transformer Language Model (e.g., ChatGPT)



linear → vocab size

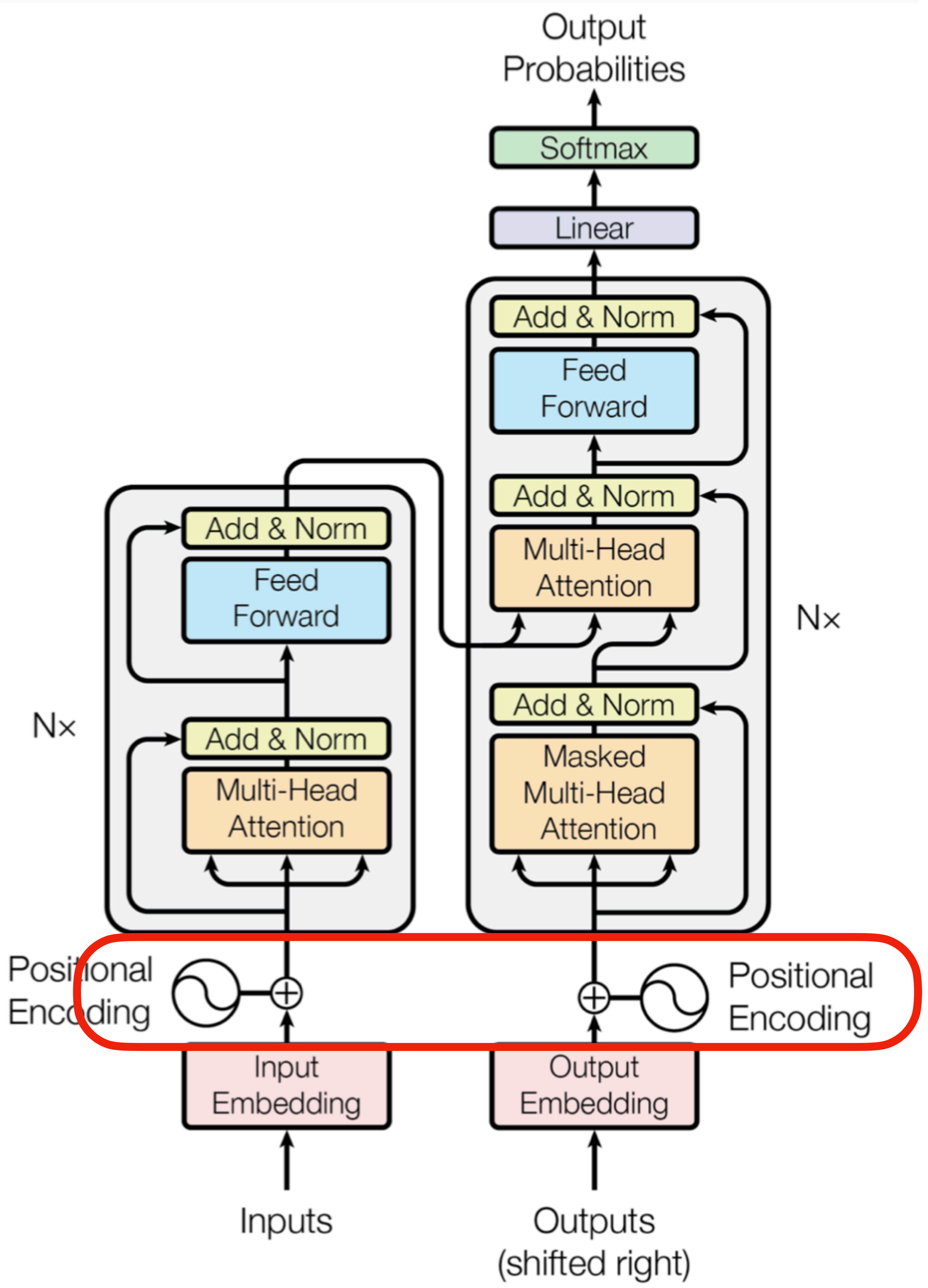
softmax →

masked attention

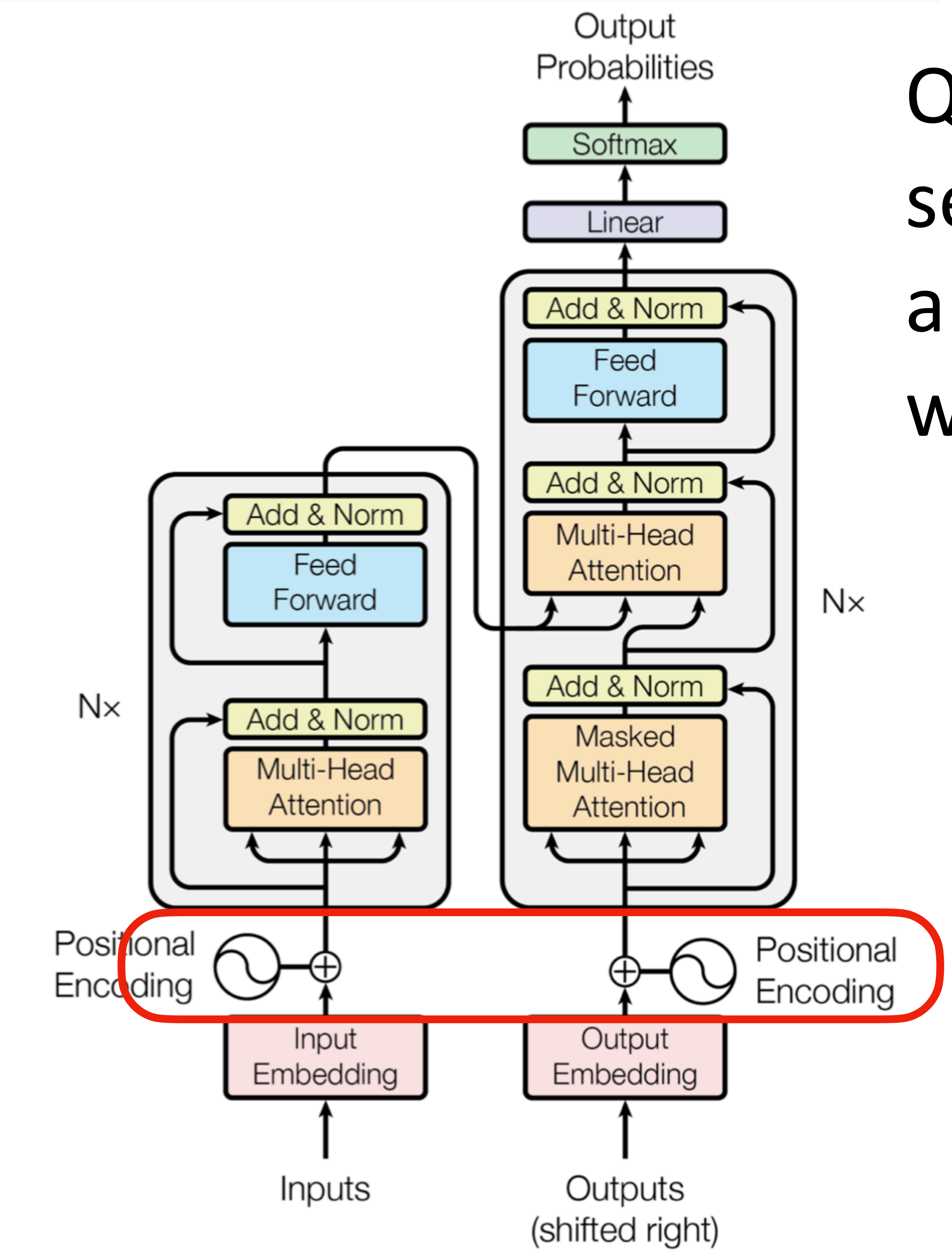


the holidy is xx...
transformer
What is the holiday in fall?
the holidy is

Position Embeddings



Position Embeddings



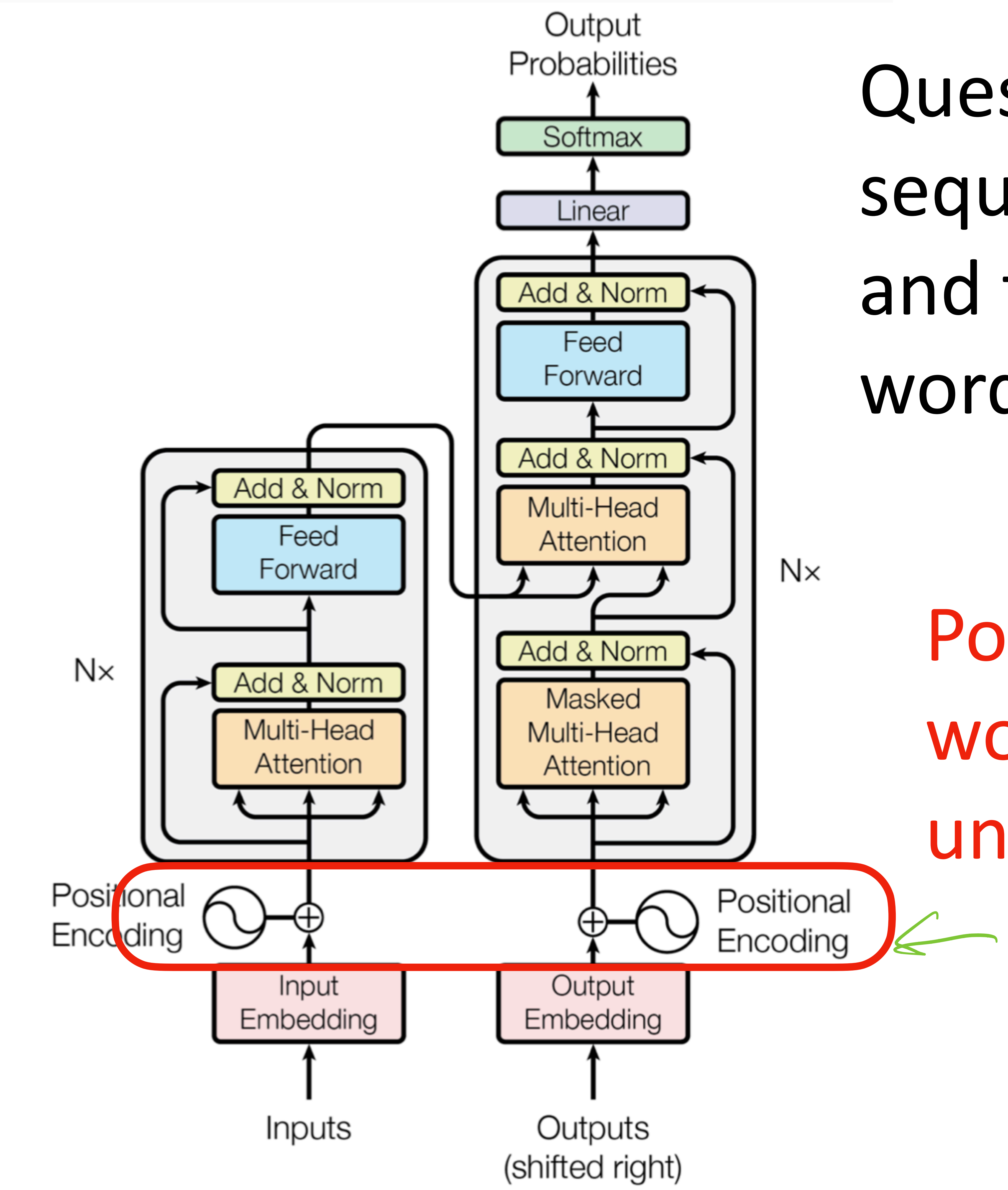
Question: If we shuffle the order of words in the sequence, will that change the attention output and feed forward output of the corresponding word?

