# Transformer Model

**Shusen Wang**

# Transformer Model

- **Original paper:** Vaswani et al. Attention Is All You Need. In *NIPS*, 2017.

**Attention Is All You Need**


**Ashish Vaswani*** Google Brain avaswani@google.com

**Noam Shazeer*** Google Brain noam@google.com

**Niki Parmar*** Google Research nikip@google.com

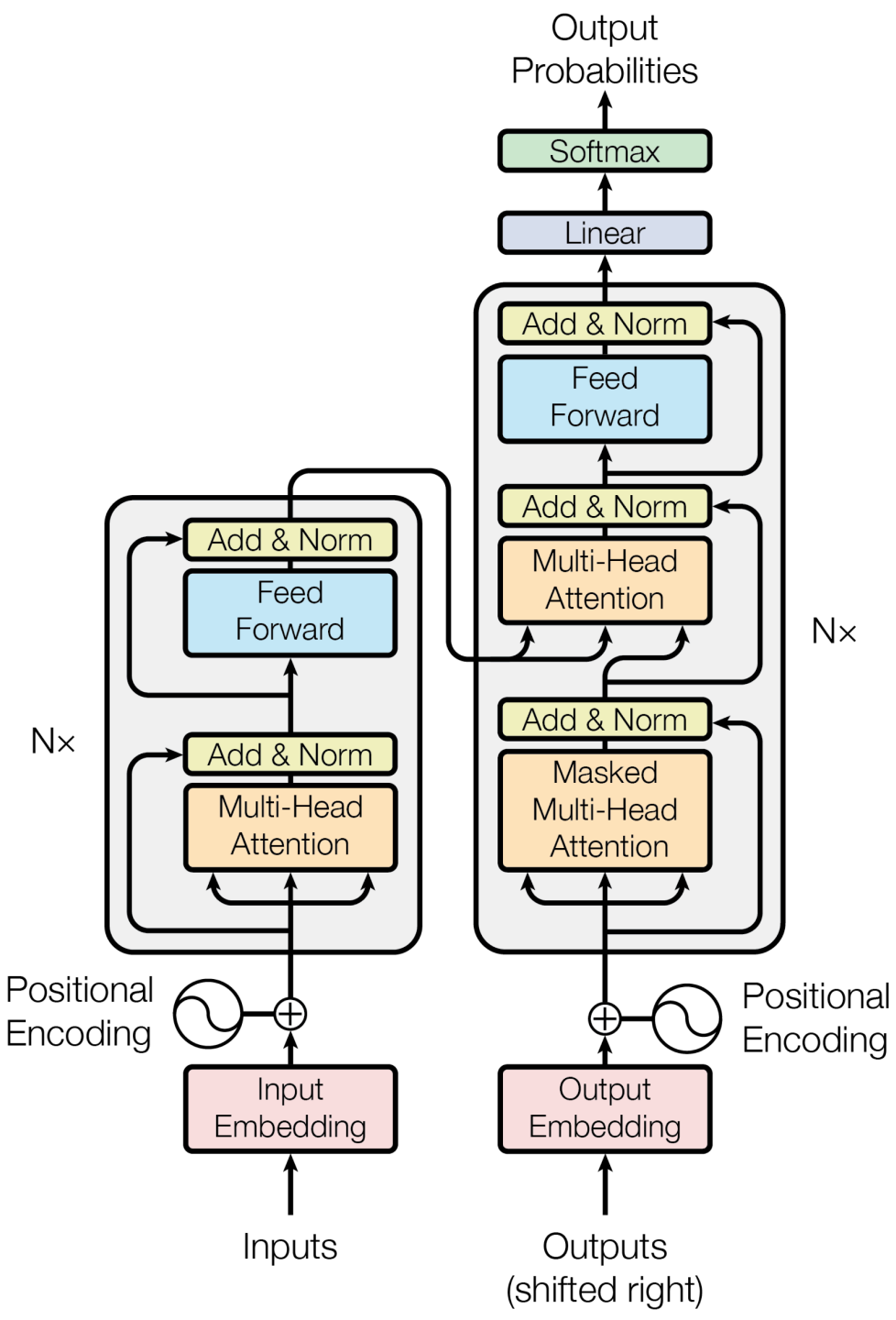
**Jakob Uszkoreit*** Google Research usz@google.com

**Llion Jones*** Google Research llion@google.com

**Aidan N. Gomez*** † University of Toronto aidan@cs.toronto.edu

**Łukasz Kaiser*** Google Brain lukaszkaiser@google.com

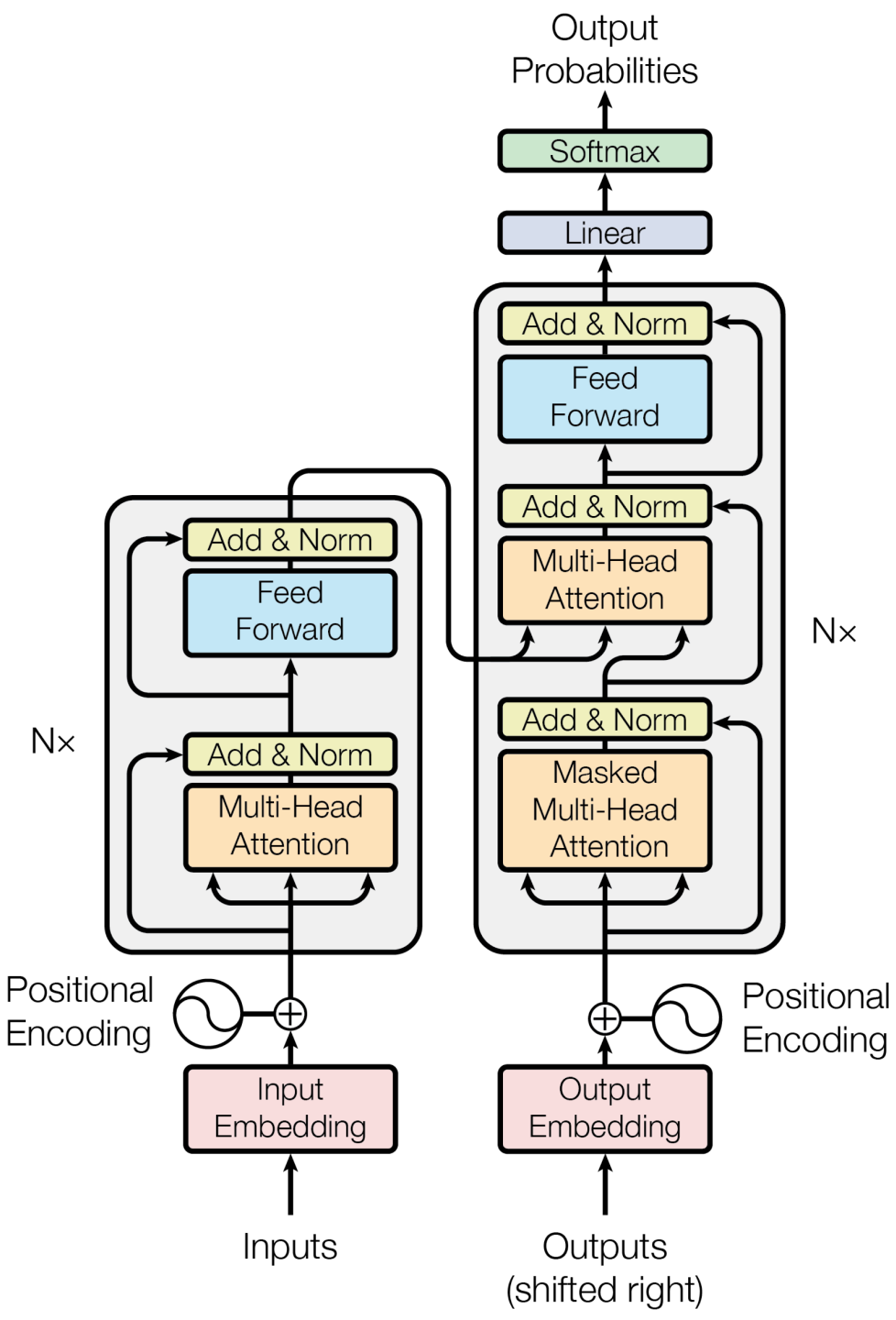
**Illia Polosukhin*** ‡ illia.polosukhin@gmail.com

# Transformer Model

- Transformer is a Seq2Seq model.
- Transformer is not RNN.
- Purely based attention and fully-connected layers.

- Much more computations than RNNs.
- Higher performance than RNNs on large datasets.

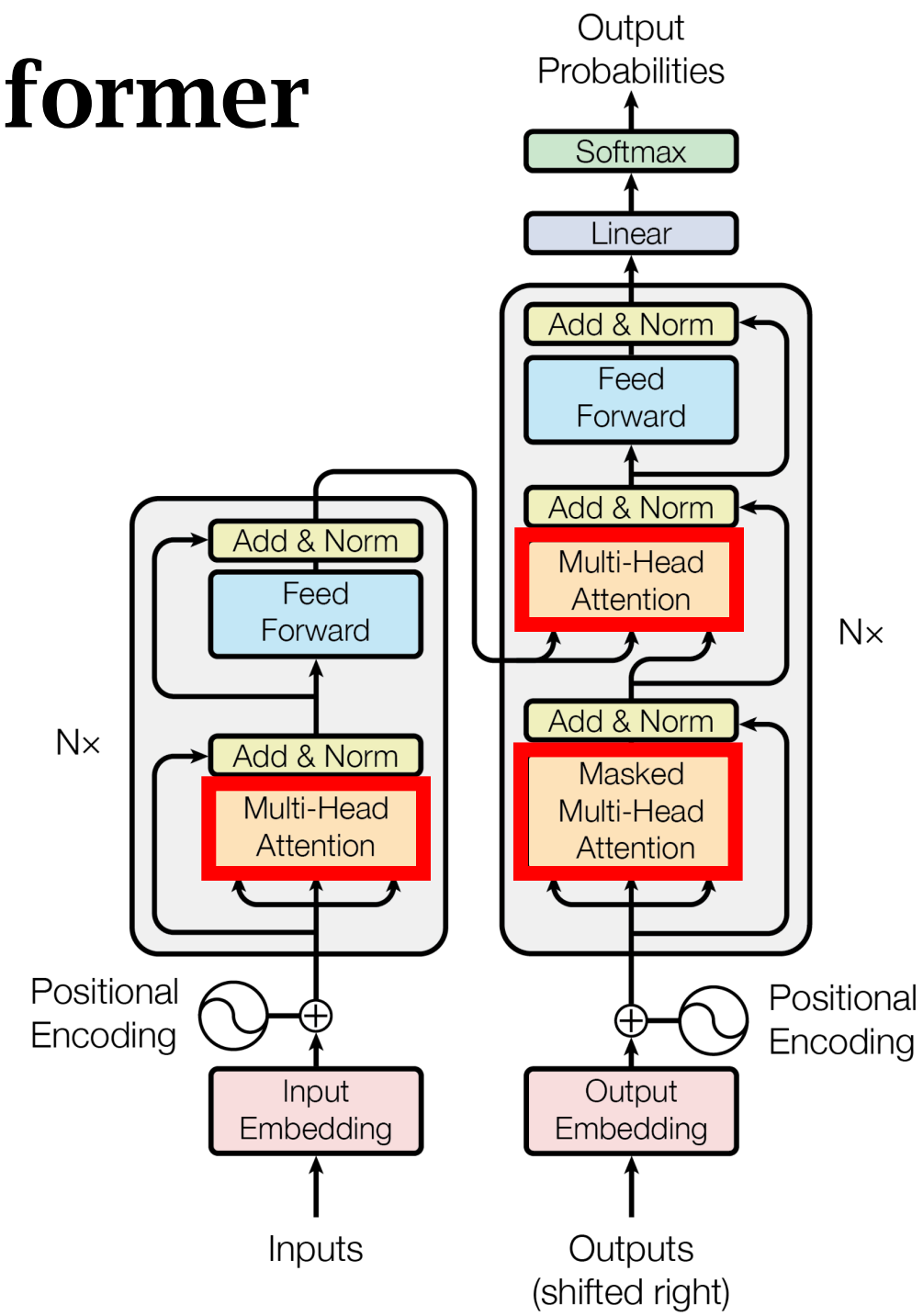


# Attention beyond RNNs

# Attention in Transformer

**Multi-head attention:**

- Multiple single-head attentions, each has its own parameter matrices.
- Concatenate the outputs.



