Attention is all you need

Author: Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I..

Publish: *Advances in Neural Information Processing Systems*
Pp: 5998 - 6008.

Presenter: WENWEI KANG

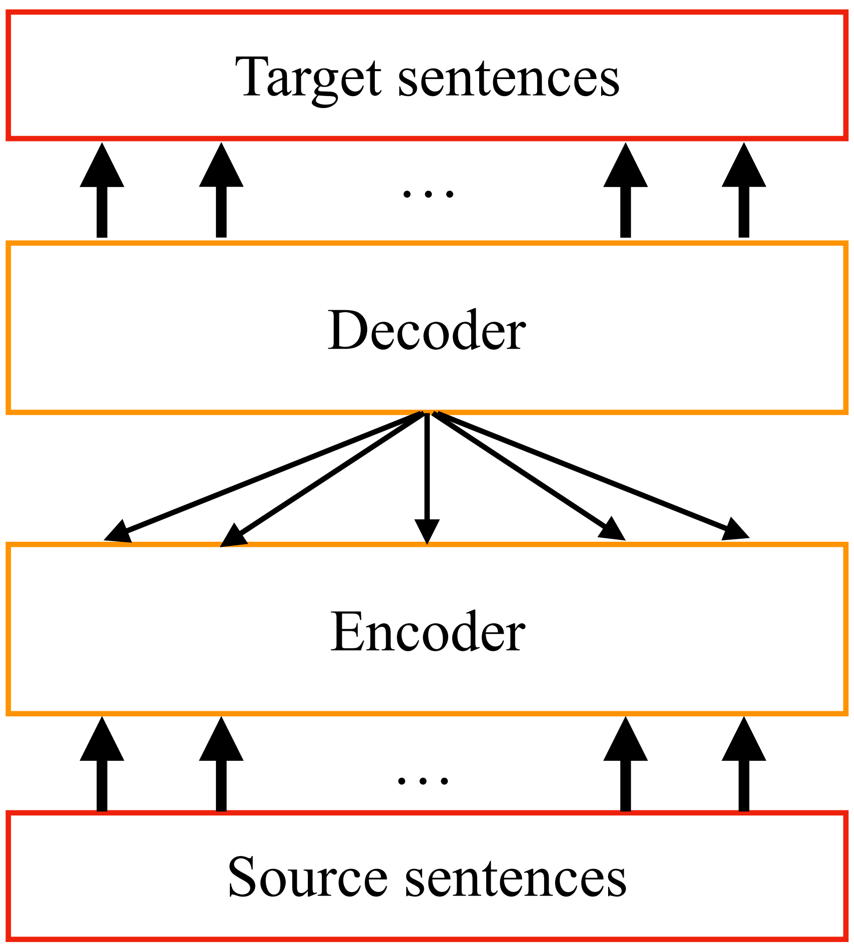
https://arxiv.org/abs/1706.03762

- Introduction
- Encoder - Decoder
- Mechanism
- Evaluation

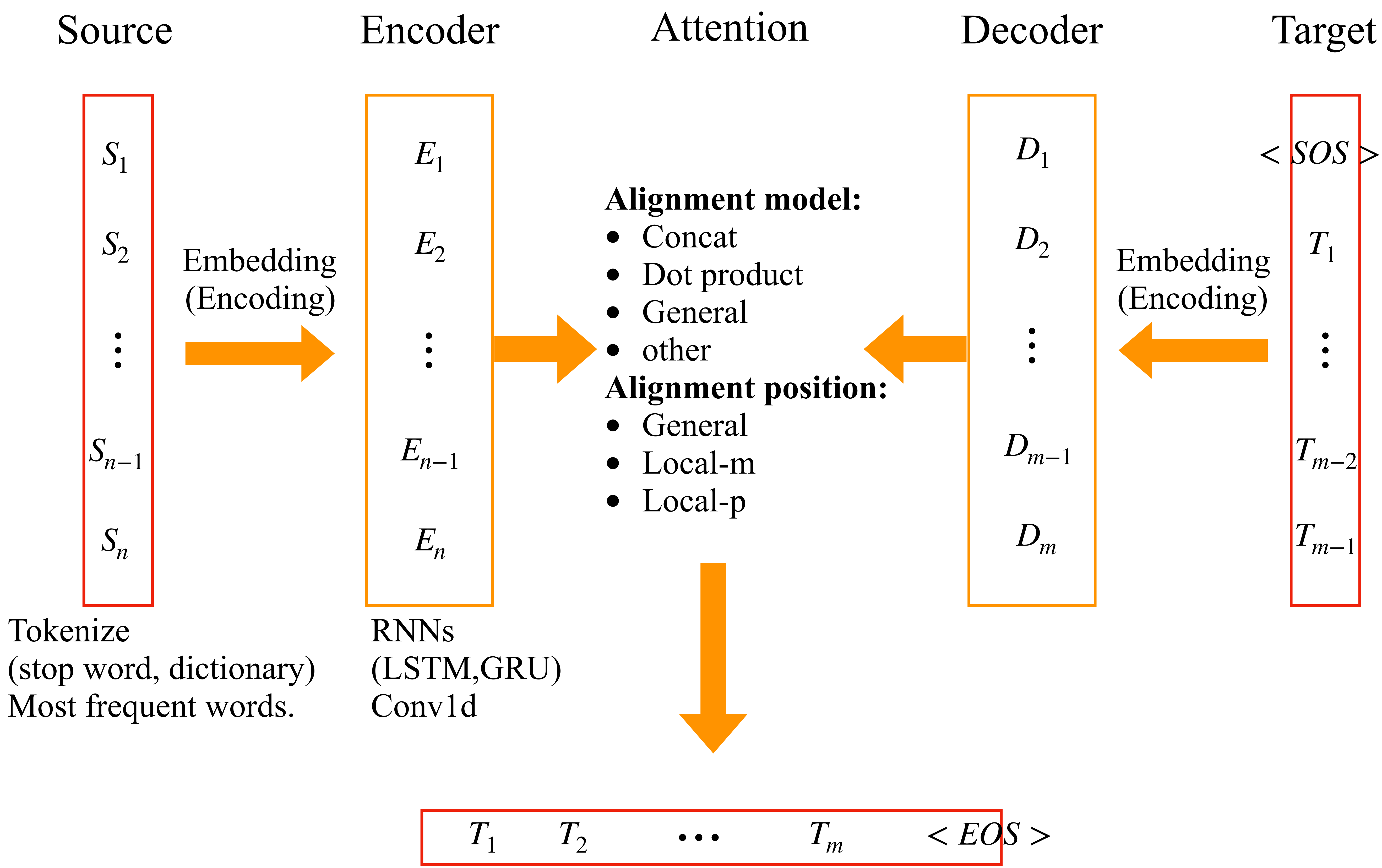
Introduction

Neural Machine Translation(NMT):

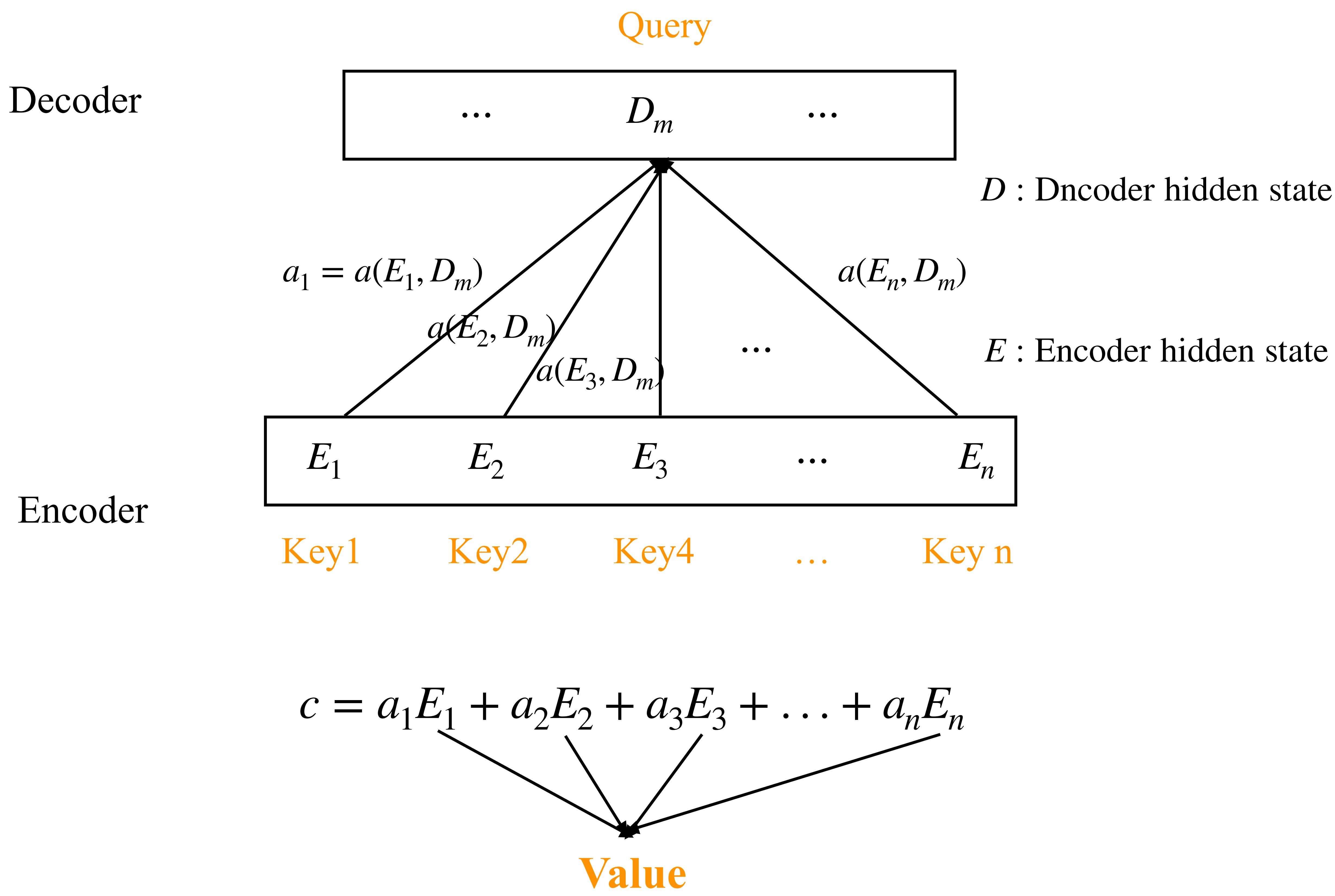
- **Statistical based:** Phrase-based + large LM (Moses)
- **NN based:** Encoder - Decoder (Seq2seq, ConvS2S, ensemble …)