equal

I am having a class

class have a a am
I

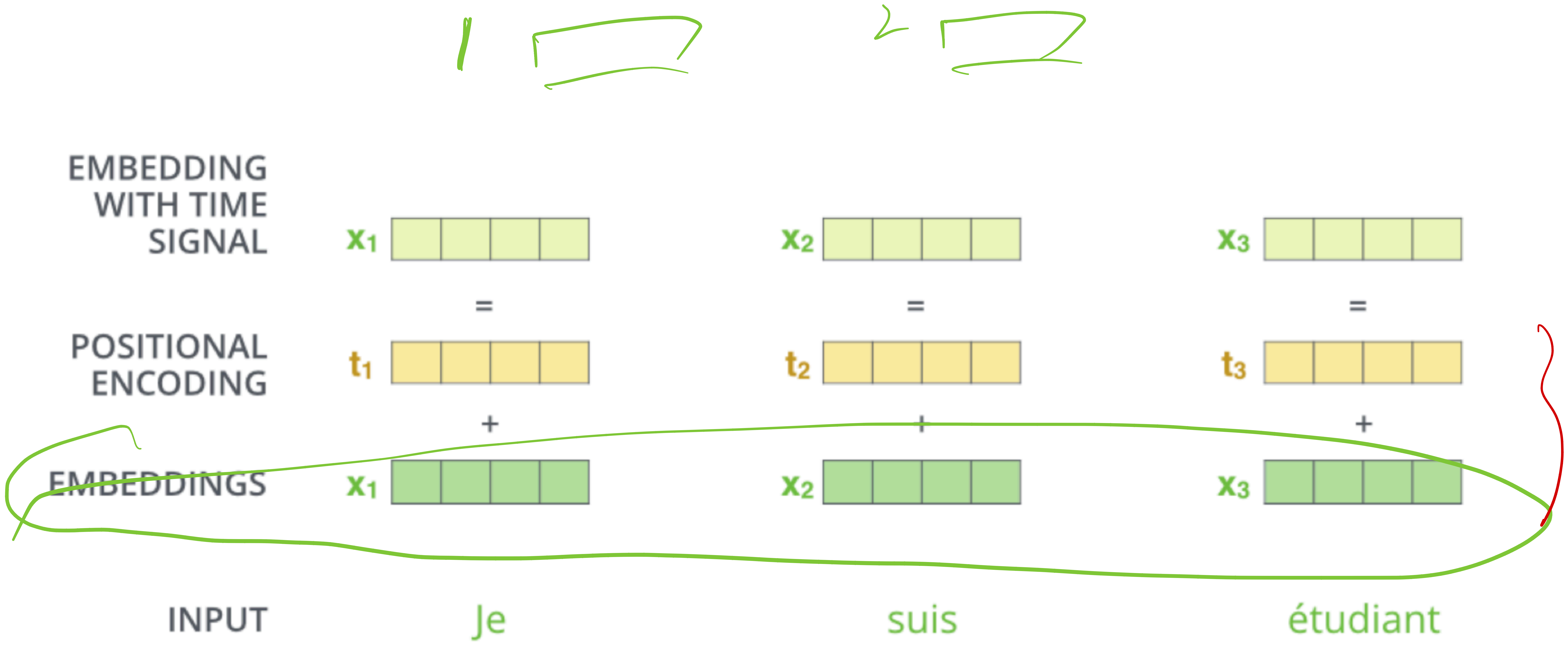
Position Embeddings



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

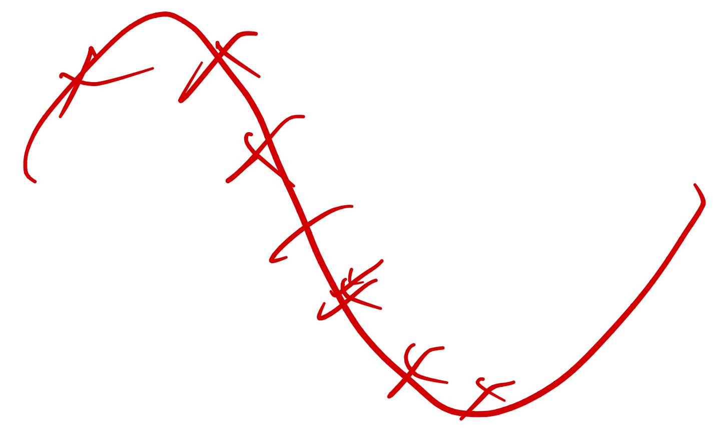


parameter

Transformer Positional Encoding

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

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



3, 4, 5

Positional encoding is a 512d vector
i = a particular dimension of this vector
pos = dimension of the word
d_model = 512

512

[x x x x x]

2i+1

pos = 3

3

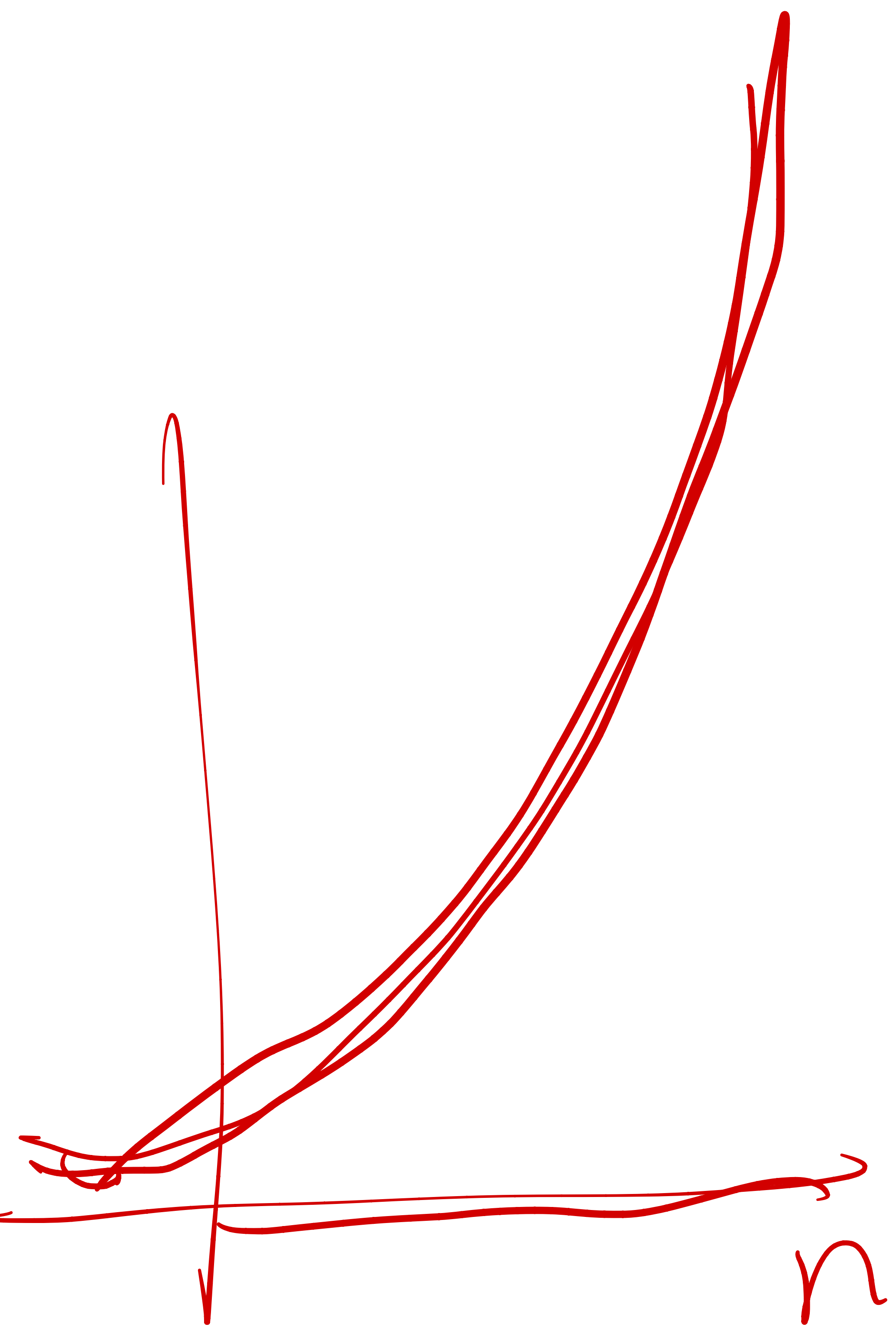
am

$\sin\left(\frac{}{1000^{2i/d=512}}\right)$

Complexity

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

n is sequence length, d is embedding dimension.



$\boxed{q \cdot k}$ QK^T $\boxed{O(n^2)}$

$Q \in \mathbb{R}^{n \times d}$

$K \in \mathbb{R}^{n \times d}$

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

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.