# Attention in Transformer

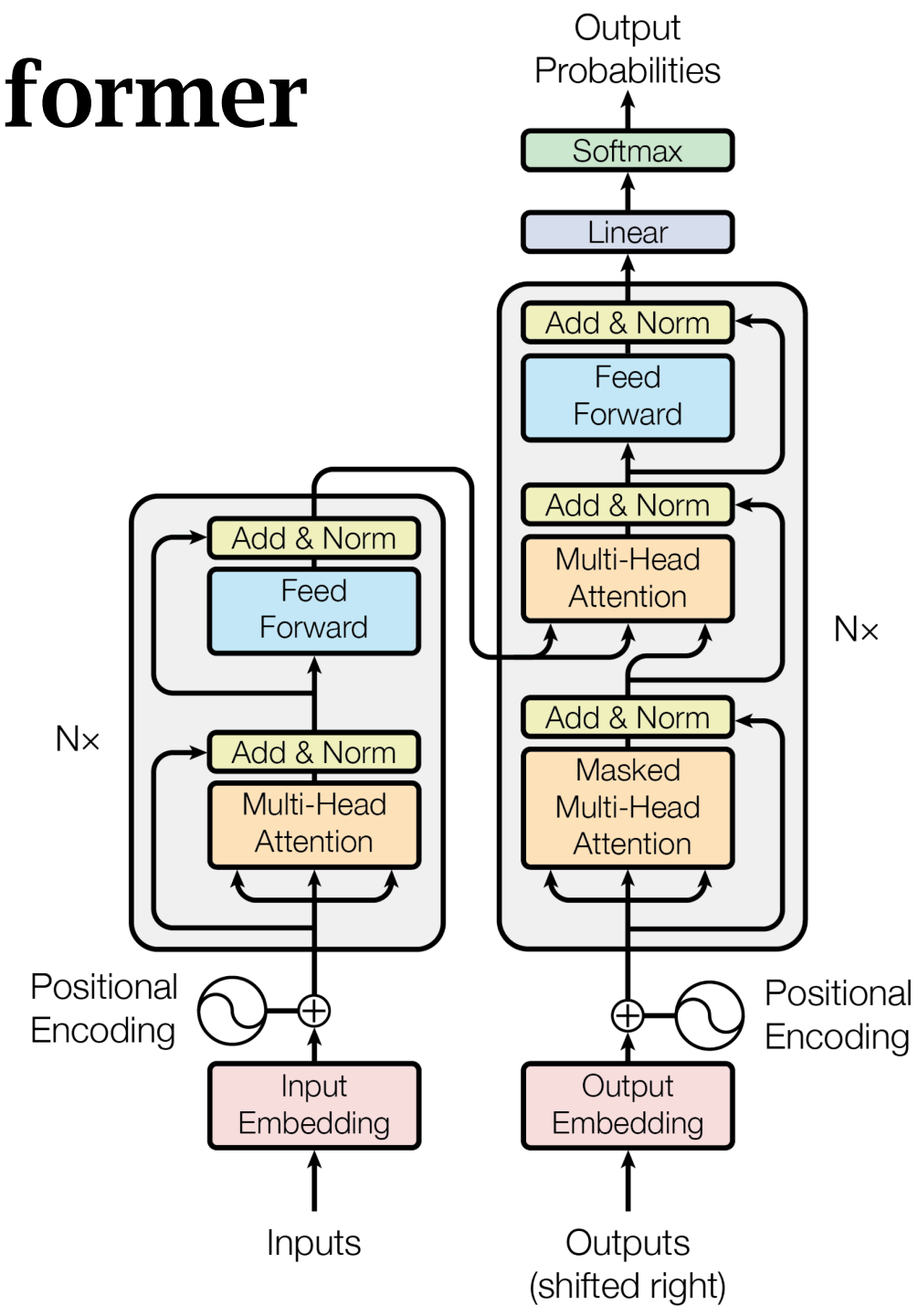
**Multi-head attention:**

- Multiple single-head attentions, each has its own parameter matrices.
- Concatenate the outputs.

**Single-head attention:**

- $\mathbf{Z} = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$.

query key value



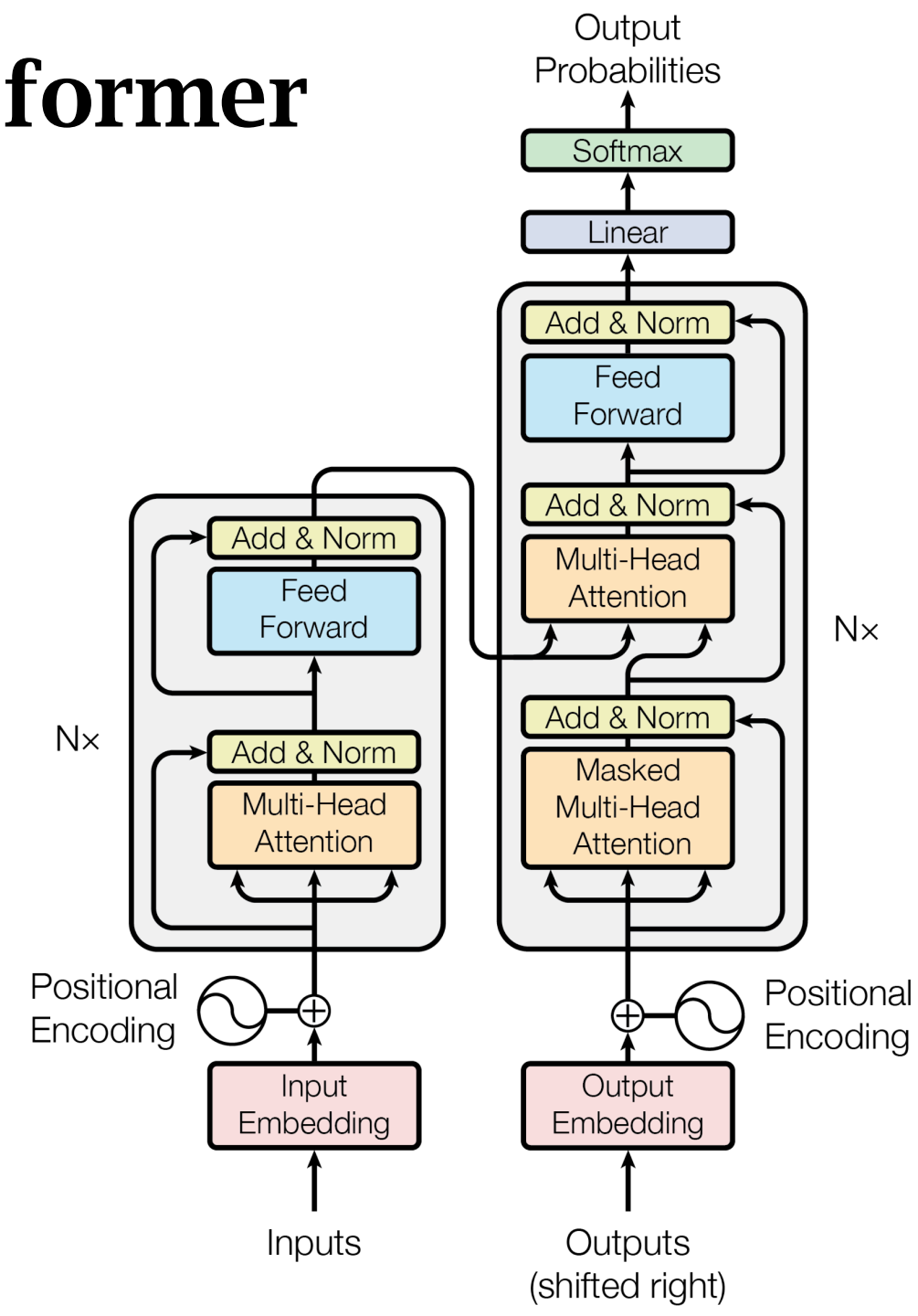
# Attention in Transformer

**Multi-head attention:**

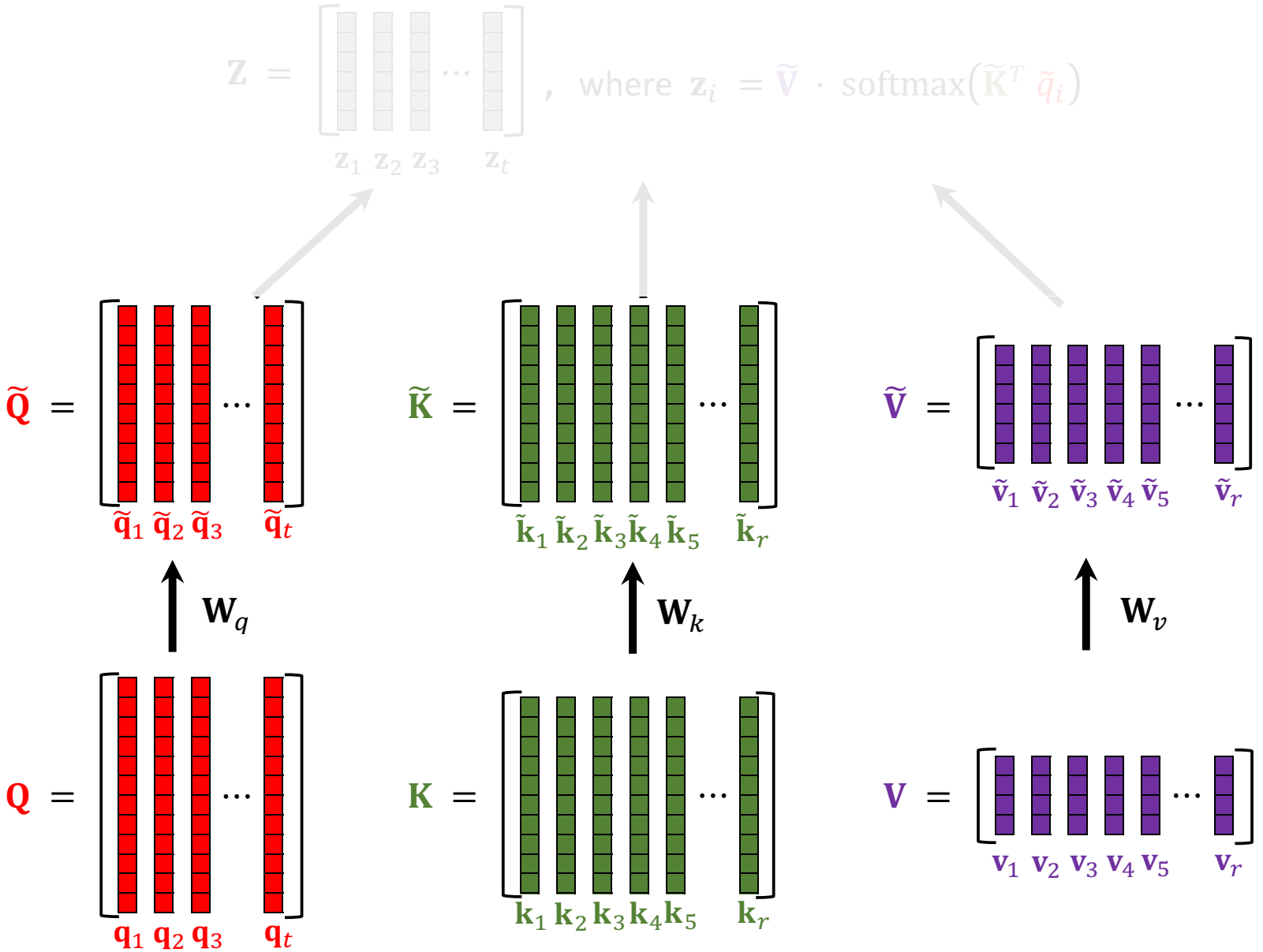
- Multiple single-head attentions, each has its own parameter matrices.
- Concatenate the outputs.