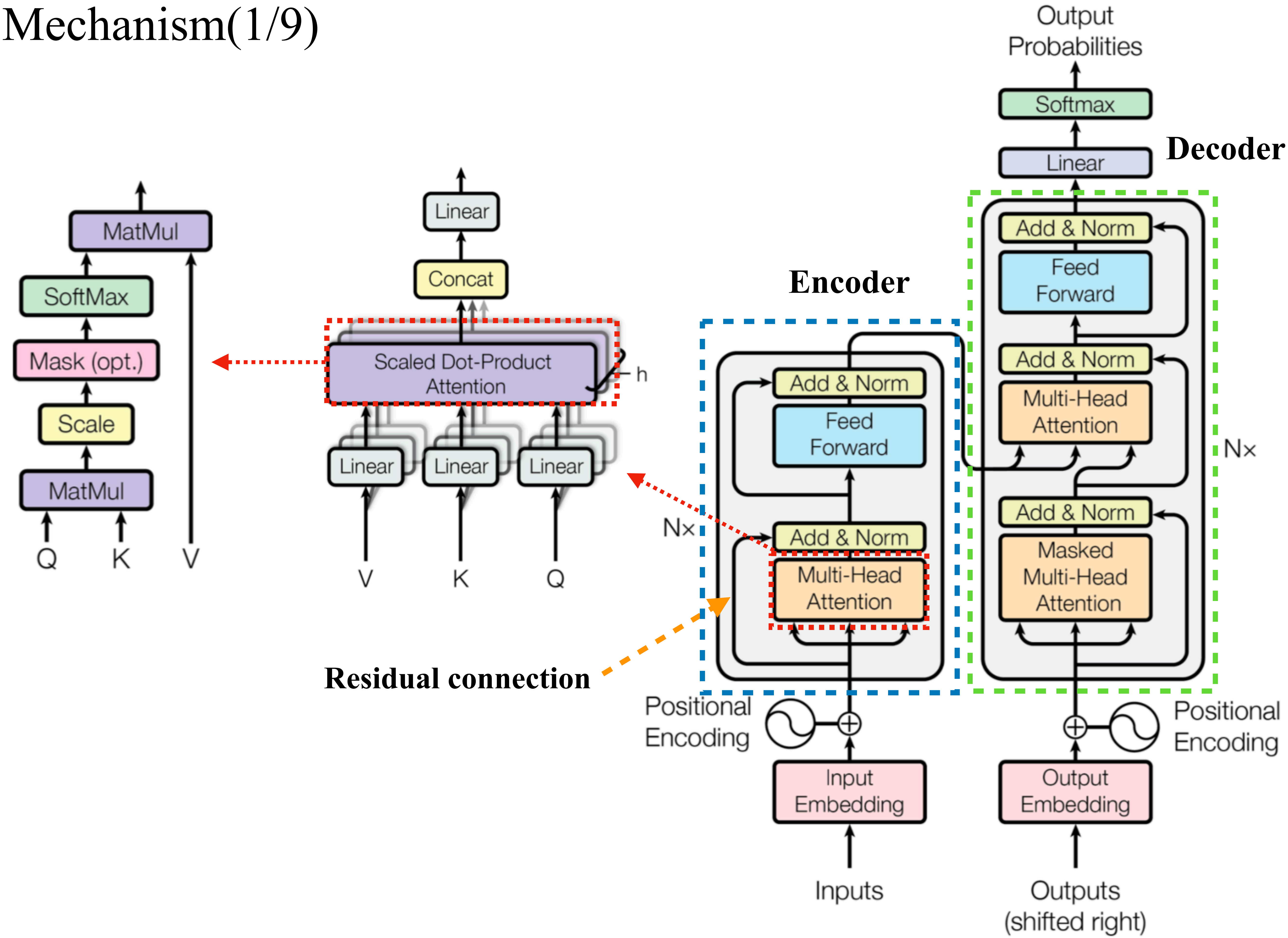
Encoder - Decoder(1/2)



Encoder - Decoder(1/2)

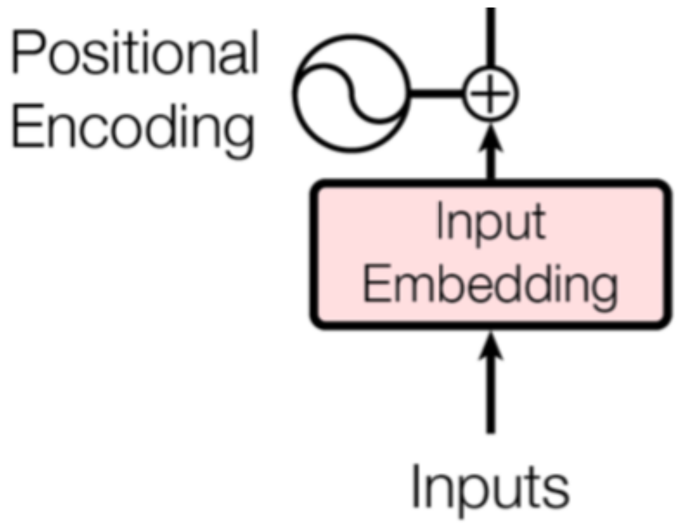


Mechanism(1/9)



Mechanism(2/9)

$$\underbrace{\underset{\text{(Query)}}{\mathbf{Q}} \quad \underset{\text{(Key)}}{\mathbf{K}} \quad \underset{\text{(Value)}}{\mathbf{V}}}$$



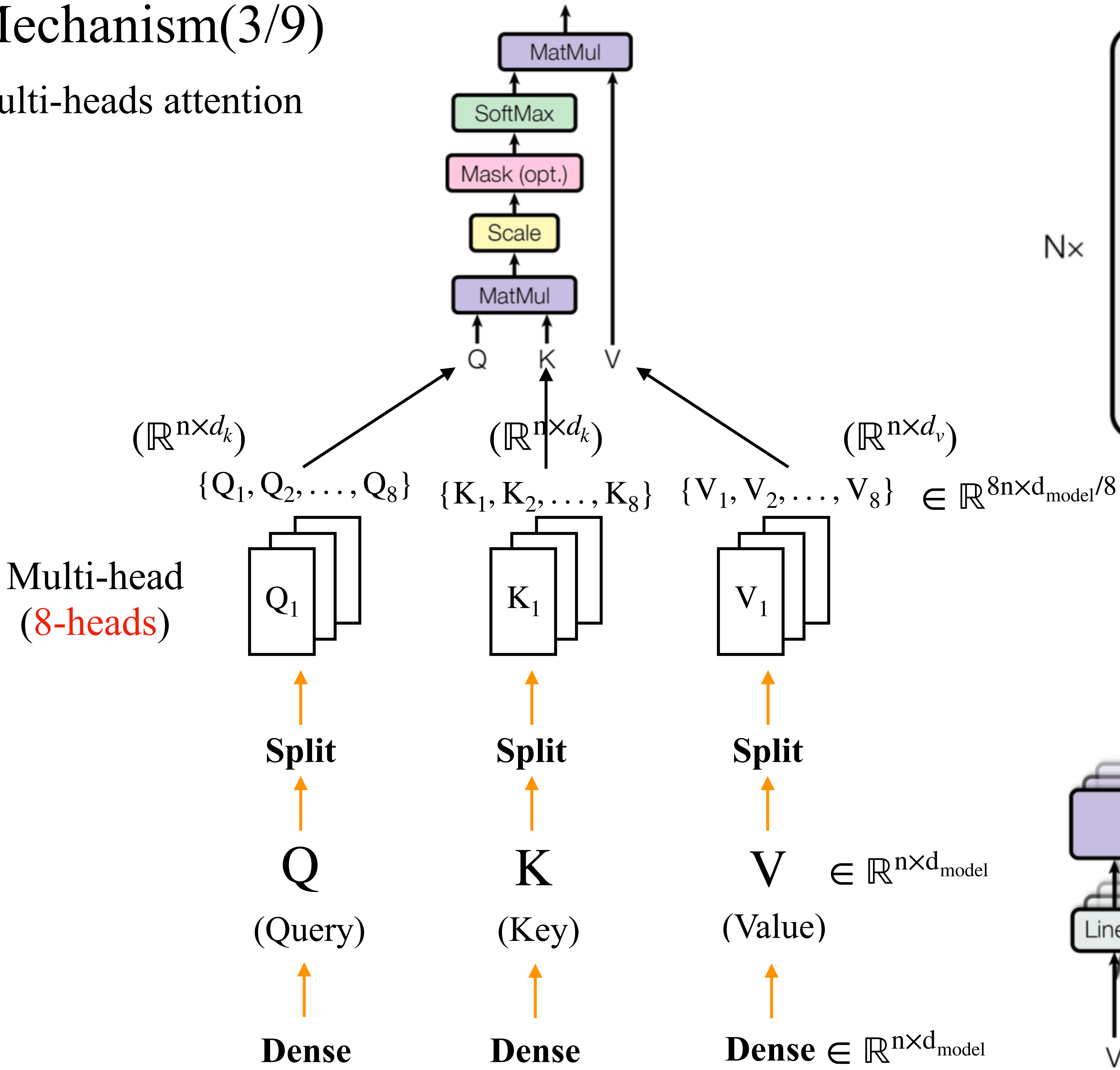
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