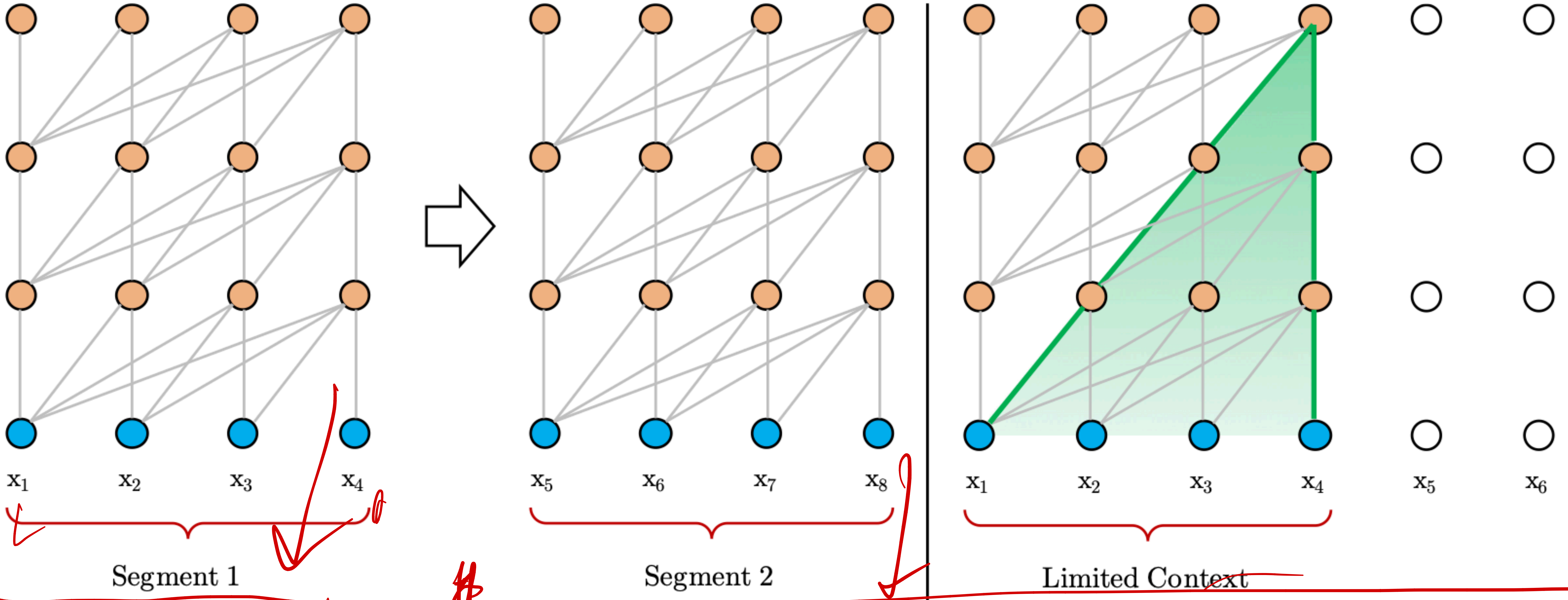
Restricted self-attention means not attending all words in the sequence, but only a restricted field

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

Language Model Training with Limited Context

context length = 2000 tokens

$O(2000^2)$



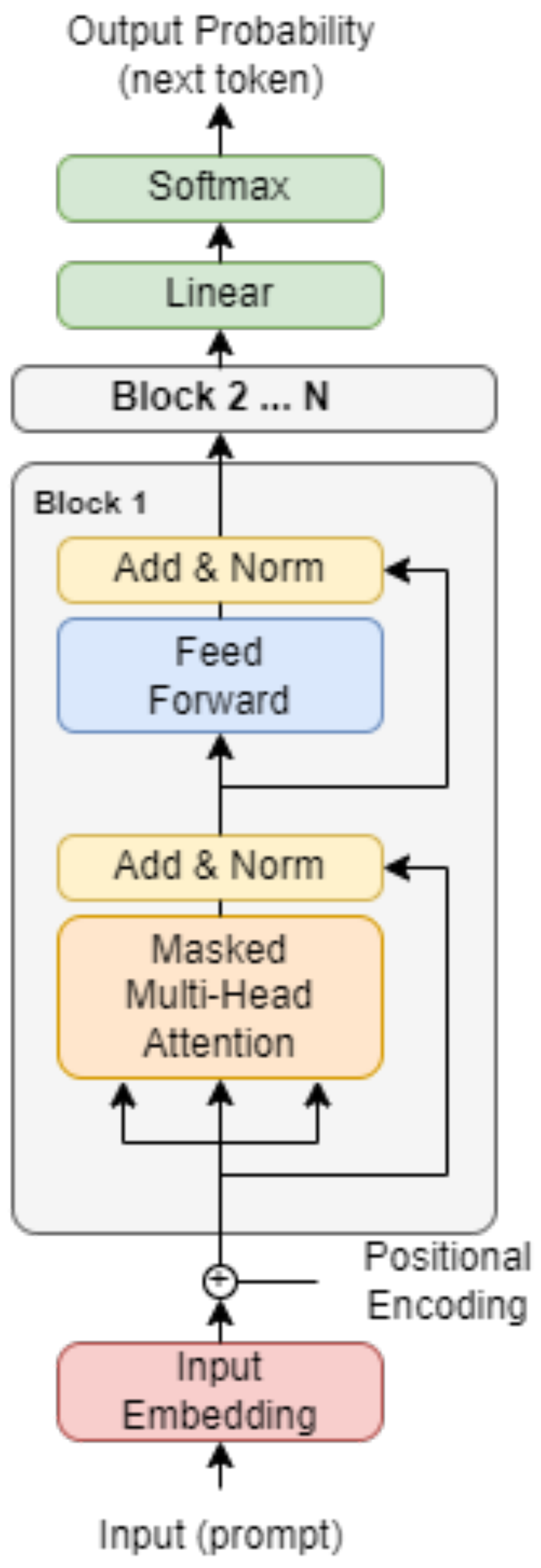
2000 tokens

2000 tokens

$O(n^2)$

Dai et al. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. 2019.

Transformer Language Model (e.g., ChatGPT)



Language Model Pretraining

Pretraining

Pretraining

Target Data B

Pretraining

Source Data A (maybe a different task)

Target Data B

Pretraining

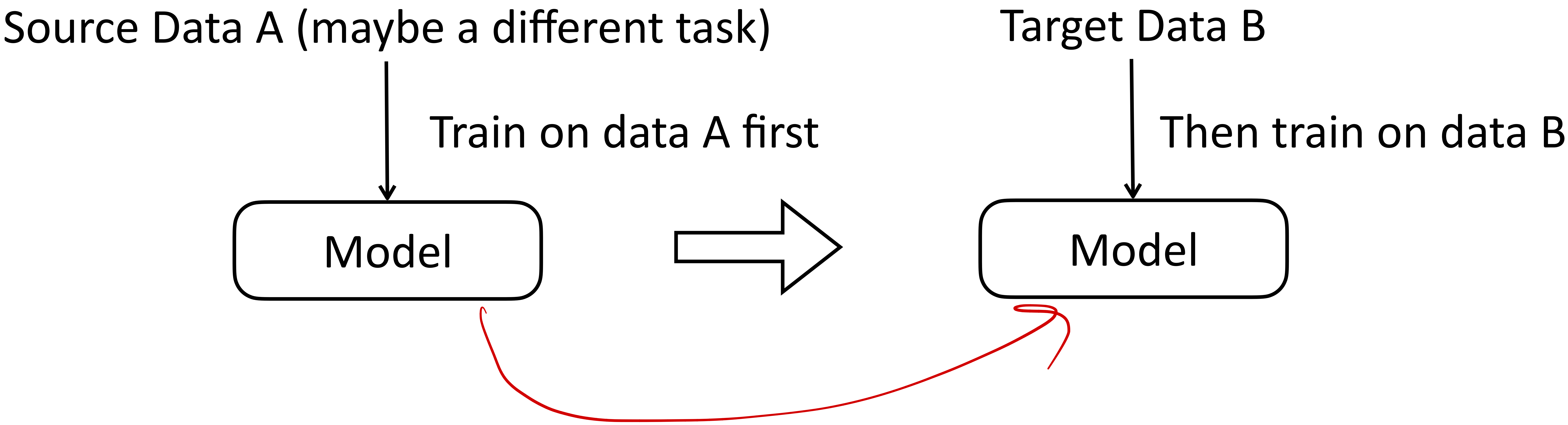
Source Data A (maybe a different task)

Target Data B

Train on data A first

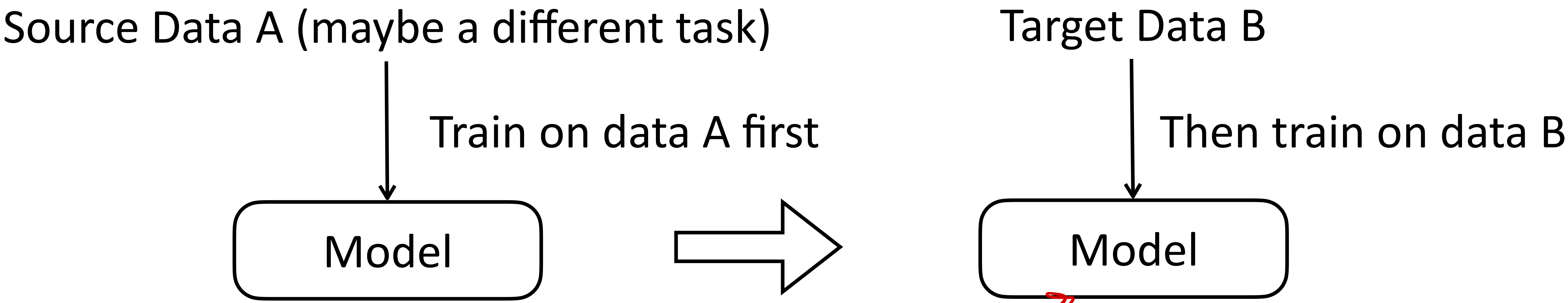
Model

Pretraining



Pretraining

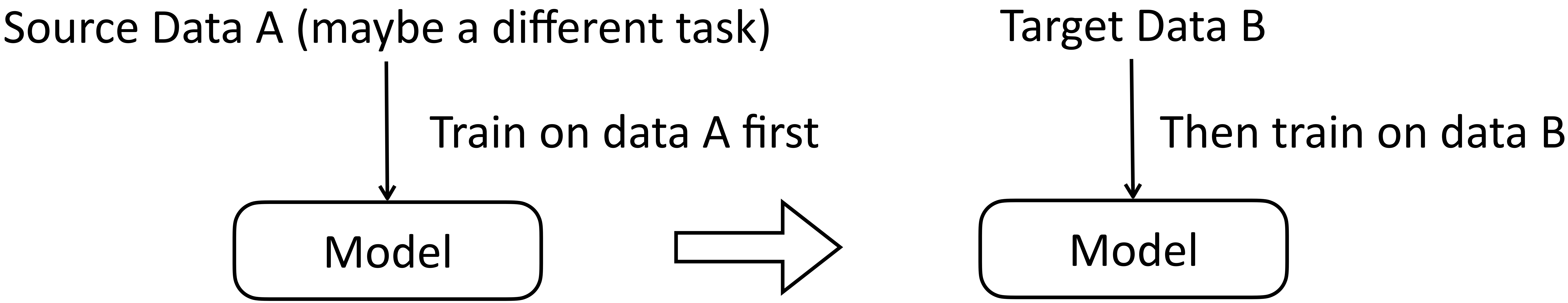
post-training



pretraining

Classically, this is transfer Learning

Pretraining



Classically, this is transfer Learning

It is now called pretraining because of the scale of A large-scale

Pretraining

energy → intelligence

no annotated data

Source Data A (maybe a different task)

Target Data B

Train on data A first

Then train on data B

Model

Model

next-token prediction

For supervised training, data A is often limited

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

Language Model

output

$\boxed{\text{Self-supervised}}$

unsupervised learning ?