**Single-head attention:**

- $\mathbf{Z} = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$.
- $\mathbf{Q}$ and $\mathbf{Z}$ have $t$ columns.
- $t$: sequence length.
- $\mathbf{K}$ and $\mathbf{V}$ have $r$ columns ($r$ is arbitrary).



# Single-Head Attention: $\mathbf{Z} = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$.



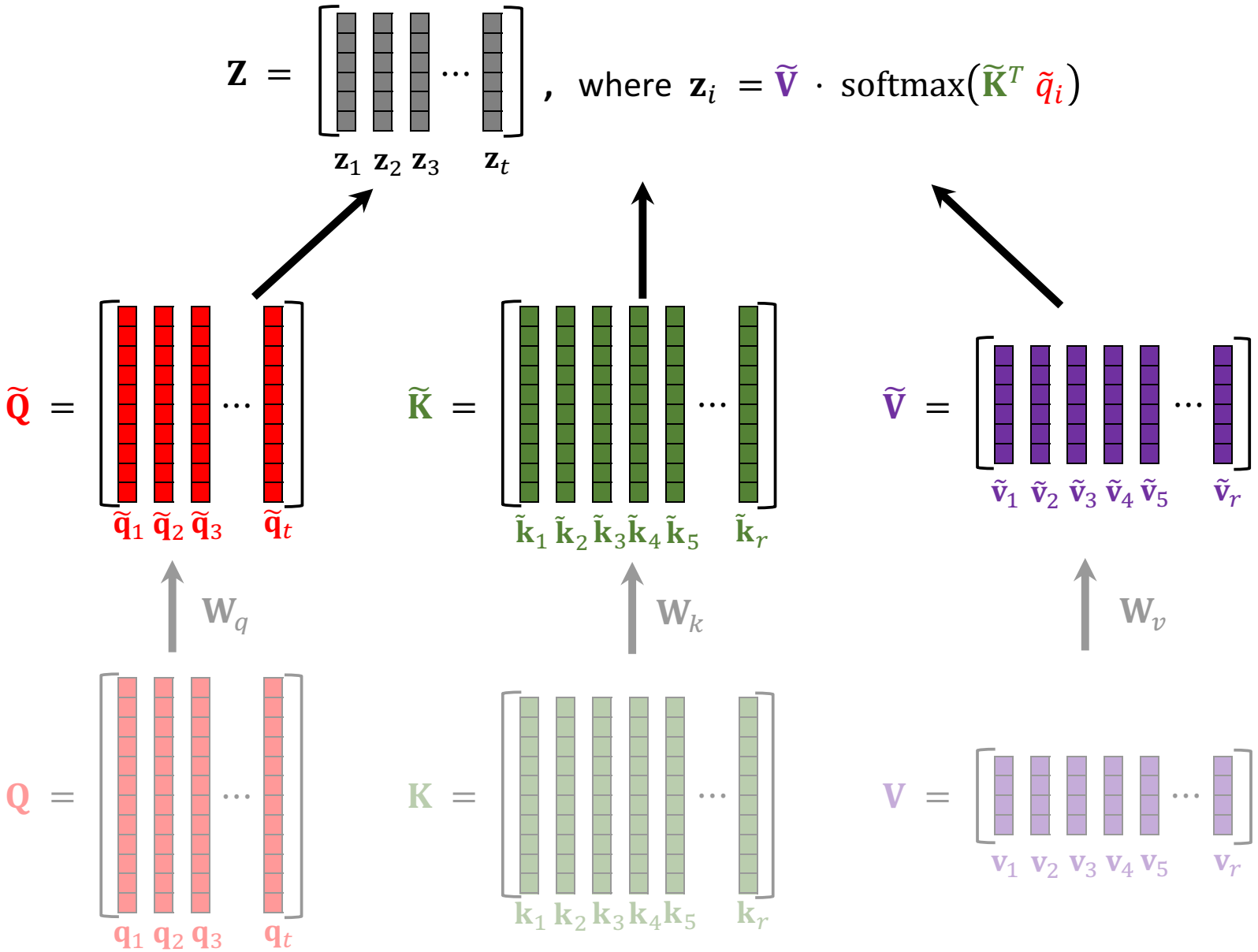

**Linear maps:**

- $\tilde{\mathbf{Q}} = \mathbf{W}_q \mathbf{Q}$,
- $\tilde{\mathbf{K}} = \mathbf{W}_k \mathbf{K}$,
- $\tilde{\mathbf{V}} = \mathbf{W}_v \mathbf{V}$ .

**Trainable parameters:**

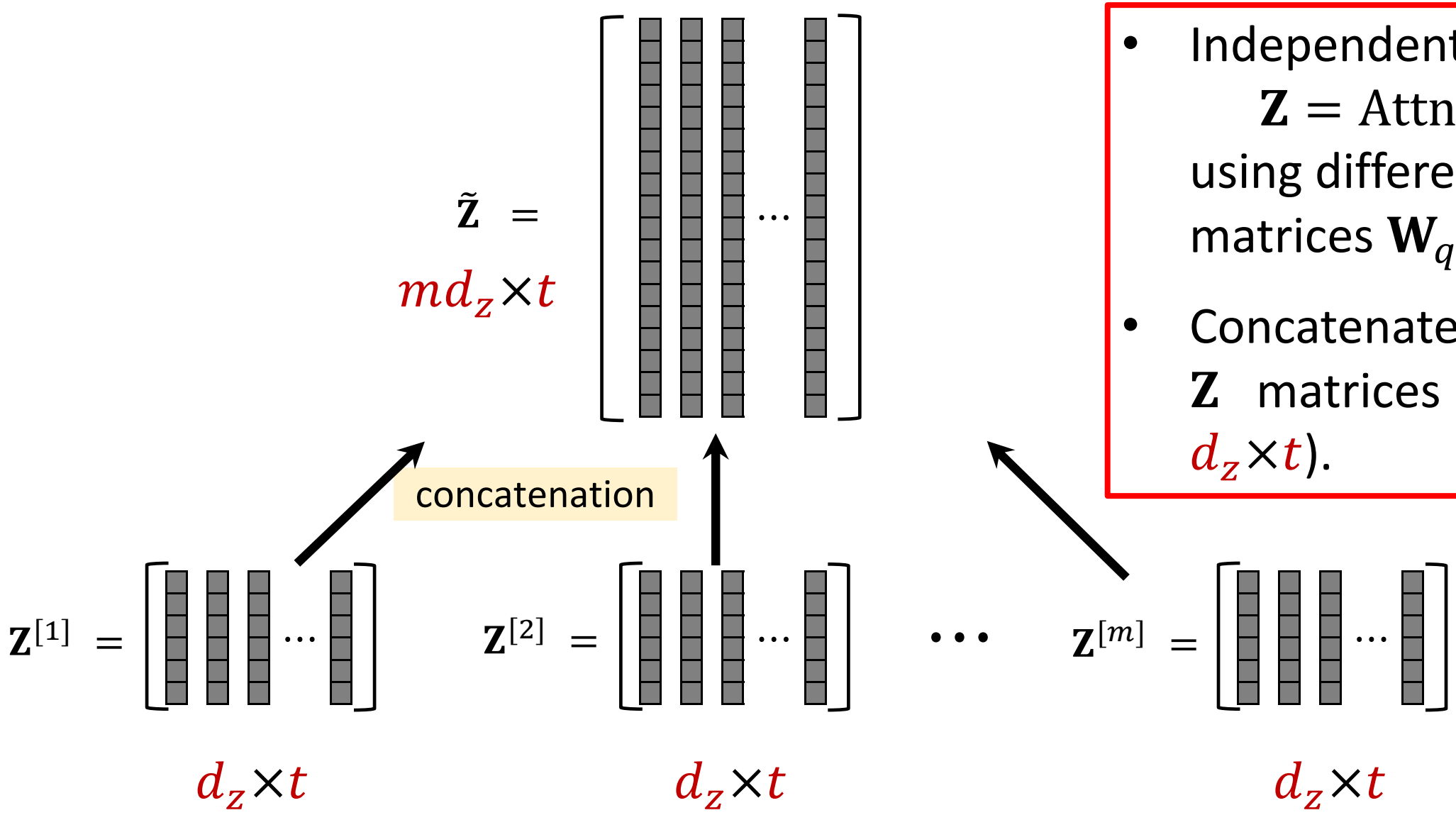
- $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$.