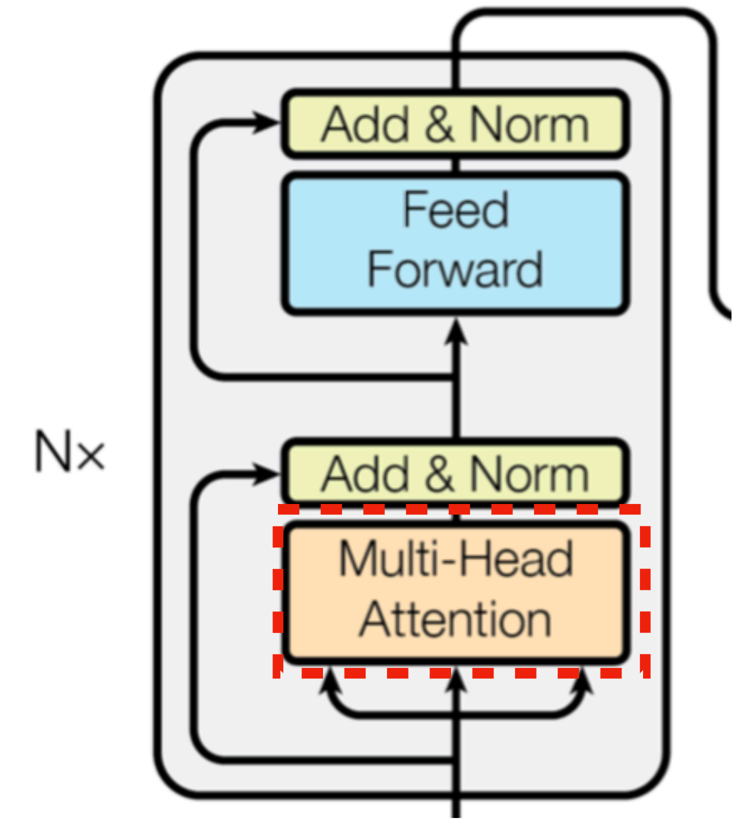
$$\begin{bmatrix} 0 \\ 1 \\ \vdots \\ n-2 \\ n-1 \end{bmatrix} \Rightarrow \begin{array}{cccc} \text{sin} & \text{cos} & & \text{sin} \\ PE_{0,0} & PE_{0,1} & \cdots & PE_{0,d_{model}-1} \\ PE_{1,0} & PE_{1,1} & \cdots & PE_{1,d_{model}-1} \\ \vdots & \vdots & \vdots & \vdots \\ PE_{n-2,0} & PE_{n-2,1} & \cdots & PE_{n-2,d_{model}-1} \\ PE_{n-1,0} & PE_{n-1,1} & \cdots & PE_{n-1,d_{model}-1} \end{array} \quad + \quad \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,d_{model}} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,d_{model}} \\ \vdots & \vdots & \vdots & \vdots \\ S_{n-1,1} & S_{n-1,2} & \cdots & S_{n-1,d_{model}} \\ S_{n,1} & S_{n,2} & \cdots & S_{n,d_{model}} \end{bmatrix} \Leftarrow \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{n-1} \\ S_n \end{bmatrix}$$

Positional Encoding **Embedding**

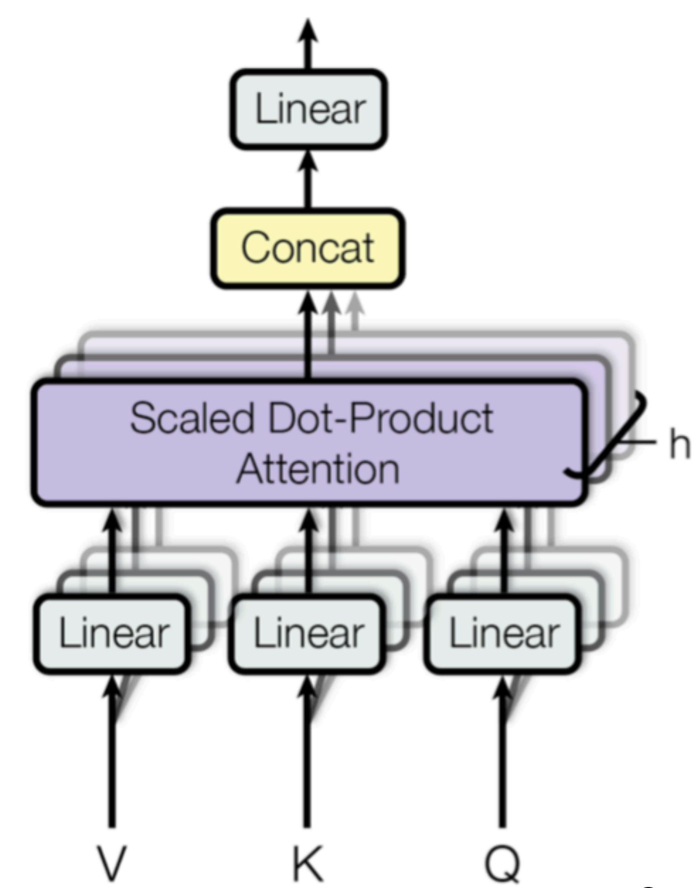
Mechanism(3/9)