BERT

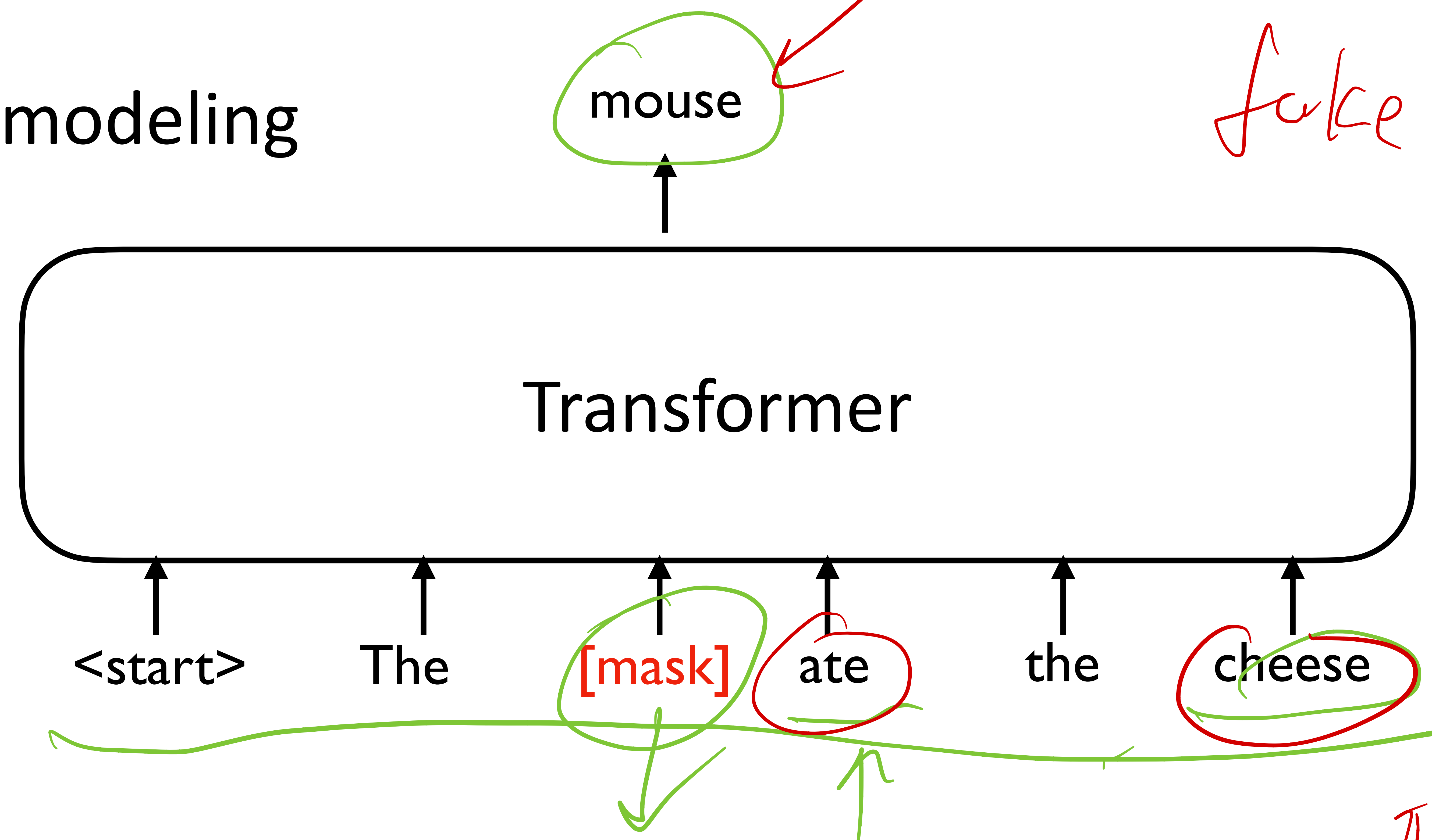
Mask language modeling

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

BERT

Mask language modeling

fake task



cheese

<start> the mouse ate the <mask>

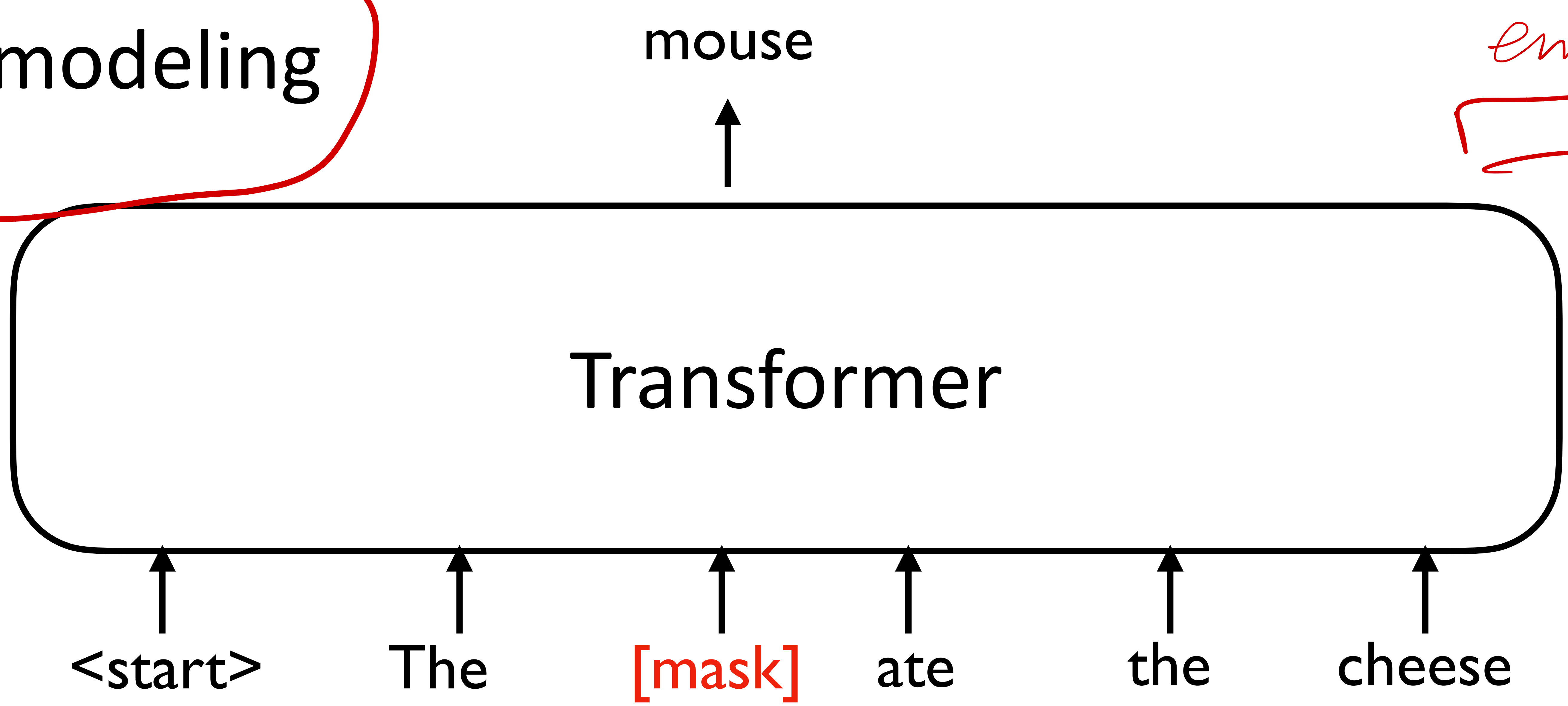
Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019.

BERT

classification

embedding

Mask language modeling



P(x)

P(x)