# Single-Head Attention: $\mathbf{Z} = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$.




- $\mathrm{rows}(\tilde{\mathbf{Q}}) = \mathrm{rows}(\tilde{\mathbf{K}})$
- $\mathrm{cols}(\tilde{\mathbf{K}}) = \mathrm{cols}(\tilde{\mathbf{V}})$
- $\mathrm{rows}(\mathbf{Z}) = \mathrm{rows}(\tilde{\mathbf{V}})$
- $\mathrm{cols}(\mathbf{Z}) = \mathrm{cols}(\tilde{\mathbf{Q}})$

# Multi-Head Attention

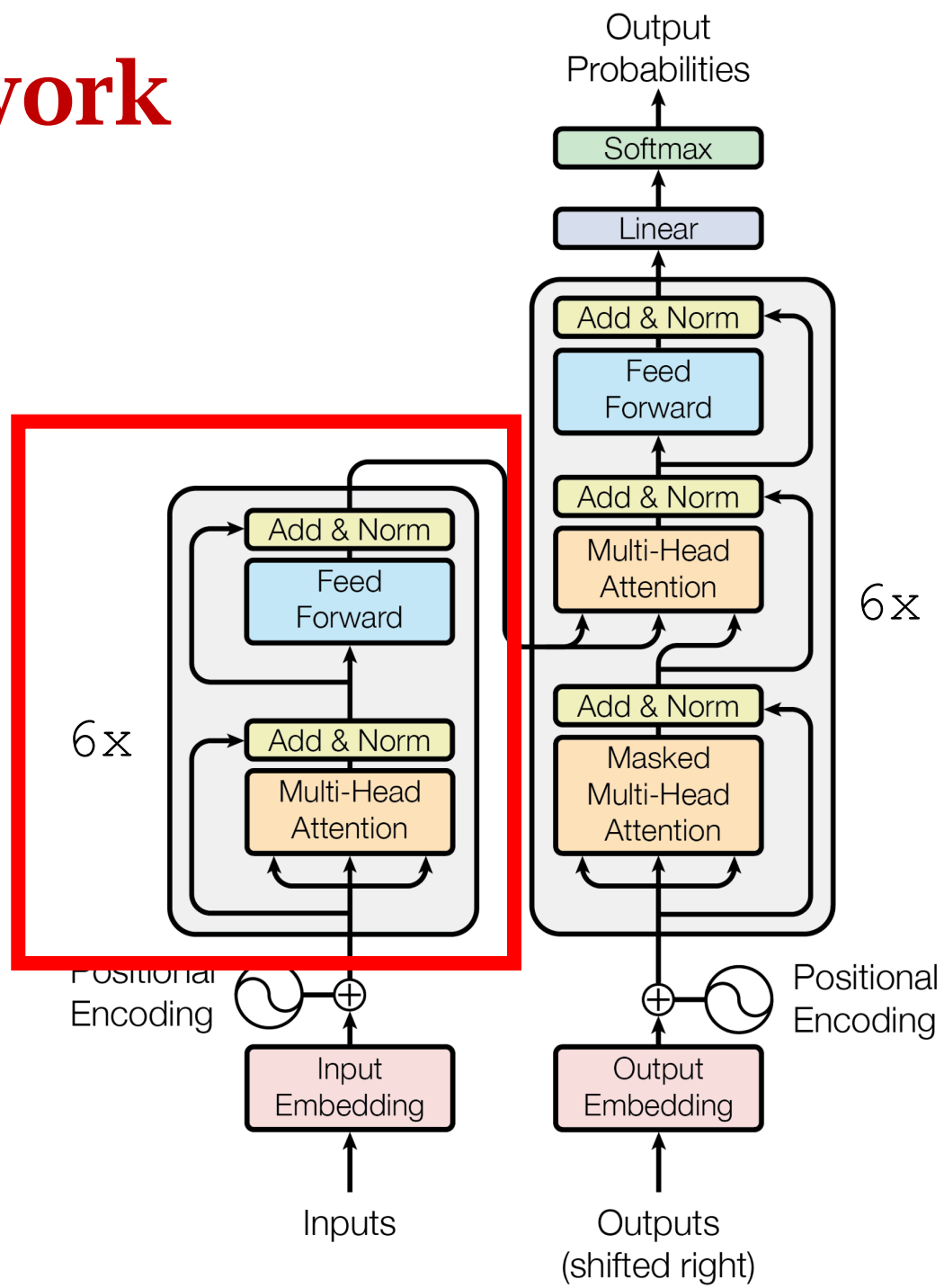



- Independently perform $\mathbf{Z} = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ using different parameter matrices $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$.
- Concatenate the obtained $\mathbf{Z}$ matrices (which are $d_z \times t$).

# Encoder of Transformer

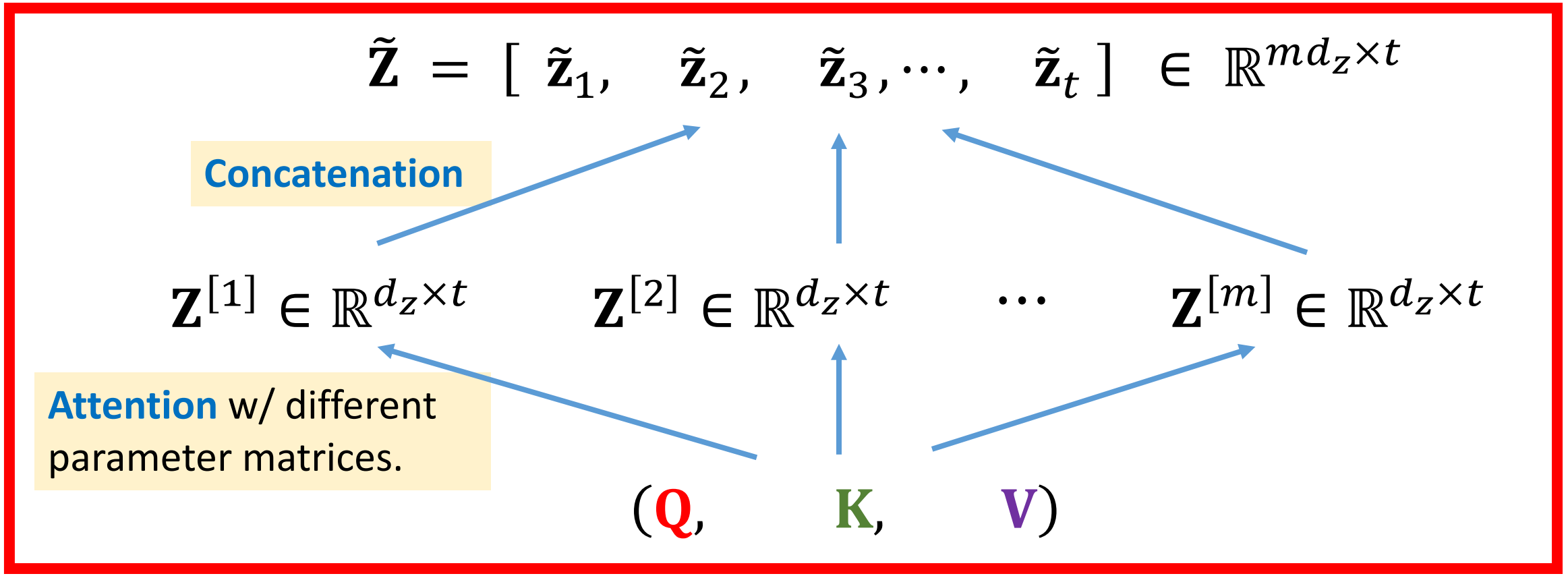
# Encoder Network

- Encoder has 6 blocks.
- 1 Block = Multi-head attention + Dense.
- 6 is the result of hyper-parameter tuning; nothing magical about 6.
- Other tricks:
  - Skip connection.
  - Normalization.



# Multi-Head Attention + Dense Layer

## Multi-Head Attention



# Multi-Head Attention + Dense Layer

$$\tilde{\mathbf{U}} = [\ \tilde{\mathbf{u}}_1,\ \tilde{\mathbf{u}}_2,\ \tilde{\mathbf{u}}_3, \cdots,\ \tilde{\mathbf{u}}_t\ ] \in \mathbb{R}^{d_u \times t}$$

**Dense layer** is applied to every column independently.

$$\tilde{\mathbf{Z}} = [\ \tilde{\mathbf{z}}_1,\ \tilde{\mathbf{z}}_2,\ \tilde{\mathbf{z}}_3, \cdots,\ \tilde{\mathbf{z}}_t\ ] \in \mathbb{R}^{md_z \times t}$$

Concatenation

$\mathbf{Z}^{[1]} \in \mathbb{R}^{d_z \times t}$ $\mathbf{Z}^{[2]} \in \mathbb{R}^{d_z \times t}$ $\cdots$ $\mathbf{Z}^{[m]} \in \mathbb{R}^{d_z \times t}$

Attention w/ different parameter matrices.

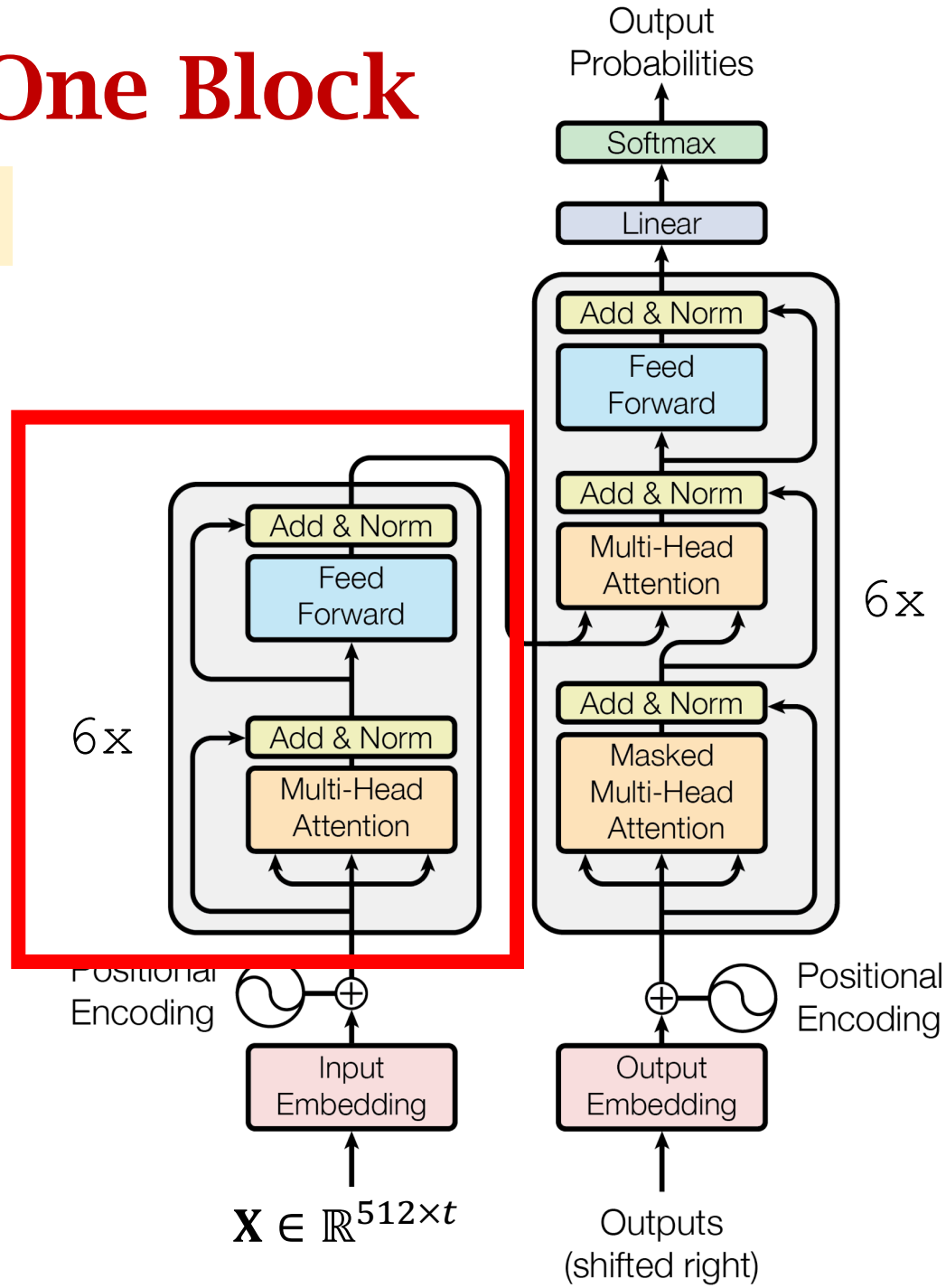
$(\mathbf{Q}, \mathbf{K}, \mathbf{V})$

# Encoder Network: One Block

Ignore skip connection and normalization.