Multi-heads attention





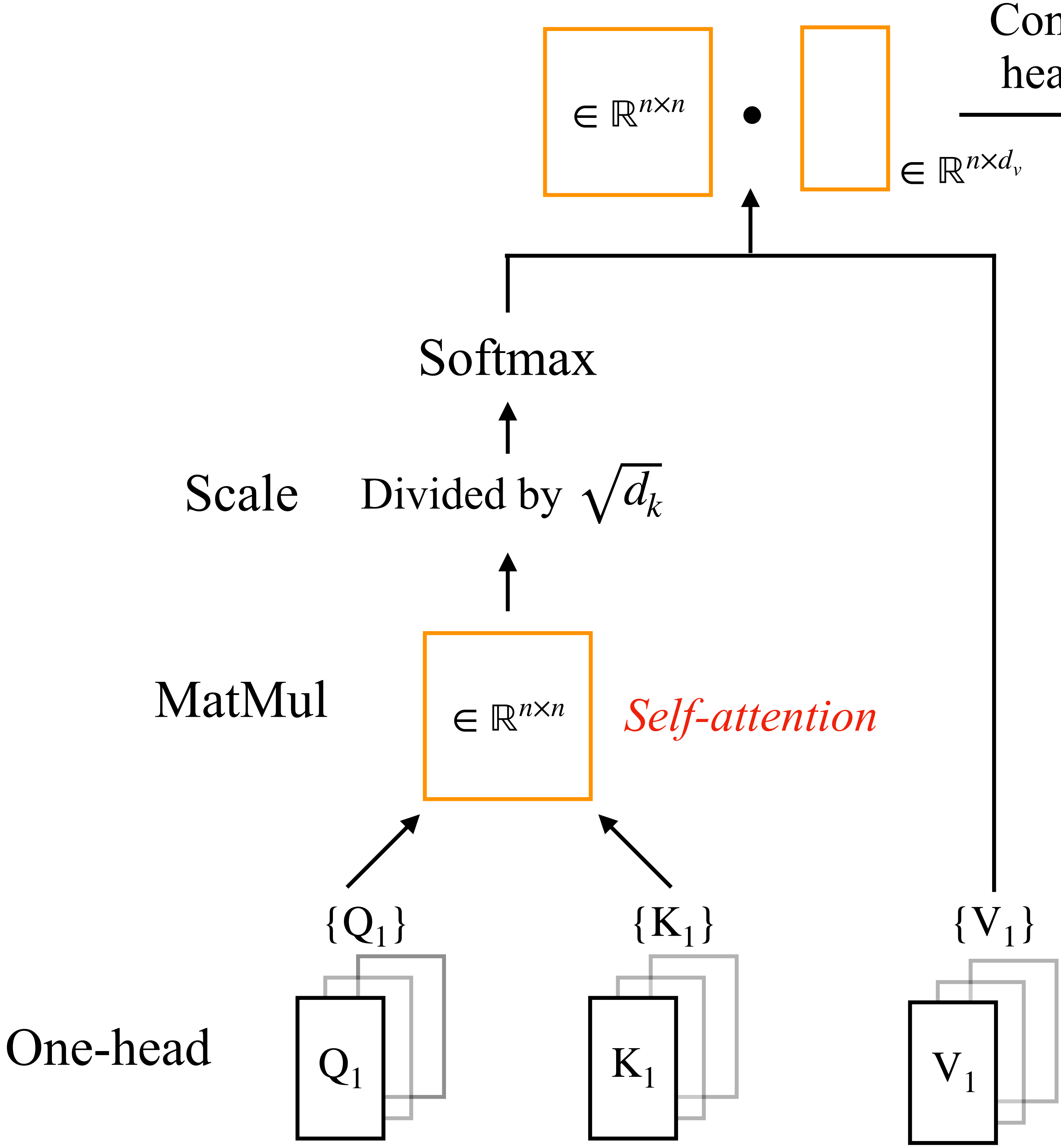
Encoder

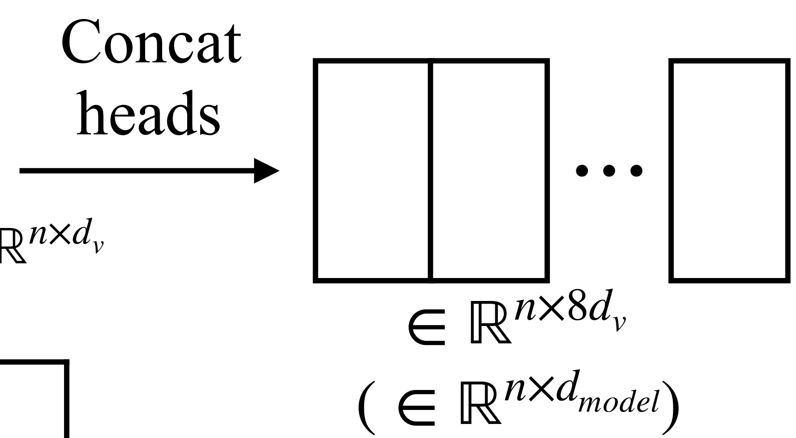


Mechanism(4/9)

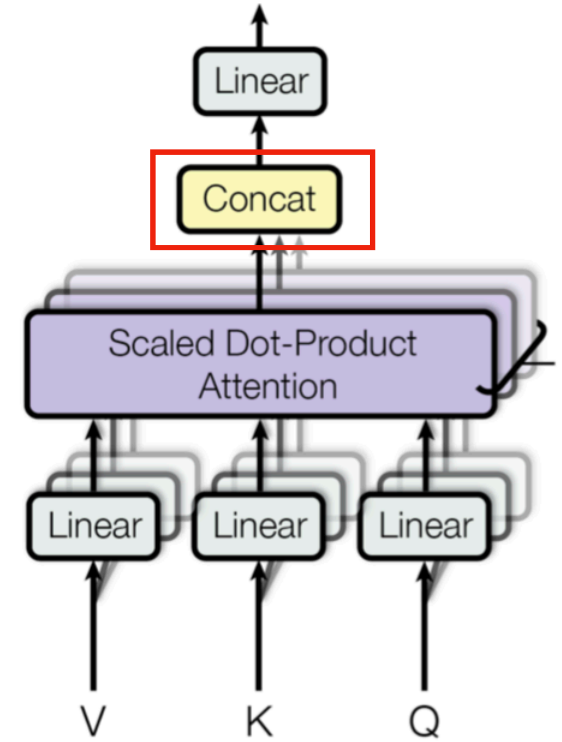
$$\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-heads attention

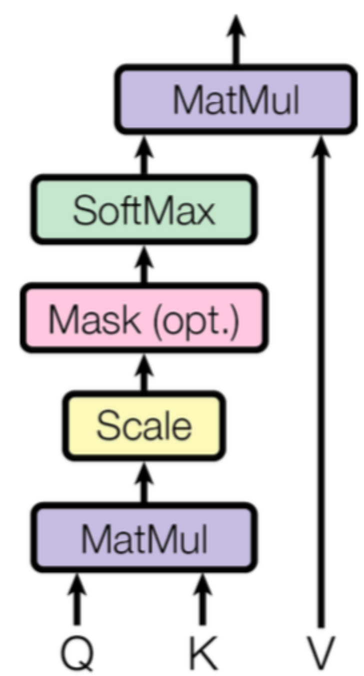




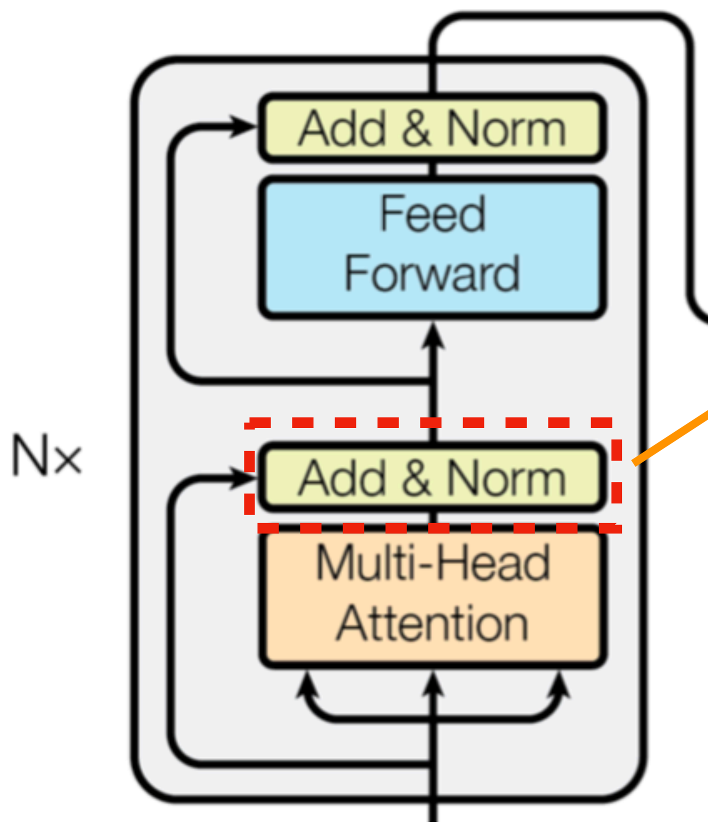
Multi-heads attention



Scaled Dot-Product Attention



Mechanism(5/9)



Encoder

Residual connection

$\in \mathbb{R}^{n\times 8d_v}$

$(\in \mathbb{R}^{n\times d_{model}})$

Normalize($\cdots$ + Q (Query))

$\in \mathbb{R}^{n\times d_{model}}$

=Normalize($\in \mathbb{R}^{n\times d_{model}}$)

$X = \in \mathbb{R}^{n\times d_{model}}$

Layer Normalize: $LN(z;\alpha,\beta) = \frac{(z-\mu)}{\sigma} \odot \alpha + \beta$

Mean : $\mu^l = \frac{1}{D}\sum_{i=1}^{D} z_i^l$

Gains : α

Biases : β

Standard Deviation : $\sigma^l = \sqrt{\frac{1}{D}\sum_{i=1}^{D}(z_i^l - \mu^l)^2}$

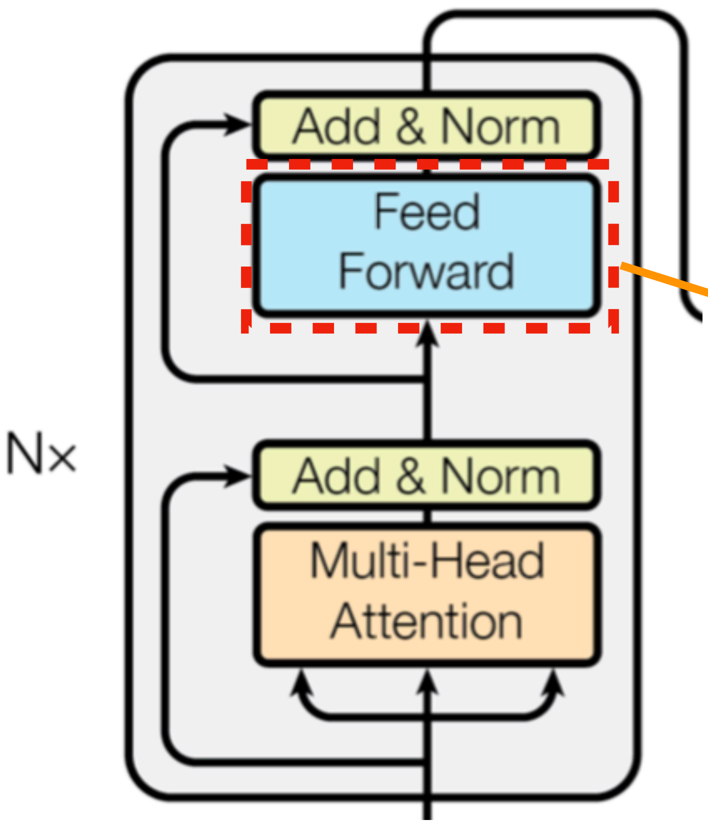
Mechanism(6/9)