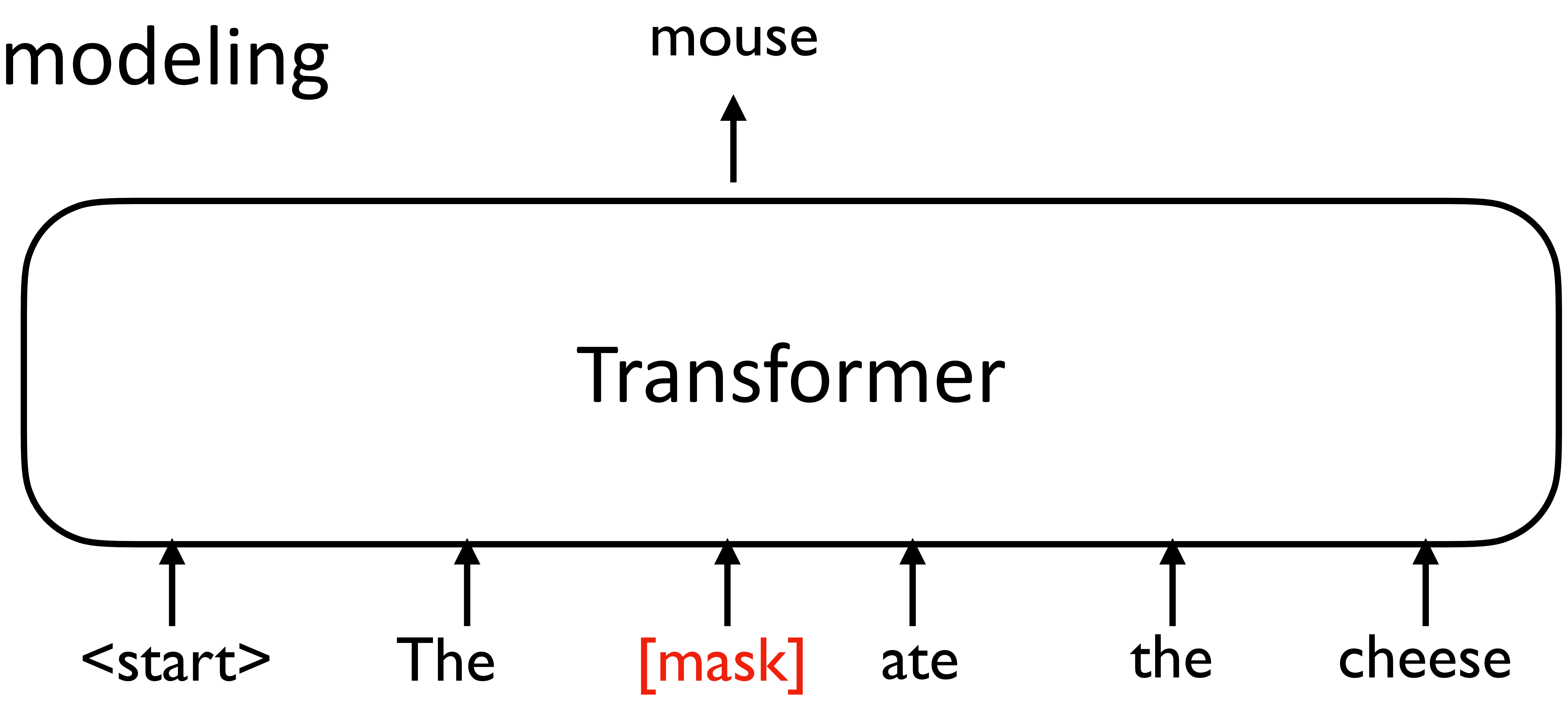
P(x|c)

Construct a synthetic task from raw text only

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

BERT

Mask language modeling



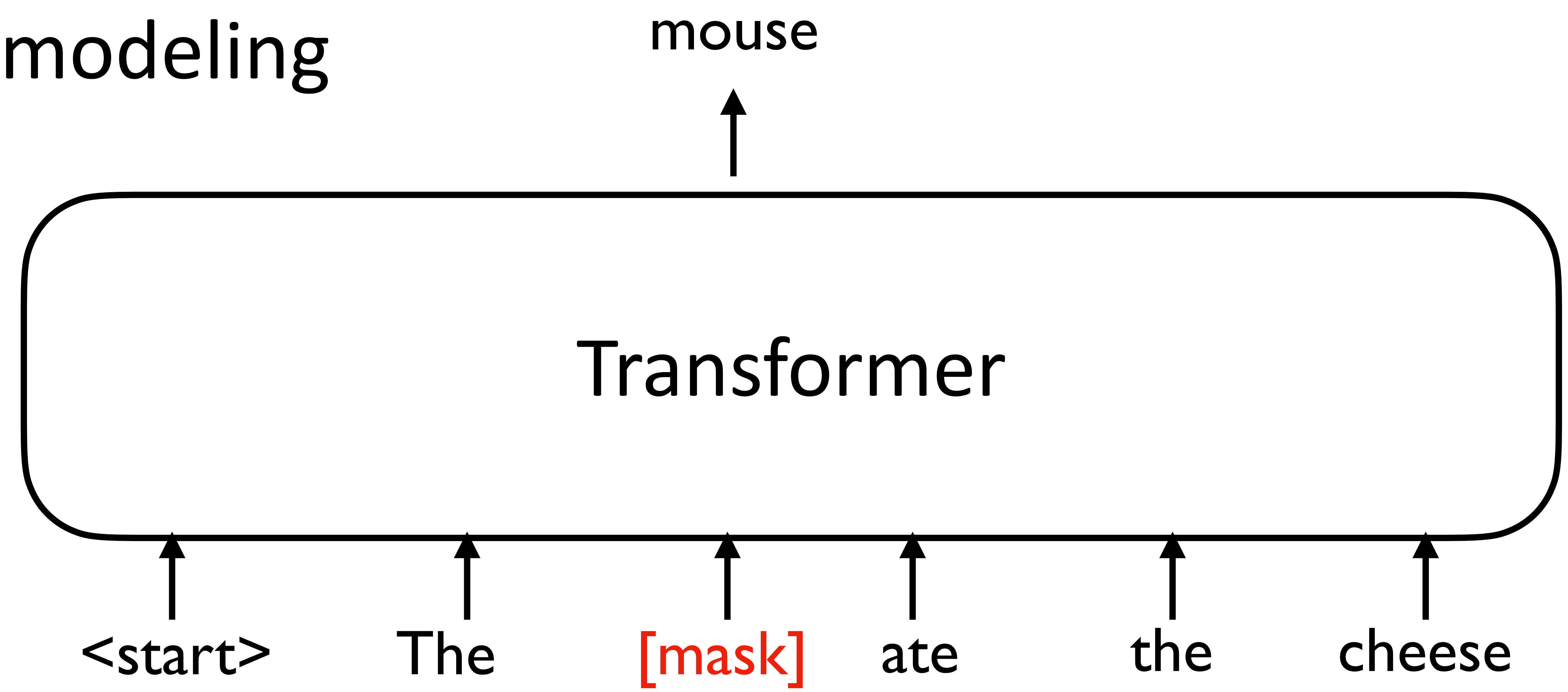
Self-supervised Learning

Construct a synthetic task from raw text only

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

BERT

Mask language modeling



Self-supervised Learning

Construct a synthetic task from raw text only

Can be made very large-scale

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

BERT

B → bidirectional

attention

Mask language modeling

mouse

↑

Transformer

↑ <start> ↑ The ↑ [mask] ↑ ate ↑ the ↑ cheese

Self-supervised Learning

Construct a synthetic task from raw text only

Can be made very large-scale

No No

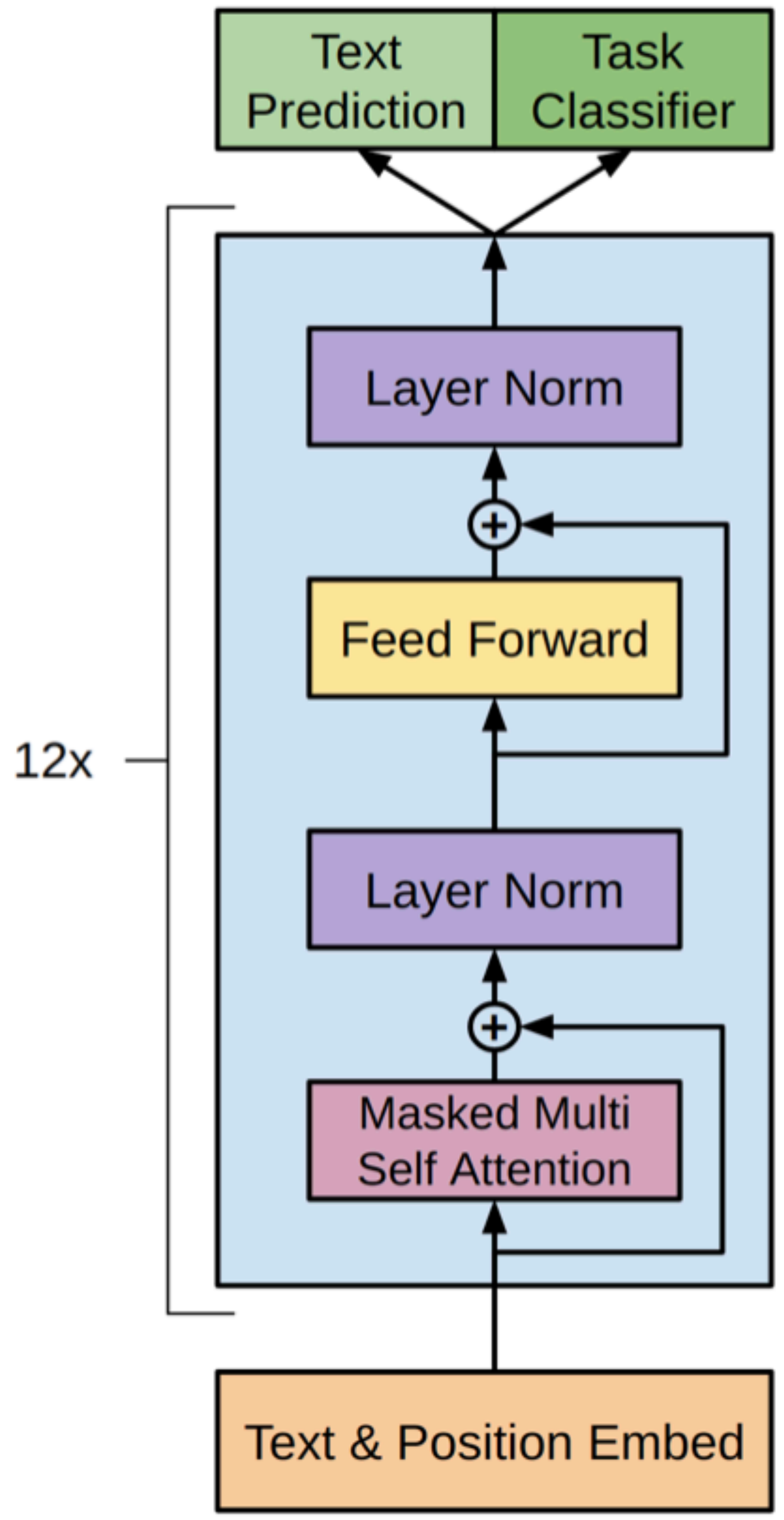
Is Bert a language model? Is it a generative model?