- Input: $\mathbf{X} \in \mathbb{R}^{512 \times t}$; ($t$ is the seq length.)
- Set $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$.

query key value



# Encoder Network: One Block

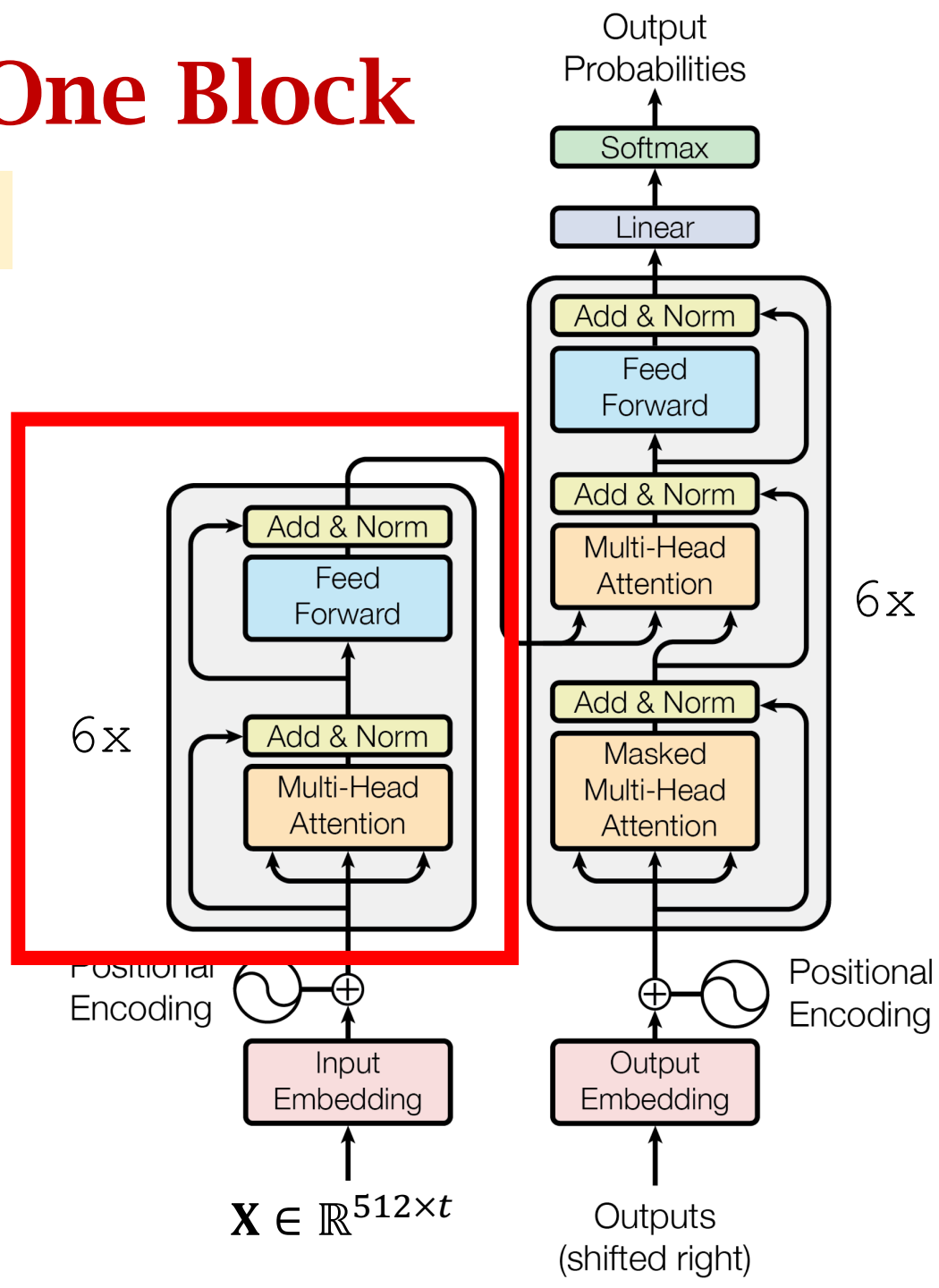
Ignore skip connection and normalization.

- Input: $\mathbf{X} \in \mathbb{R}^{512\times t}$; ($t$ is the seq length.)
- Set $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$.
- Repeat $m = 8$ times:

$$\mathbf{Z}^{[i]} = \text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{64\times t}.$$

- $\tilde{\mathbf{Z}} = \text{Concatenate}\left(\mathbf{Z}^{[1]}, \cdots, \mathbf{Z}^{[m]}\right) \in \mathbb{R}^{512\times t}$.

- Make sure the input shape and output shape are the same.
- Otherwise, skip connection cannot be applied.



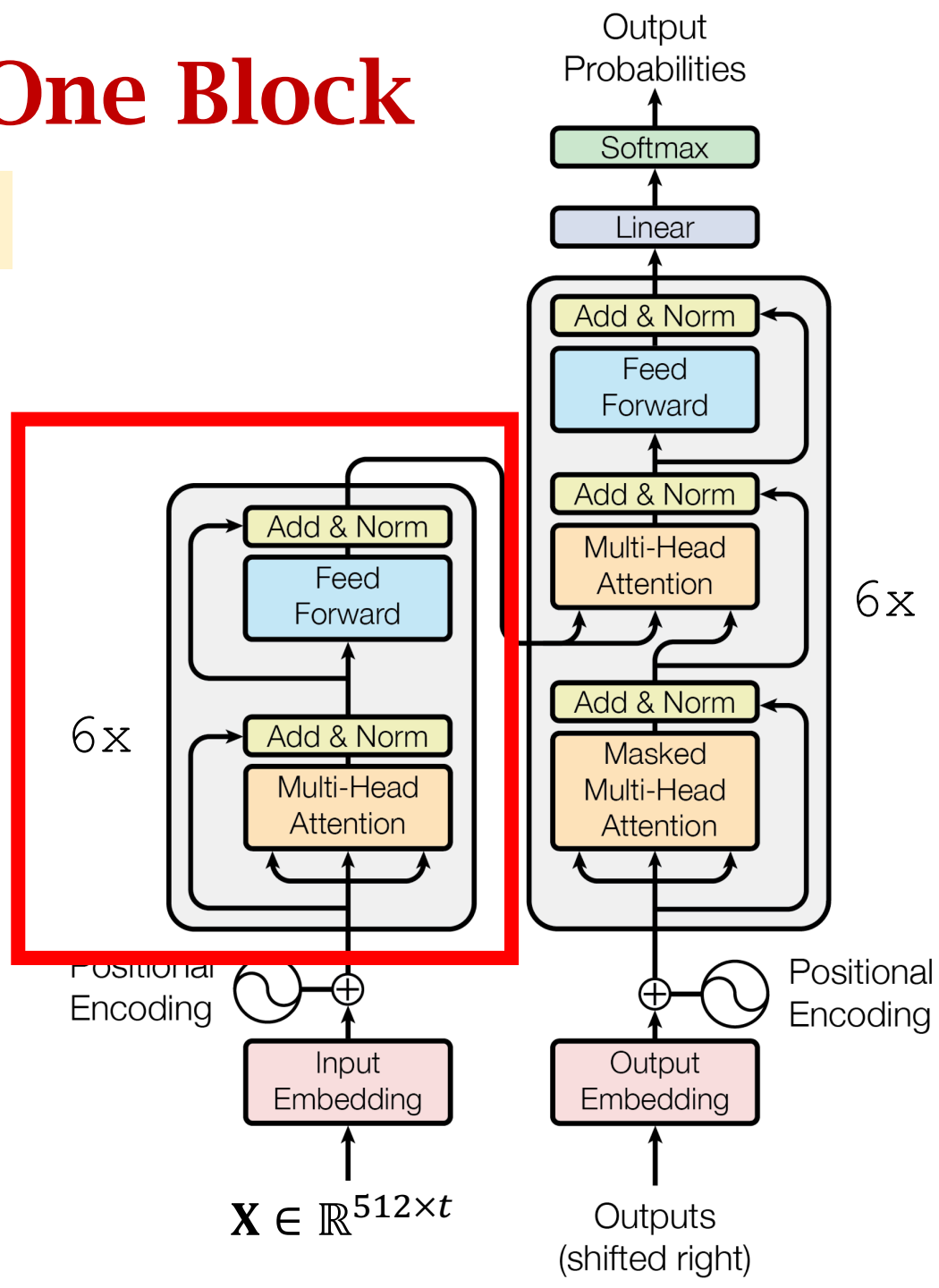
# Encoder Network: One Block

Ignore skip connection and normalization.

- Input: $\mathbf{X} \in \mathbb{R}^{512\times t}$; ($t$ is the seq length.)
- Set $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$.
- Repeat $m = 8$ times:

$$\mathbf{Z}^{[i]} = \mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{64\times t}.$$

- $\tilde{\mathbf{Z}} = \mathrm{Concatenate}\left(\mathbf{Z}^{[1]}, \cdots, \mathbf{Z}^{[m]}\right) \in \mathbb{R}^{512\times t}$.
- $\mathbf{U} = \mathrm{DenseLayer}\left(\tilde{\mathbf{Z}}\right) \in \mathbb{R}^{512\times t}$.
- Output: $\mathbf{U} \in \mathbb{R}^{512\times t}$. (The same shape as $\mathbf{X}$.)