$\hat{X} = \boxed{\in \mathbb{R}^{n \times d_{model}}}$

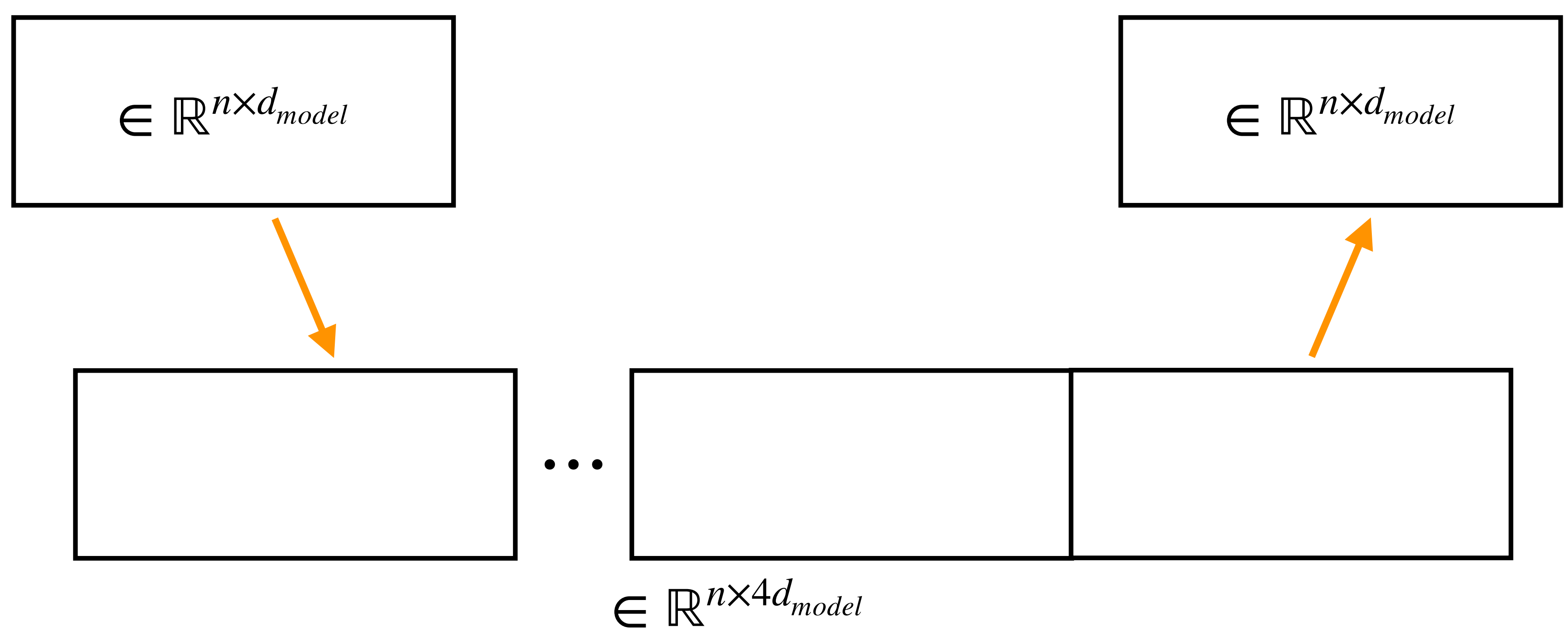
Feed Forward
(Dense、Conv1d)

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

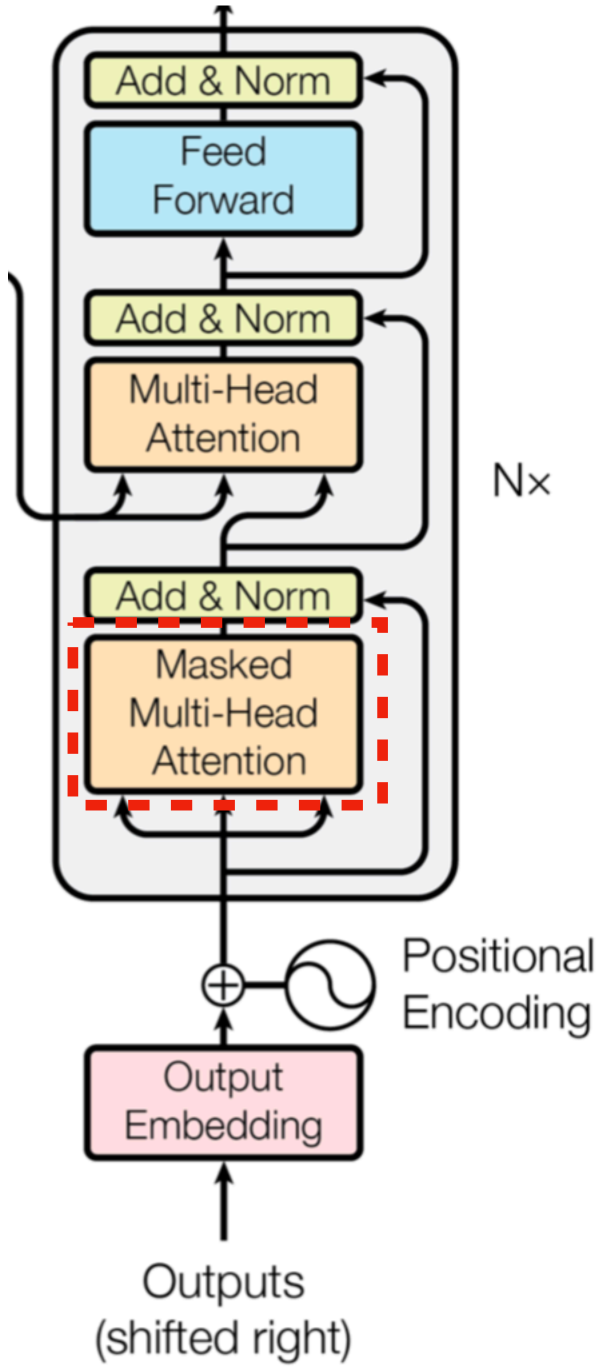
$$\text{Dense}(d_{model}) \rightarrow \text{Dense}\ (4 \times d_{model}) \rightarrow \text{Dense}(d_{model})$$