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

Generative Pre-Training (GPT)

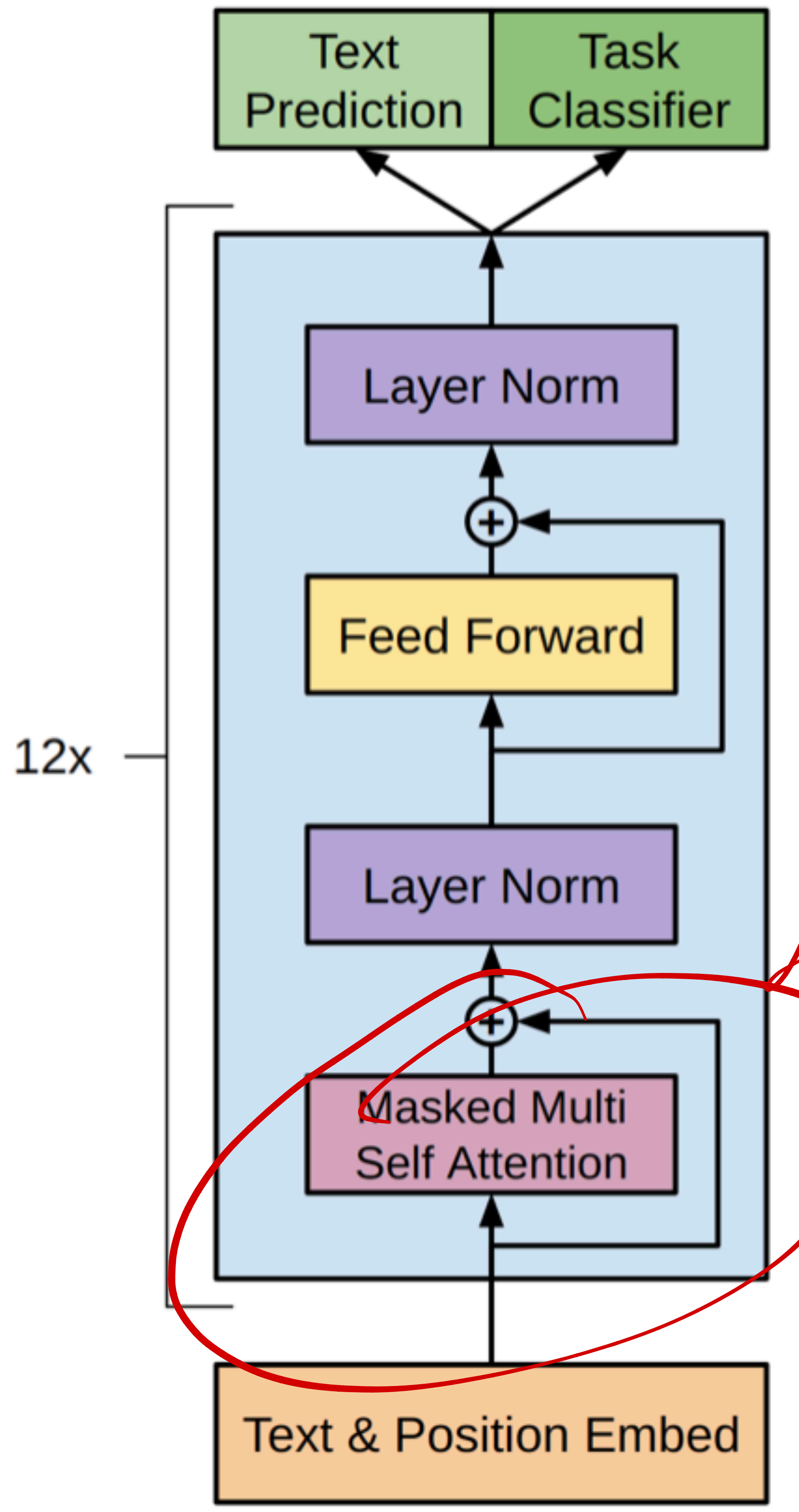


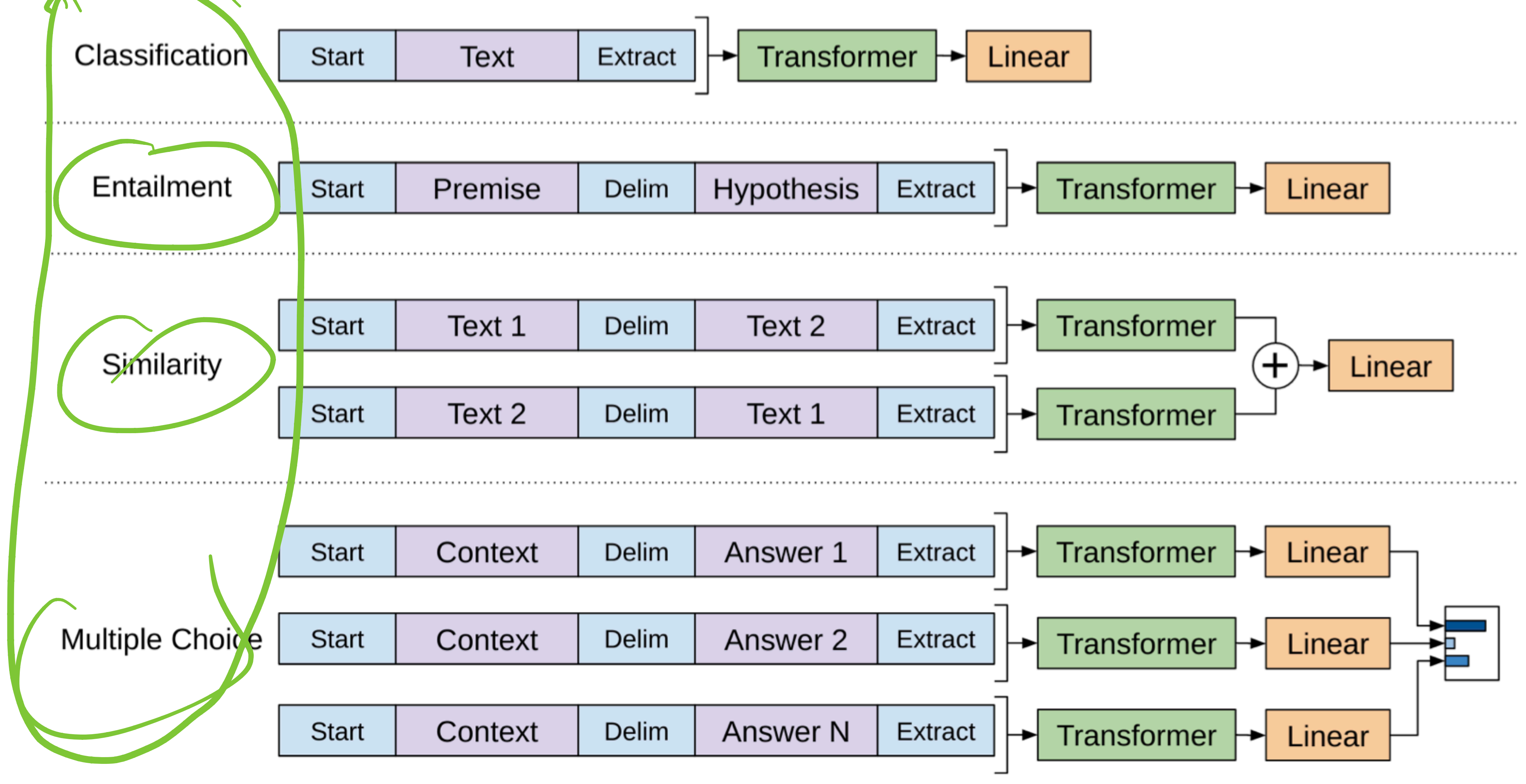
Radford et al. Improving Language Understanding by Generative Pre-Training. 2018

Generative Pre-Training (GPT)

GPT-1

masked attention





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

logistic regression

embedding → label

transformer

sentence

fine tuning

Suppose I just want to do a sentence classification task, bidirectional or masked attention is better?

I have a class

classification

IB

Suppose I just want to do a sentence classification task, bidirectional or masked attention is better for pretraining?

we don't care

suppose next-word prediction

masked attention → task more difficult

1×1= 1
1×2= 2
10×2= 20

energy → intelligence

data sample is wasted

I am from CS dept, we are having a class ___,

~~in this LLM class, we talked~~

Pretraining Data

We want to start with clean text

- Wikipedia
- Books

seg1

History [edit]

In the late 1980s, the Hong Kong Government anticipated a strong demand for university graduates to fuel an economy increasingly based on services. Sir Sze-Yuen Chung and the territory's governor, Sir Edward Youde, conceived the idea of establishing a third university, in addition to the pre-existing University of Hong Kong and Chinese University of Hong Kong.[7]

Planning for the "Third University", as the university was known provisionally, began in 1986. On 8 November 1989, Charles, Prince of Wales (now King Charles III) laid the foundation stone of the campus,[8] which was constructed at the Kohima Barracks site in Tai Po Tsai on the Clear Water Bay Peninsula. The site was earmarked for the construction of a new British Army garrison to house the 2nd King Edward VII's Own and 7th Duke of Edinburgh's Own Gurkha Rifles,[9] but plans for its construction were shelved after the 1984 signing of the Sino-British Joint Declaration resulted in the downsizing of army presence in Hong Kong.[10]

seg2

Originally scheduled to finish in 1994, the planning committee for the university decided in 1987 that the new institution should open its doors three years early, in keeping with the community's need and in fulfilment of the wishes of Youde, who died in 1986.[11][12] The university was officially opened by Youde's successor as governor, Sir David Wilson, on 10 October 1991.[13] Several leading scientists and researchers took up positions at the university in its early years, including physicist Leroy Chang who arrived in 1993 as Dean of Science and went on to become vice-president for academic affairs.[14] Thomas E. Stelson was also a founding member of the administration.[15]

Pretraining Data Reality

In practice, the web is the most viable option for data collection.

In the digital era, this is the go-to place for general domain human knowledge.