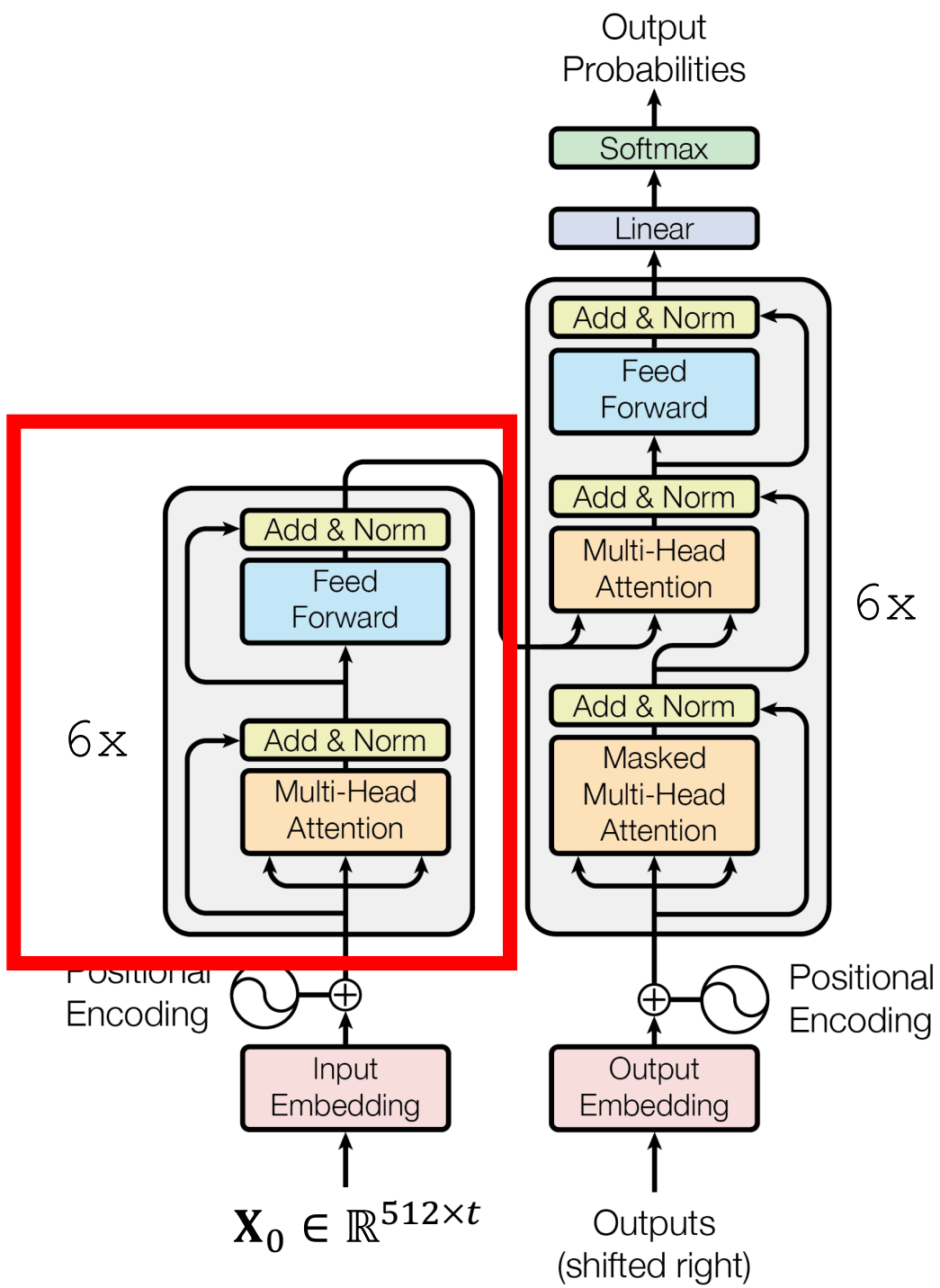


# Encoder Network

$\mathbf{X}_{(1)} \in \mathbb{R}^{512 \times t}$

**Block 1**

$\mathbf{X}_{(0)} \in \mathbb{R}^{512 \times t}$



# Encoder Network

**Encoder**

$\mathbf{X}_{(6)} \in \mathbb{R}^{512 \times t}$

**Block 6**

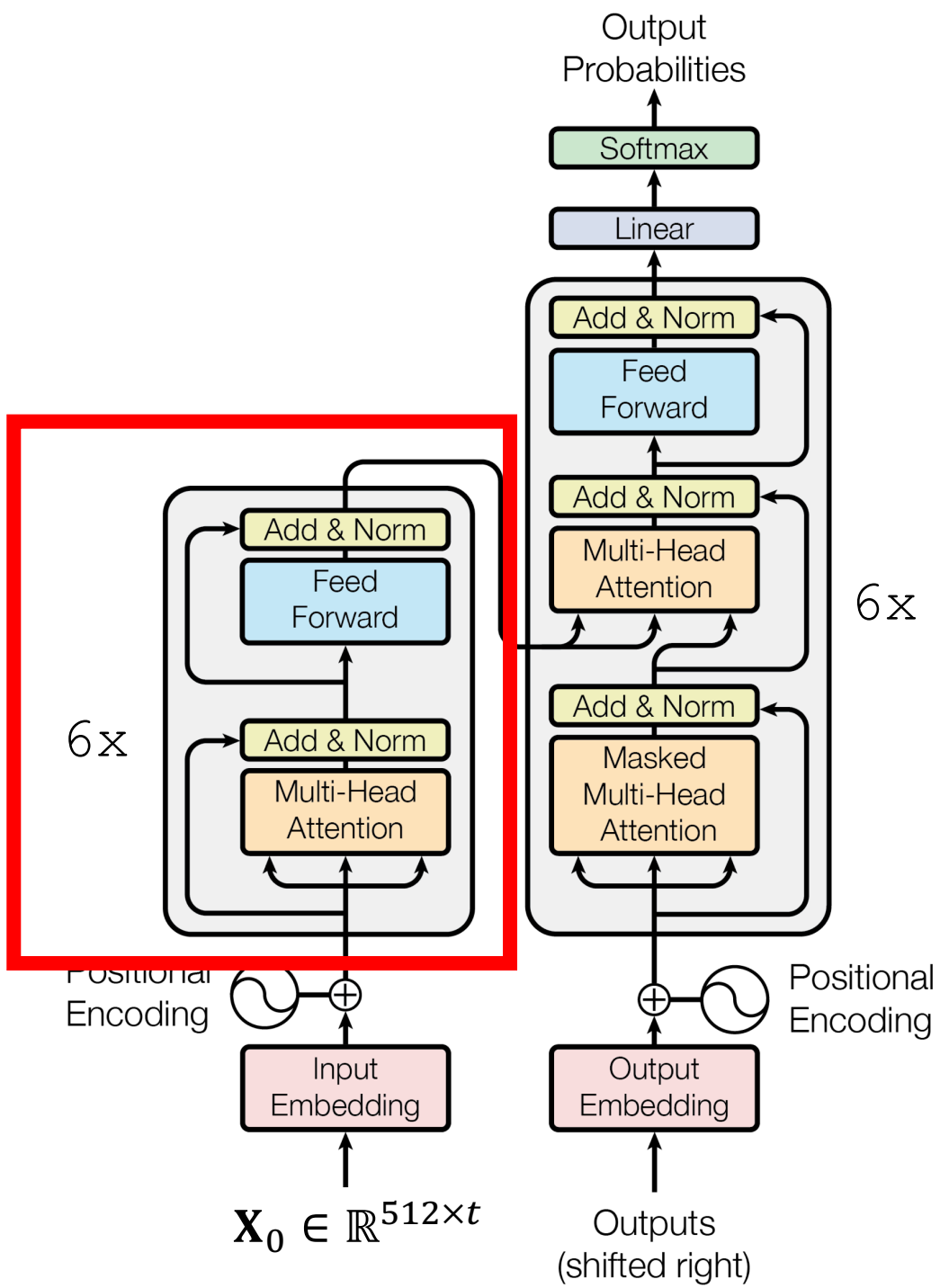
⋮

**Block 2**

$\mathbf{X}_{(1)} \in \mathbb{R}^{512 \times t}$

**Block 1**

$\mathbf{X}_{(0)} \in \mathbb{R}^{512 \times t}$



# Decoder of Transformer

# Decoder Network: One Block

Encoder

$\mathbf{X}_{(6)} \in \mathbb{R}^{512 \times t}$

Block 6

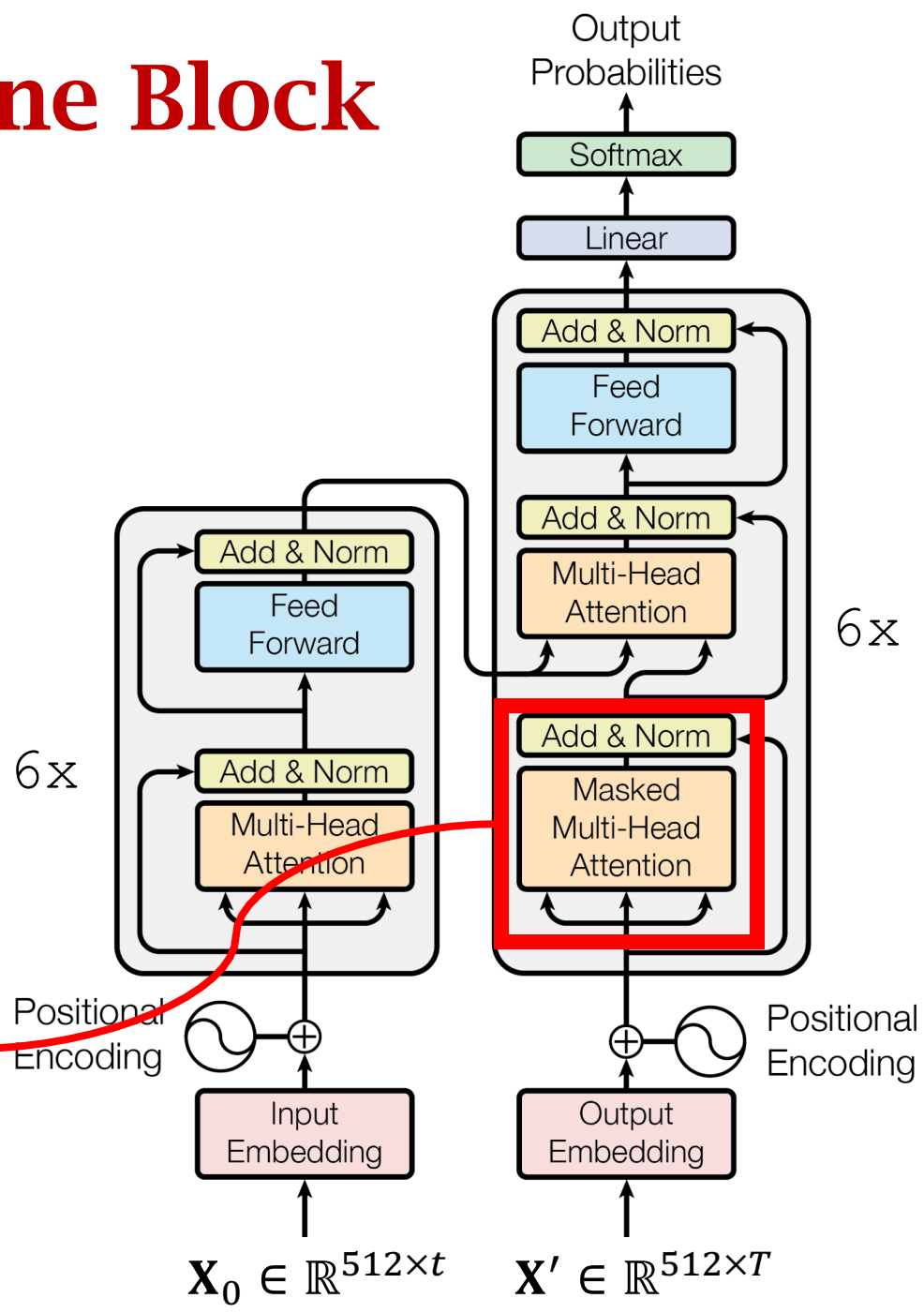
Block 2

$\mathbf{X}_{(1)} \in \mathbb{R}^{512 \times t}$

Block 1

$\mathbf{X}_{(0)} \in \mathbb{R}^{512 \times t}$

- Similar to encoder.
- Set $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}'$.