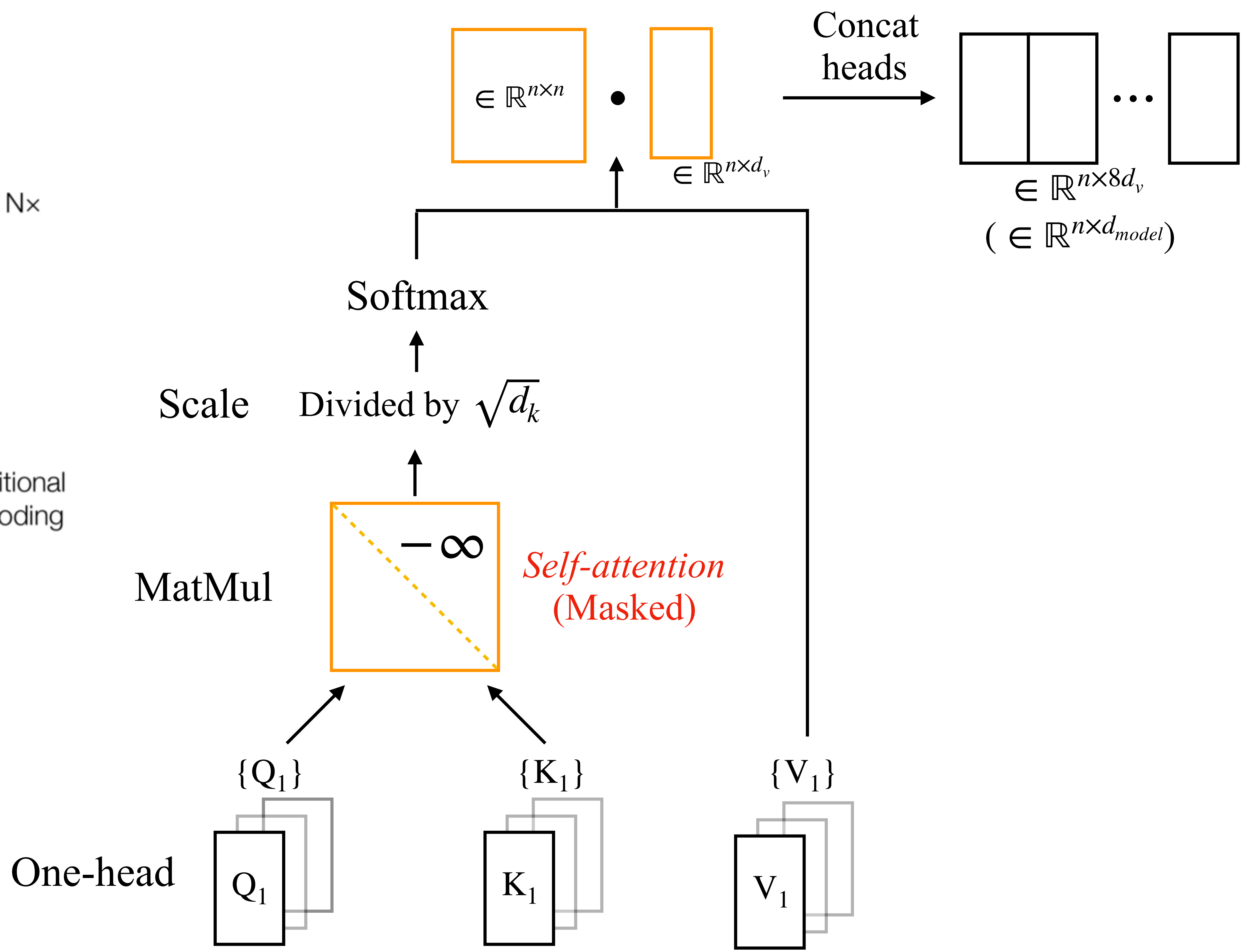
Encoder



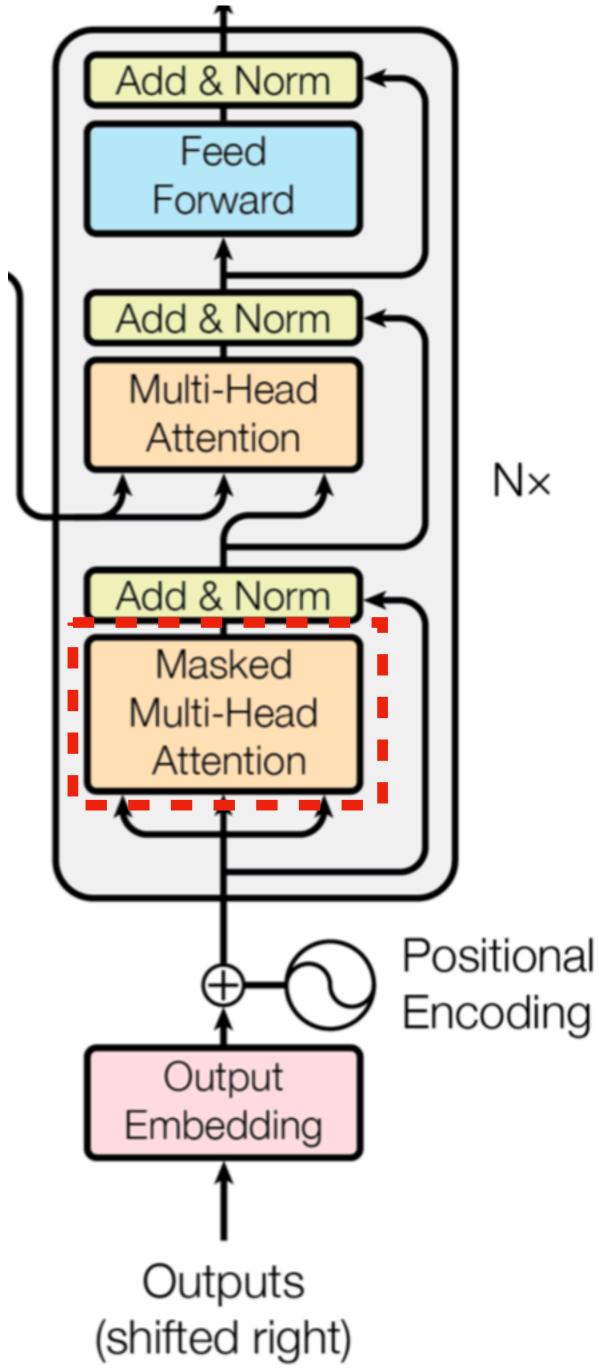
Mechanism(7/9)



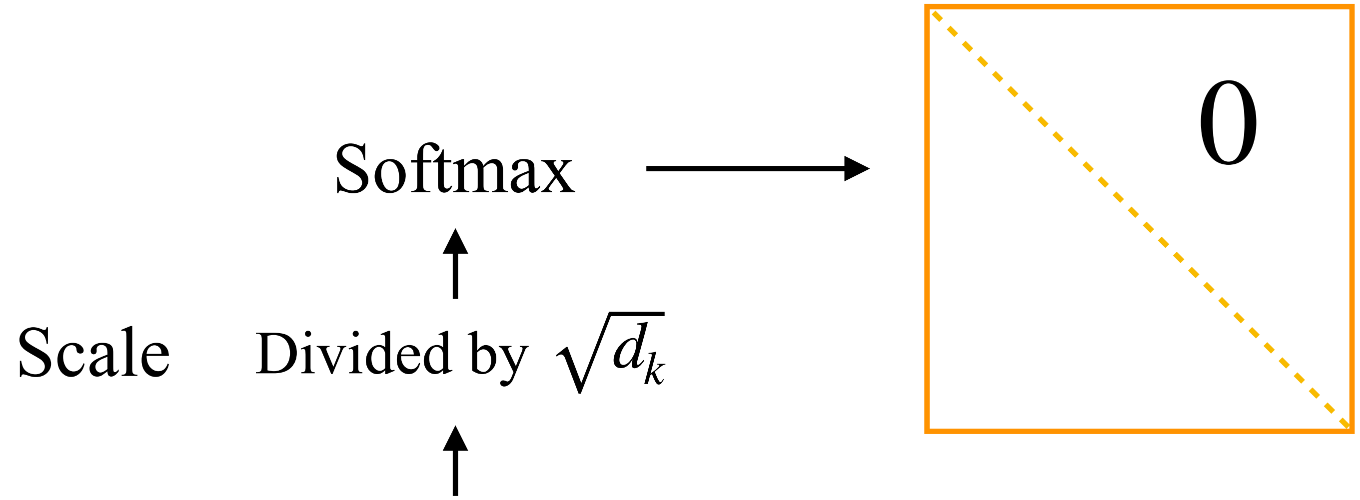
Decoder



Mechanism(8/9)