But web data can be challenging to work with

- Copyright and usage constraints, privacy
- Data is noisy, dirty, and biased

wikipedia is small

Example Noisy Web Data

```
<html ⚡ lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
  <title>Best Coffee Beans 2025 | Best Coffee Beans 2025 | Buy Coffee Now!</title>
  <meta name="description" content="best coffee beans best coffee beans best coffee bea
  <link rel="canonical" href="http://example.com/best-coffee?utm_source=spam&utm_campai
  <meta property="og:title" content="Best Coffee Beans 2025">
  <script type="application/ld+json">
  {"@context":"http://schema.org","@type":"Article","headline":"Best Coffee Beans 2025"
  </script>
  <script>
    // tracking & A/B test noise
    (function(){try{var u='https://trk.example.net/p.js?id=UA-XXX';var s=document.creat
    window.__AB__={"exp":"homepage-v17","bucket":Math.random()<0.5?'A':'B'};
  </script>
  <style>
    /* inline CSS with dead classes */
    .hero {background:url(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAA...);height:480
    .hidden{display:none} .cookie{position:fixed;bottom:0;background:#000;color:#fff;pa
    @media (max-width: 600px){ .table{display:block;overflow:auto} }
  </style>
</head>
<body id="top" class="post post post-1234" oncopy="return false">
  <!-- BEGIN Cookie banner, duplicated -->
  <div class="cookie" role="dialog" aria-live="polite">
    We use cookies to improve your experience. <button id="ok">OK</button>
  </div>
  <div class="cookie" style="display:none">We use cookies <a href="/privacy?ref=popup">
  <!-- END Cookie banner -->

  <noscript><img src="https://track.example.org/pixel.gif?id=abc" alt=""></noscript>

  <header>
    <h1>Best Coffee Beans 2025</h1>
    <div class="rating">★★★★★ 4.9/5 (3,214)</div>
    <div class="breadcrumbs">
      <a href="/">Home</a> › <a href="/category?c=cof%66ee">Coffee</a> › Best Coffee Be
    </div>
```

1+1=2

ads

Web Data Pipeline

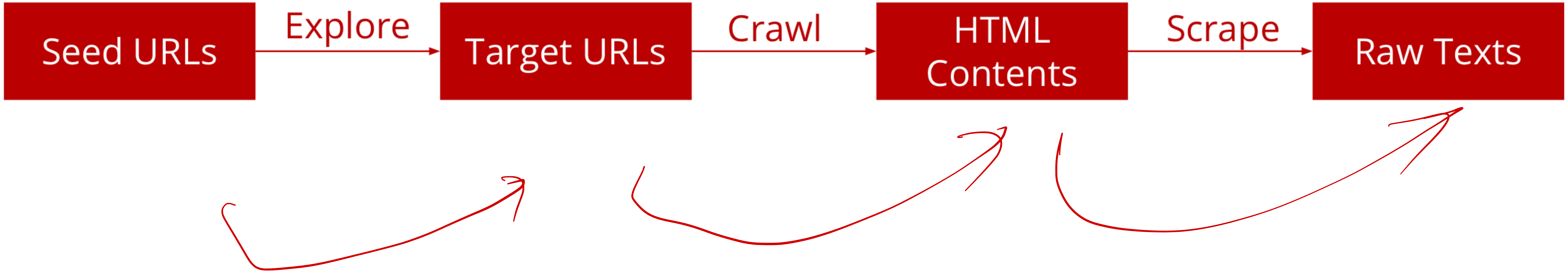
- Content is posted to the web
- Web crawlers identify and download a portion of the content
- The data is filtered and cleaned

high-quality

Web Data Pipeline

General Idea

1. Start with a set of seed websites
2. Explore outward by following all hyperlinks on the webpage.
3. Systematically download each webpage and extract the raw text.



How to Clean Text Data

How to Clean Text Data

1. Remove noisy, spammy, templated, and fragmented texts

How to Clean Text Data

1. Remove noisy, spammy, templated, and fragmented texts
2. Select higher quality texts from a massive candidate pool

How to Clean Text Data

1. Remove noisy, spammy, templated, and fragmented texts
2. Select higher quality texts from a massive candidate pool
3. Avoid toxic and biased content

What Defines Good Pretraining Data?

What Defines Good Pretraining Data?

1. Clean (fluent)

What Defines Good Pretraining Data?

1. Clean (fluent)
2. Diverse (covers many domains)

What Defines Good Pretraining Data?

1. Clean (fluent)
2. Diverse (covers many domains)
3. Non-trivial (a trivial case is to learn from massive documents and each has no more than 20 words)

What Defines Good Pretraining Data?

1. Clean (fluent)
2. Diverse (covers many domains)
3. Non-trivial (a trivial case is to learn from massive documents and each has no more than 20 words)
4. “high-quality”

What Defines Good Pretraining Data?

1. Clean (fluent)
2. Diverse (covers many domains)
3. Non-trivial (a trivial case is to learn from massive documents and each has no more than 20 words)
4. “high-quality”

    “intelligence” of the data is high, generally requiring a lot of knowledge and reasoning to predict the next word

How to Identify High-Quality Content

How to Identify High-Quality Content

- Rule-based Heuristics

How to Identify High-Quality Content

- Rule-based Heuristics
- Classifiers (how to use GPT4 to help train GPT5?)

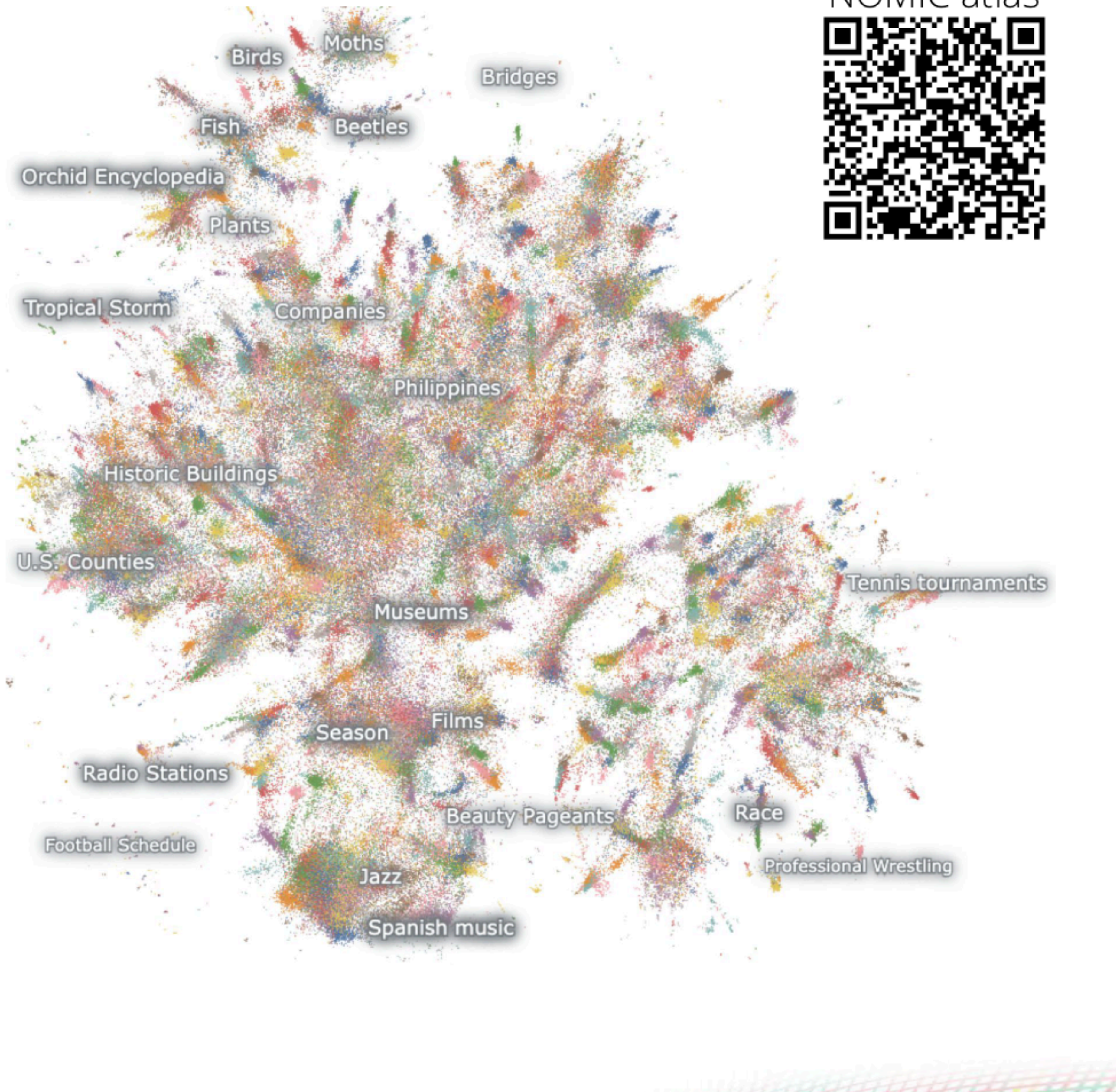
Notable Datasets

- Wikipedia dataset
- CommonCrawl
- Colossal Clean Crawled Corpus (C4)
- FineWeb
- Dolma

Wikipedia Dataset

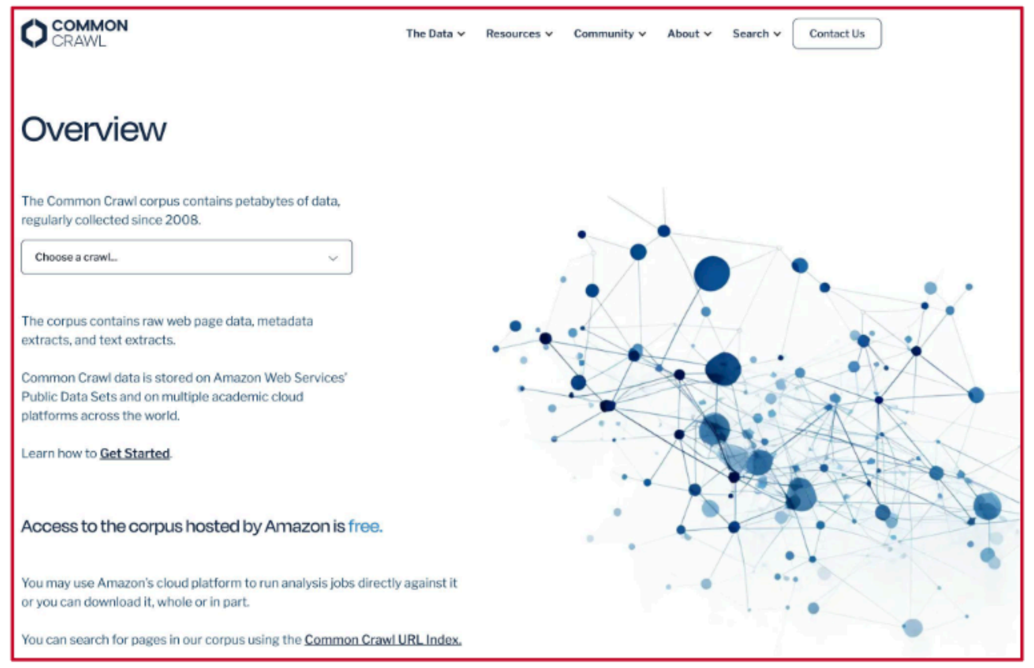
- Contains cleaned articles (65M) written in many languages (~350).
- The dataset is built from the Wikipedia dumps and split per language.
- Each example contains a cleaned article with stripped markdown and unwanted sections.
- The data fields are `id`, `url`, `title`, and `text`.
- Conveniently available on HuggingFace.