# Decoder Network: One Block

Encoder

$\mathbf{X}_{(6)} \in \mathbb{R}^{512\times t}$

Block 6

Block 2

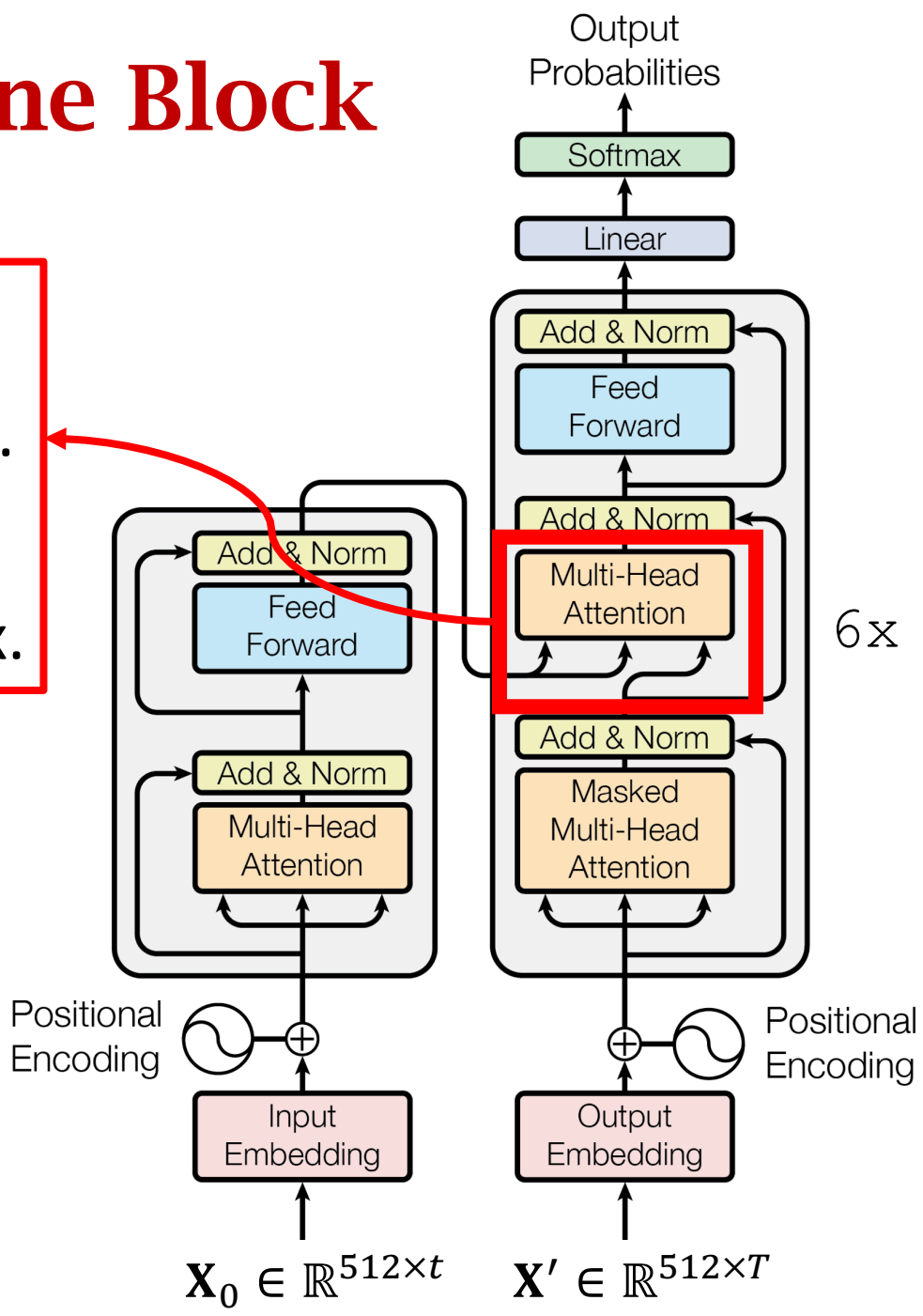
$\mathbf{X}_{(1)} \in \mathbb{R}^{512\times t}$

Block 1

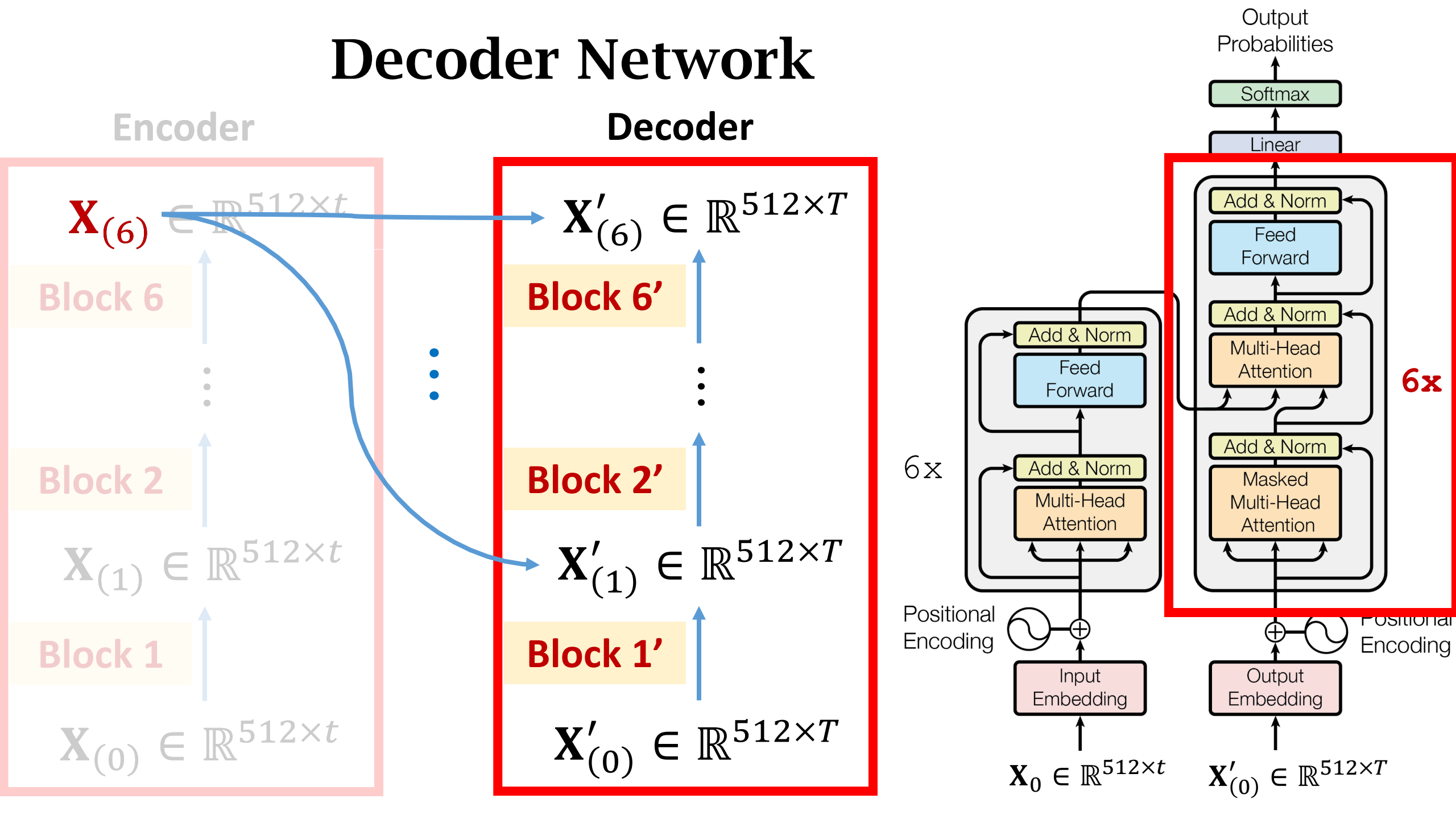
$\mathbf{X}_{(0)} \in \mathbb{R}^{512\times t}$

- Set $\mathbf{Q} = \mathbf{X}' \in \mathbb{R}^{512\times T}$.
- $\mathbf{K} = \mathbf{V} = \mathbf{X}_{(6)} \in \mathbb{R}^{512\times t}$.
- Multi-head attention outputs a $512\times T$ matrix.

- Similar to encoder.
- Set $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}'$.

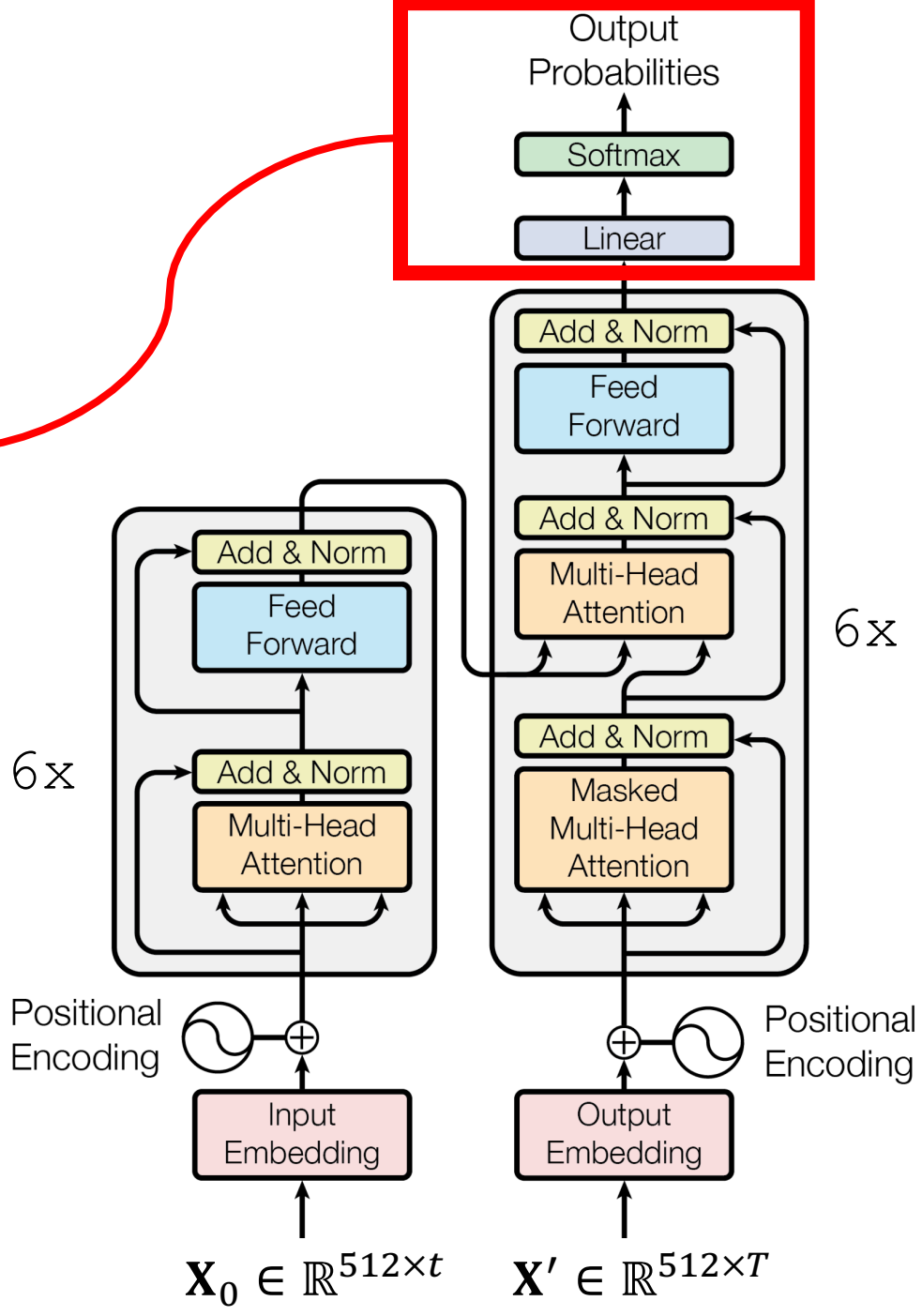


# Decoder Network



# Decoder Network

- Output a distribution over the vocabulary.
- Sample the next word according to the distribution.
- Append the new word's embedding to $\mathbf{X}'$.
- Run the decoder again, taking $\mathbf{X}' \in \mathbb{R}^{512\times(T+1)}$ as input.