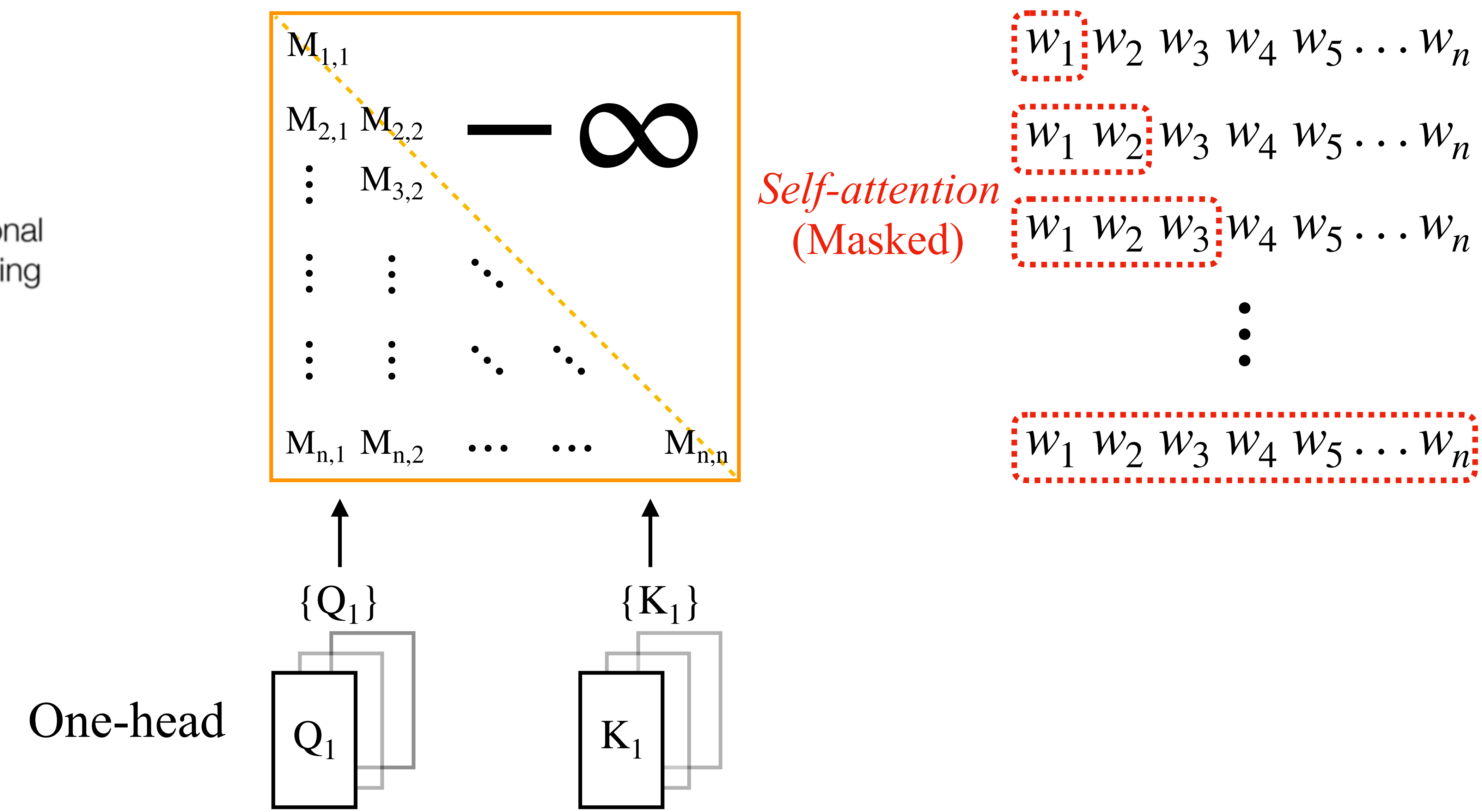


Decoder

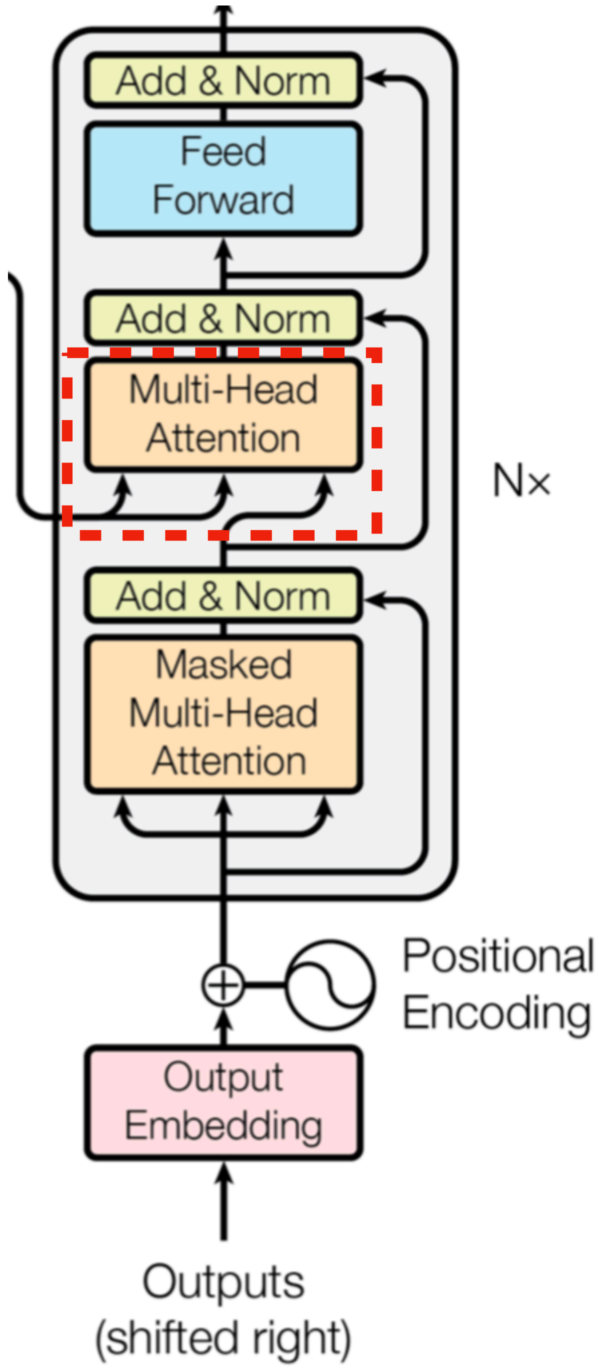


We need to prevent leftward information flow in the decoder

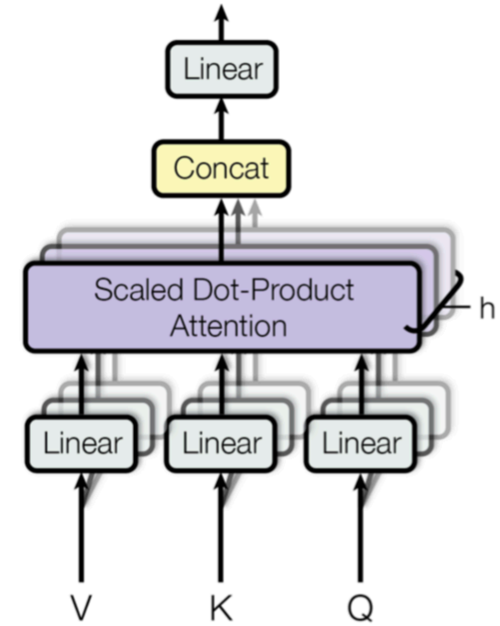
$M_{1,1}$: The similarity between Q_1 and K_1

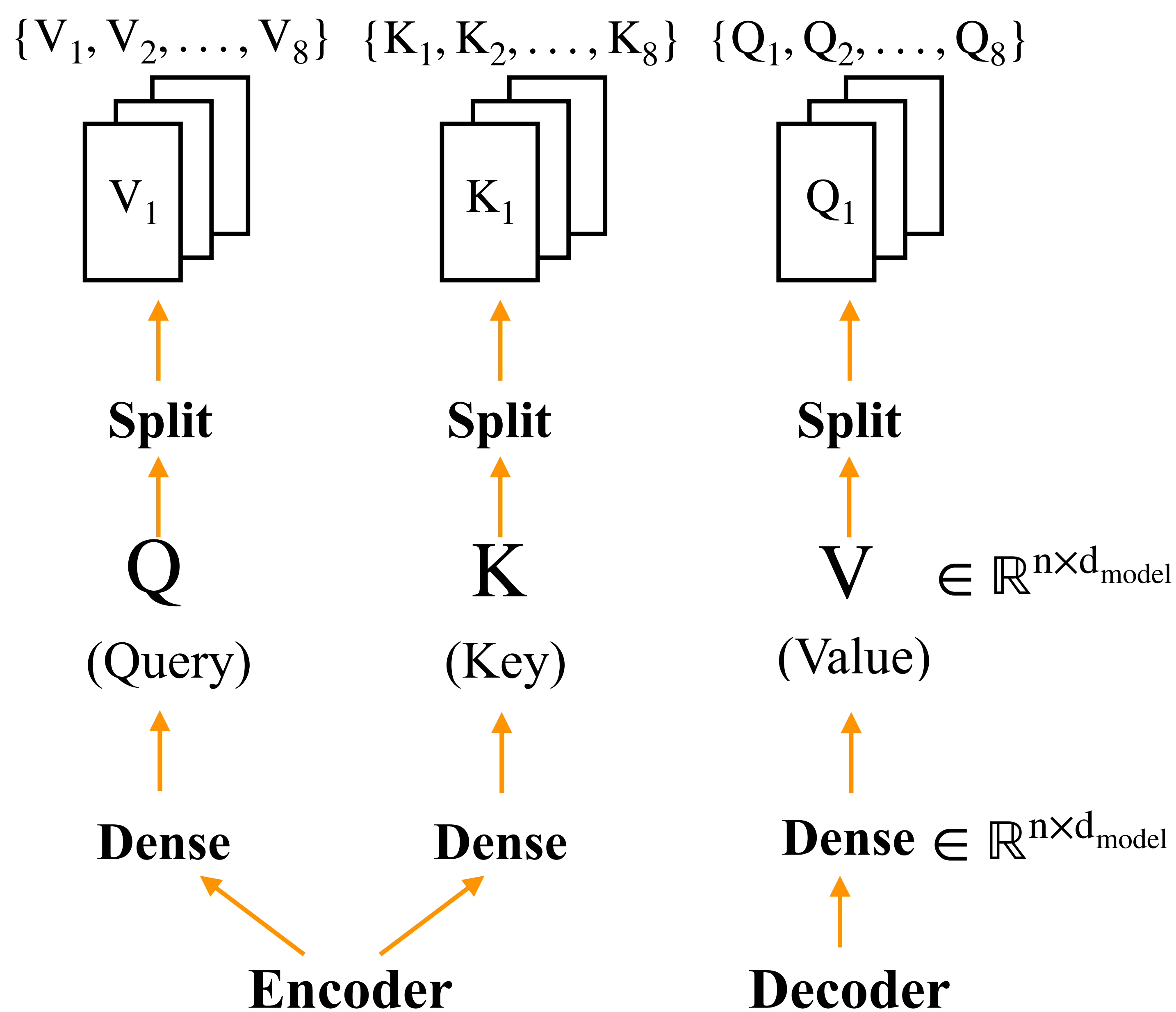


Mechanism(9/9)



Decoder





Evaluation(1/5)

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

n: Sequence length

d: represetation dimension(d_{model})

k: kernel size of convolutions

r: size of the neighborhood in restricted self attention

Why Self-Attention

1. The total computational complexity per layer.
2. The amount of computation that can be parallelized.
3. The path length between long-range dependencies.

Evaluation(2/5)

Hardware and Schedule

- 8 Nvidia P100 GPUS.
- 6 layers.
- **Base model:** training 100,000 steps(12 hours), 0.4 seconds per steps.
- **Big model:** training 300,000 steps(3.5 days), 1.0 seconds per steps.

Optimizer

- Adam with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$
- Learning rate $lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$

$warmup\_steps = 4000$

Regularization

- Apply **dropout** to the output of each sub-layer, before it is added to the sub-layer input and normalized.
- Apply **dropout** to the sum of the embeddings and the positional encodings.
- **Residual connection**
- **Label smoothing**

$q'(y|x) = (1 - \epsilon) \bullet q(y|x) + \frac{\epsilon}{K}$, K is target vocabulary size

Dataset

- Training set: WMT'14 English - German: 4.5M sentence pairs.
- Training set: WMT'14 English - French: 36M sentence pairs.
- Sentence pairs were batched together by approximate sequence length.
- Testing set: newstest2014.

Evaluation(3/5)

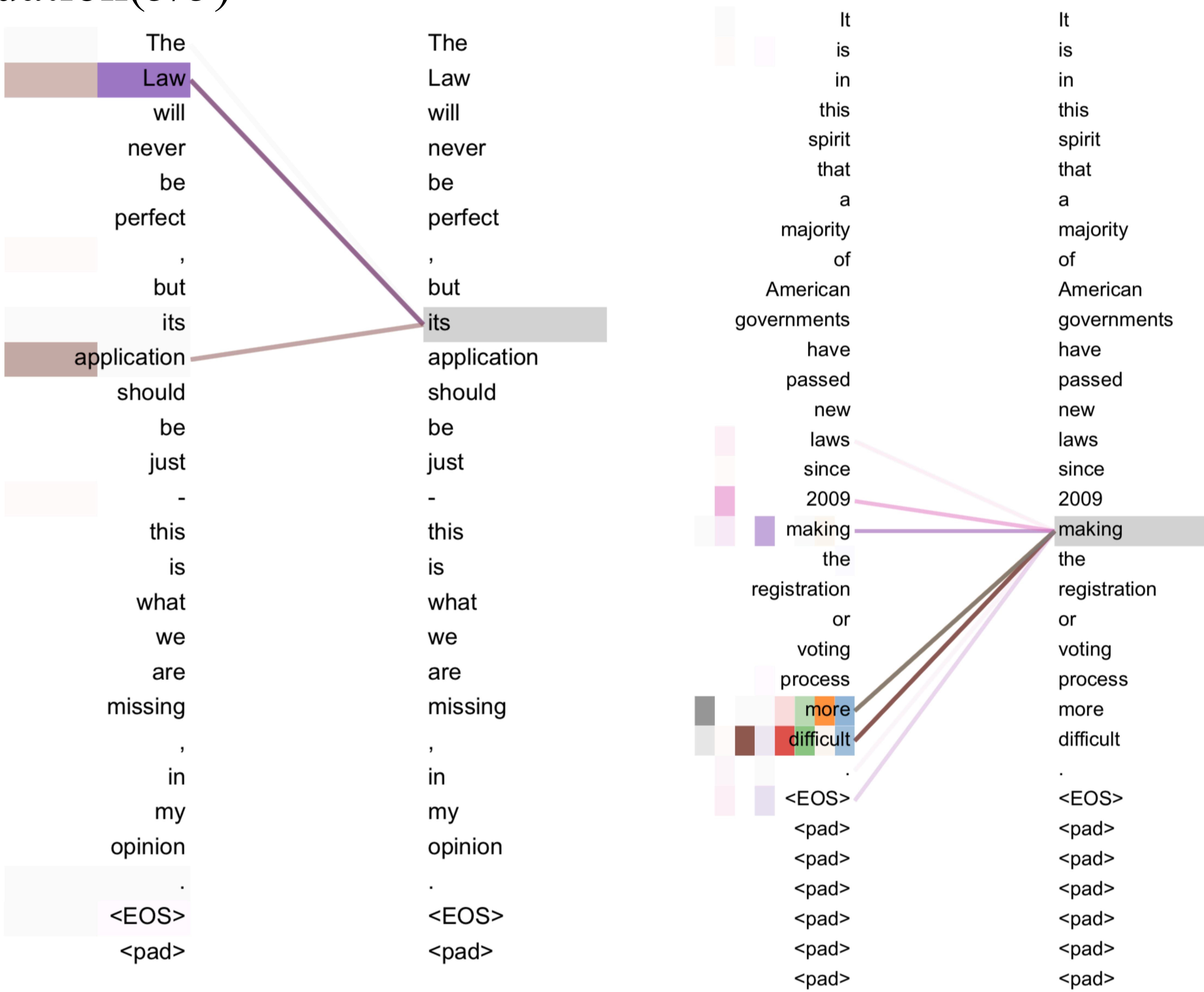
	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