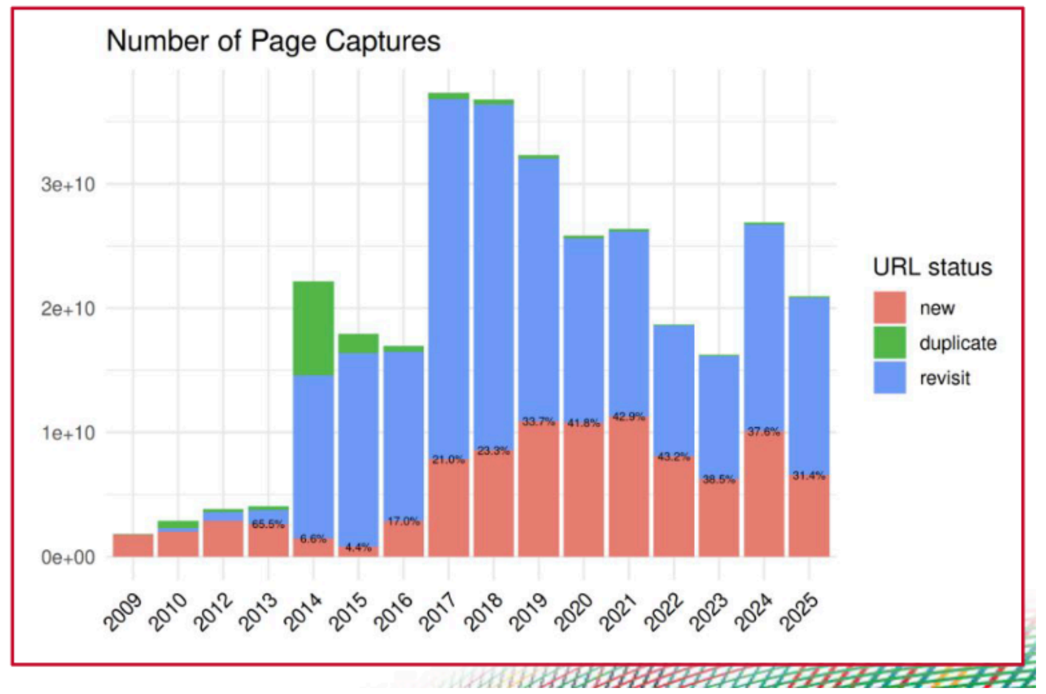
NOMIC atlas



CommonCrawl

- Non-profit organization that provides open access to large scale web crawls
- Petabytes of web pages are available
- Monthly crawls and dumps
    - Re-crawled web pages and fresh dumps (bi)monthly
    - The dumps are ~k billion pages
- Dates back to 2008





Preprocessing Clean Text

After the text is cleaned, now we need to convert it into a batch of training data

The Steelers enjoy a large, widespread fanbase nicknamed Steeler Nation. They currently play their home games at Acrisure Stadium.

Raw Clean Text

Tokenization

'_The', '_Steel', 'ers', '_enjoy', '_a', '_large', ',', '_wide', 'spre', 'ad', '_fan', 'base', '_nick', 'na', 'med', '_Steel', 'er', '_Nation', '.', '_They', '_currently', '_play', '_their', '_home', '_games', '_at', '_A', 'cris', 'ure', '_Stadium', '.'

Tokenized

Batching

[580, 109027, 1313, 25224, 9, 21333, 3, 38133, 21328, 711, 1206, 37381, 128910, 75, 4805, 109027, 55, 82580, 4, 0]
[10659, 82423, 11300, 2362, 5367, 27527, 98, 61, 58531, 3407, 88259, 4,0,0,0,0,0,0,0,0]

Tensor

Tokenizing Text

Tokenizing Text

A tokenizer takes text and turns it into a sequence of discrete tokens

Tokenizing Text

A tokenizer takes text and turns it into a sequence of discrete tokens

A vocabulary is a list of all available tokens

Tokenizing Text

A tokenizer takes text and turns it into a sequence of discrete tokens

A vocabulary is a list of all available tokens

Example: "A hippopotamus ate my homework"

Tokenizing Text

A tokenizer takes text and turns it into a sequence of discrete tokens

A vocabulary is a list of all available tokens

Example: "A hippopotamus ate my homework"

Vocab Type	Example	Length
character-level	['A', ' ', 'h', 'i', 'p', 'p', 'o', 'p', 'o', 't', 'a', 'm', 'u', 's', ' ', 'a', 't', 'e', ' ', 'm', 'y', ' ', 'h', 'o', 'm', 'e', 'w', 'o', 'r', 'k', '.']	31
subword-level	['A', 'hip', '##pop', '##ota', '##mus', 'ate', 'my', 'homework', '.']	9
word-level	['A', 'hippopotamus', 'ate', 'my', 'homework', '.']	6

Word-Level Tokenization

rule-based (split text by spaces, punctuation, and other similar heuristics)

Word-Level Tokenization

rule-based (split text by spaces, punctuation, and other similar heuristics)

Challenges

Word-Level Tokenization

rule-based (split text by spaces, punctuation, and other similar heuristics)

Challenges

- Open vocabulary problem
  - Many words may never appear in training data (becomes [UNK])
  - This is more severe in other low-resource languages

Word-Level Tokenization

rule-based (split text by spaces, punctuation, and other similar heuristics)

Challenges

- Open vocabulary problem
  - Many words may never appear in training data (becomes [UNK])
  - This is more severe in other low-resource languages
- Words with typos also get tokenized as [UNK]

Character-Level Tokenization

Vocab Type	Example	Length
character-level	['A', ' ', 'h', 'i', 'p', 'p', 'o', 'p', 'o', 't', 'a', 'm', 'u', 's', ' ', 'a', 't', 'e', ' ', 'm', 'y', ' ', 'h', 'o', 'm', 'e', 'w', 'o', 'r', 'k', '.']	31
subword-level	['A', 'hip', '##pop', '##ota', '##mus', 'ate', 'my', 'homework', '.']	9
word-level	['A', 'hippopotamus', 'ate', 'my', 'homework', '.']	6

Pro: No unseen tokens anymore

Con: Sequence is unnecessarily long, expensive to work with

Sub-word Tokenization

Vocab Type	Example	Length
character-level	['A', ' ', 'h', 'i', 'p', 'p', 'o', 'p', 'o', 't', 'a', 'm', 'u', 's', ' ', 'a', 't', 'e', ' ', 'm', 'y', ' ', 'h', 'o', 'm', 'e', 'w', 'o', 'r', 'k', '.']	31
subword-level	['A', 'hip', '##pop', '##ota', '##mus', 'ate', 'my', 'homework', '.']	9
word-level	['A', 'hippopotamus', 'ate', 'my', 'homework', '.']	6

Sub-word Tokenization

- Words get split into multiple tokens

Vocab Type	Example	Length
character-level	['A', ' ', 'h', 'i', 'p', 'p', 'o', 'p', 'o', 't', 'a', 'm', 'u', 's', ' ', 'a', 't', 'e', ' ', 'm', 'y', ' ', 'h', 'o', 'm', 'e', 'w', 'o', 'r', 'k', '.']	31
subword-level	['A', 'hip', '##pop', '##ota', '##mus', 'ate', 'my', 'homework', '.']	9
word-level	['A', 'hippopotamus', 'ate', 'my', 'homework', '.']	6

Sub-word Tokenization

- Words get split into multiple tokens
- Vocabulary is build dynamically
  - Frequent words get assigned their own tokens
  - Rare words are split into subwords

Vocab Type	Example	Length
character-level	['A', ' ', 'h', 'i', 'p', 'p', 'o', 'p', 'o', 't', 'a', 'm', 'u', 's', ' ', 'a', 't', 'e', ' ', 'm', 'y', ' ', 'h', 'o', 'm', 'e', 'w', 'o', 'r', 'k', '.']	31
subword-level	['A', 'hip', '##pop', '##ota', '##mus', 'ate', 'my', 'homework', '.']	9
word-level	['A', 'hippopotamus', 'ate', 'my', 'homework', '.']	6

Byte Pair Encoding (BPE)

Main Idea

- Construct subword vocabulary by learning to merge characters
- Inspiration comes from compression algorithms

Training Steps

1. Initialize the vocabulary with characters as tokens (e.g., in English: alphabet, numbers, punctuation)
2. Merge the most frequent token pair in the corpus (vocabulary size +1)
3. Re-tokenize the corpus with the merged subword pair
4. Repeat steps 2 and 3 until the target vocabulary size is reached

Advantages of Subword Tokenization

Advantages of Subword Tokenization

- Controlled vocabulary size

Advantages of Subword Tokenization

- Controlled vocabulary size
- Strike a good balance between word-level and character-level

Advantages of Subword Tokenization

- Controlled vocabulary size
- Strike a good balance between word-level and character-level
  - Frequent words kept whole

Advantages of Subword Tokenization

- Controlled vocabulary size
- Strike a good balance between word-level and character-level
  - Frequent words kept whole
  - Tail words split to sub-words

Advantages of Subword Tokenization

- Controlled vocabulary size
- Strike a good balance between word-level and character-level
  - Frequent words kept whole
  - Tail words split to sub-words
    - More observations on sub-words

Advantages of Subword Tokenization

- Controlled vocabulary size
- Strike a good balance between word-level and character-level
  - Frequent words kept whole
  - Tail words split to sub-words
    - More observations on sub-words
    - Utilization of morphology information

Batching Data

The Steelers enjoy a large,
widespread fanbase
nicknamed Steeler Nation.
They currently play their
home games at Acrisure
Stadium.

Raw Clean Text

Tokenization

'_The', '_Steel', 'ers', '_enjoy', '_a',
'_large', ',', '_wide', 'spre', 'ad', '_fan',
'base', '_nick', 'na', 'med', '_Steel', 'er[illegible]
'_Nation', '.', '_They', '_currently',
'_play', '_their', '_home', '_games',
'_at', '_A', 'cris', 'ure', '_Stadium', '.'

Tokenized

Batching

[580, 109027, 1313, 25224, 9,
21333, 3, 38133, 21328, 711, 1206,
3[illegible]381, 128910, 75, 4805, 109027,
55, 82580, 4, 0]
[10659, 82423, 11300, 2362,
5367, 27527, 98, 61, 58531, 3407,
88259, 4,0,0,0,0,0,0,0,0]

Tensor

Thank You!