# Summary

# Summary

- Transformer model is not RNN.
- Transformer is based on attention and self-attention.
- Upside: Outperform all the state-of-the-art RNN models.
- Downside: Much more expensive than RNN models.

- Read the original paper: Vaswani et al. Attention Is All You Need. In *NIPS*, 2017.
- Google "*transformer model explained*" and read the articles.

# Key Concept: Multi-Head Attention

- Inputs: query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$.
- Linear maps: $\tilde{\mathbf{Q}} = \mathbf{W}_q \mathbf{Q}$, $\tilde{\mathbf{K}} = \mathbf{W}_k \mathbf{K}$, and $\tilde{\mathbf{V}} = \mathbf{W}_v \mathbf{V}$.
- Single-head attention:

$$\mathbf{Z} = \text{attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \tilde{\mathbf{V}} \cdot \text{softmax}\left(\tilde{\mathbf{K}}^T \tilde{\mathbf{Q}}\right).$$

# Key Concept: Multi-Head Attention

- Inputs: query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$.
- Linear maps: $\tilde{\mathbf{Q}} = \mathbf{W}_q \mathbf{Q}$, $\tilde{\mathbf{K}} = \mathbf{W}_k \mathbf{K}$, and $\tilde{\mathbf{V}} = \mathbf{W}_v \mathbf{V}$.
- Single-head attention:

$$\mathbf{Z} = \text{attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \tilde{\mathbf{V}} \cdot \text{softmax}\left(\tilde{\mathbf{K}}^T \tilde{\mathbf{Q}}\right).$$

- Multi-head attention:
  - Repeat $\text{attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ using different parameters $\mathbf{W}_q$, $\mathbf{W}_v$, $\mathbf{W}_v$.
  - Get $\mathbf{Z}^{[1]}, \mathbf{Z}^{[2]}, \cdots, \mathbf{Z}^{[m]} \in \mathbb{R}^{d_z \times t}$.
  - Concatenate the $m$ matrices to get $\tilde{\mathbf{Z}} \in \mathbb{R}^{m d_z \times t}$.

**Attention in the encoder:**

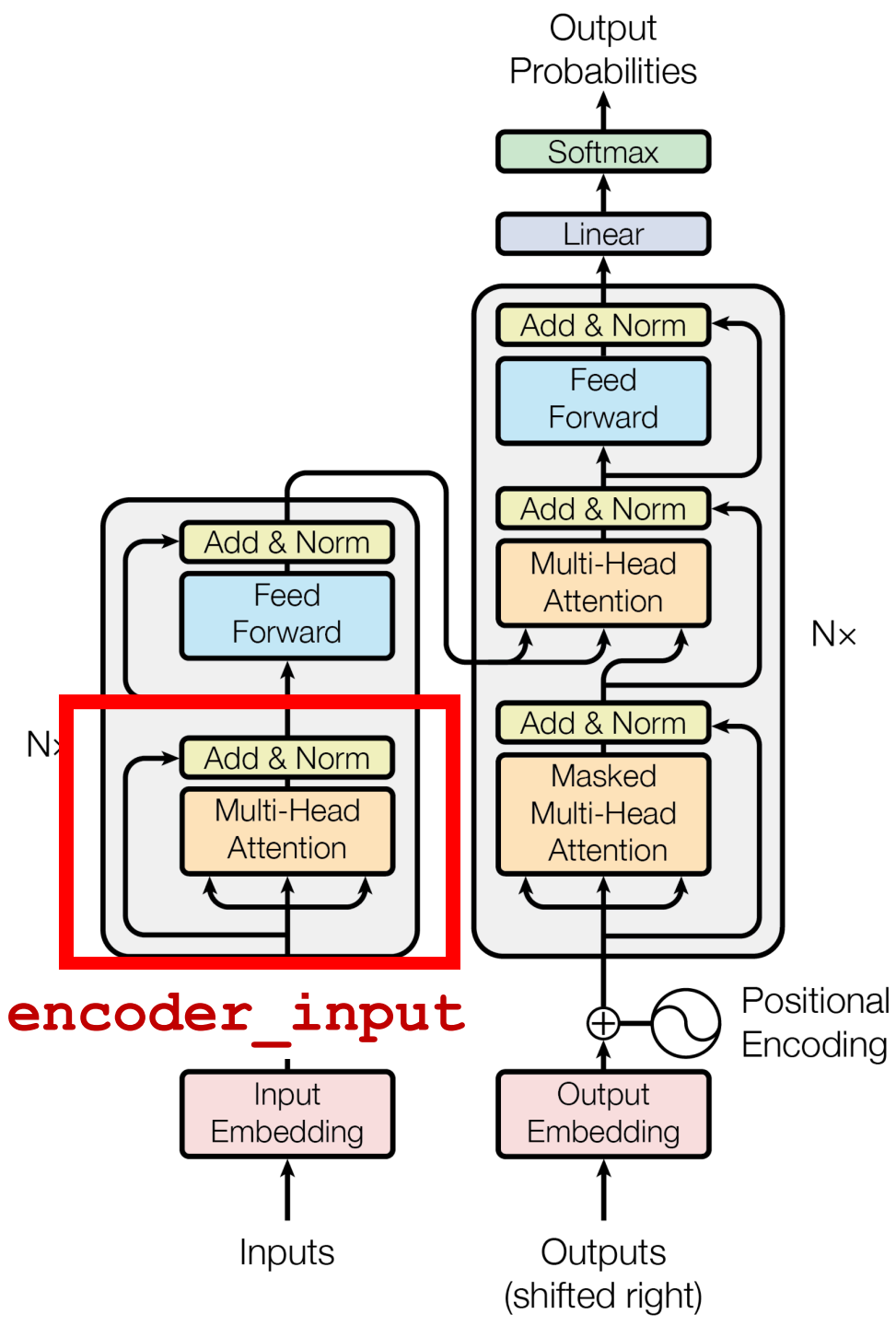
- $\mathbf{Q} = \mathbf{K} = \mathbf{V} =$ `encoder_input`.

**1st attention in the decoder:**

- $\mathbf{Q} = \mathbf{K} = \mathbf{V} =$ `decoder_input`.

**2nd attention in the decoder**

- $\mathbf{Q} =$ `decoder_input`
- $\mathbf{K} = \mathbf{V} =$ `encoder_output`.



Attention in the encoder:

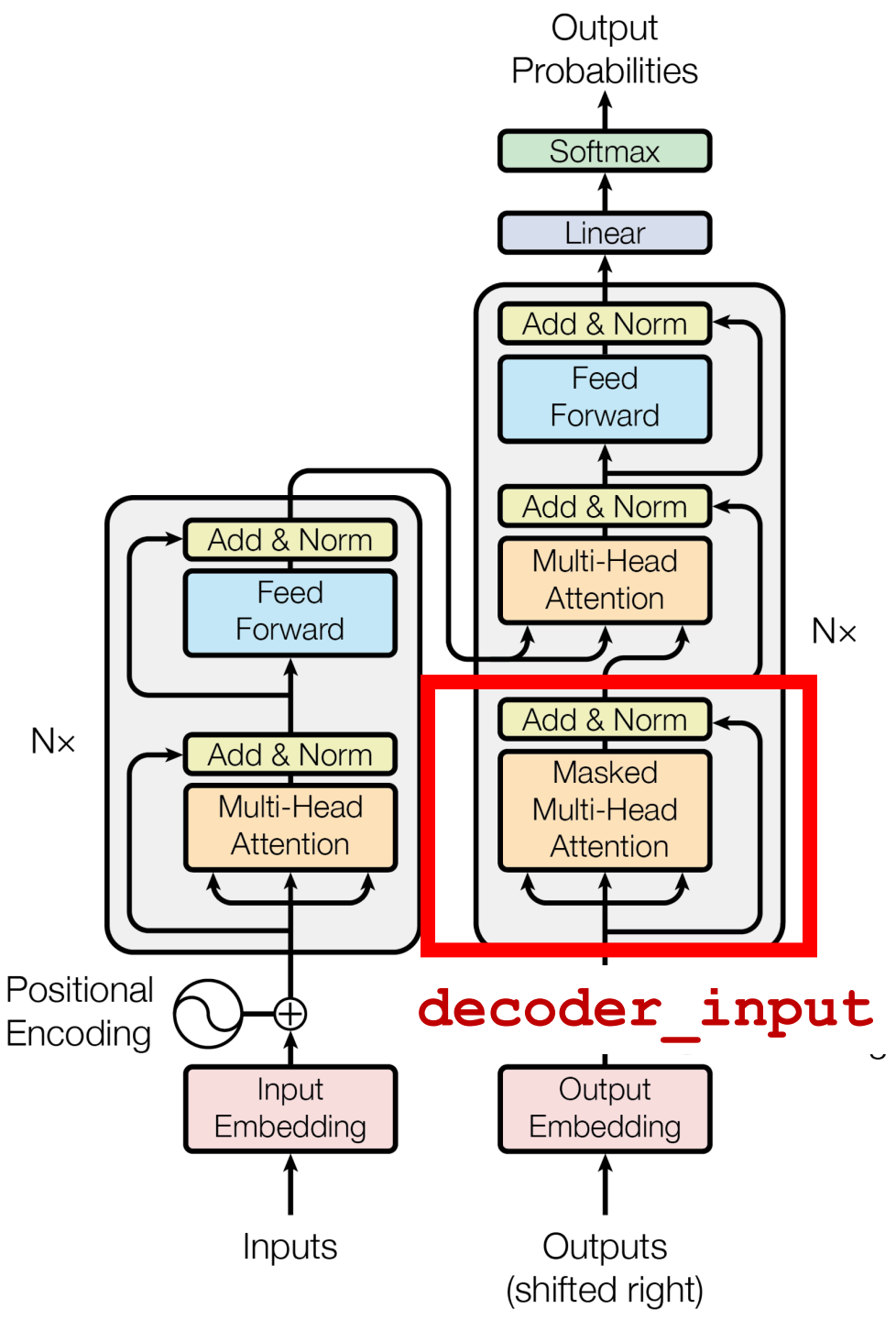
- $Q = K = V =$ `encoder_input`.

**1st attention in the decoder:**

- $Q = K = V =$ `decoder_input`.

2nd attention in the decoder

- $Q =$ `decoder_input`
- $K = V =$ `encoder_output`.



Attention in the encoder:

- $Q = K = V =$ `encoder_input`.

1st attention in the decoder:

- $Q = K = V =$ `decoder_input`.

**2nd attention in the decoder**

- $Q =$ `decoder_input`
- $K = V =$ `encoder_output`.