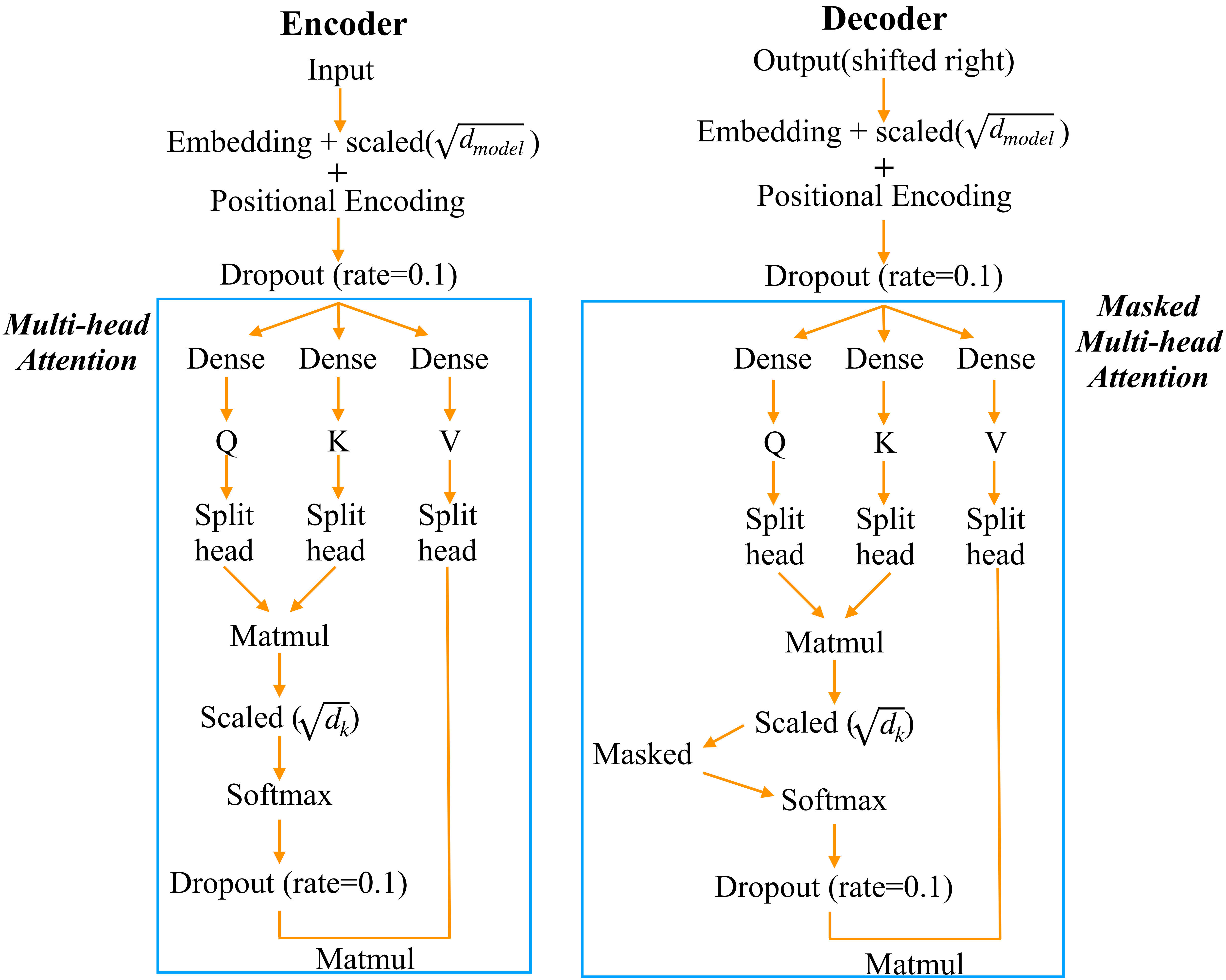
Evaluation(4/5)

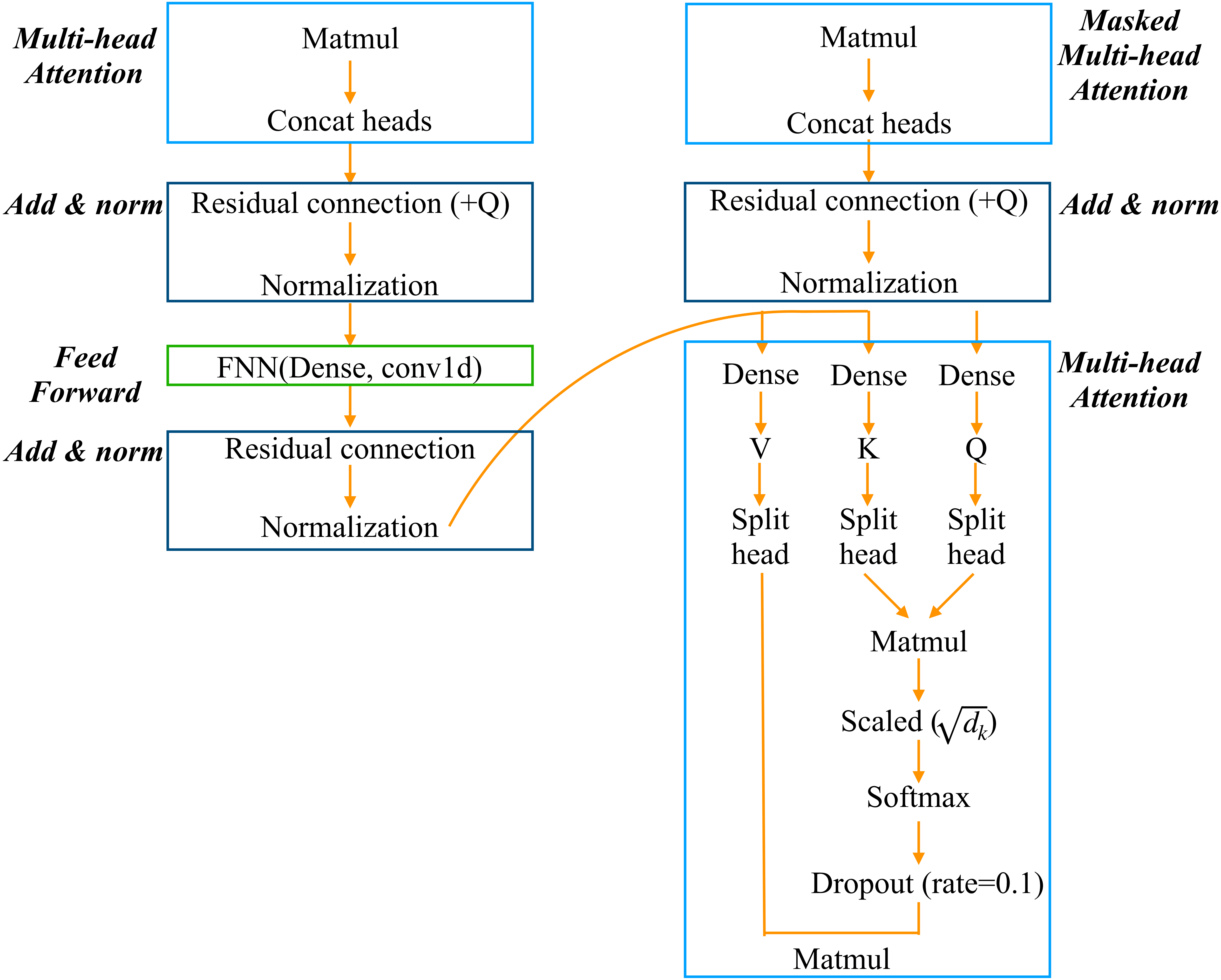
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

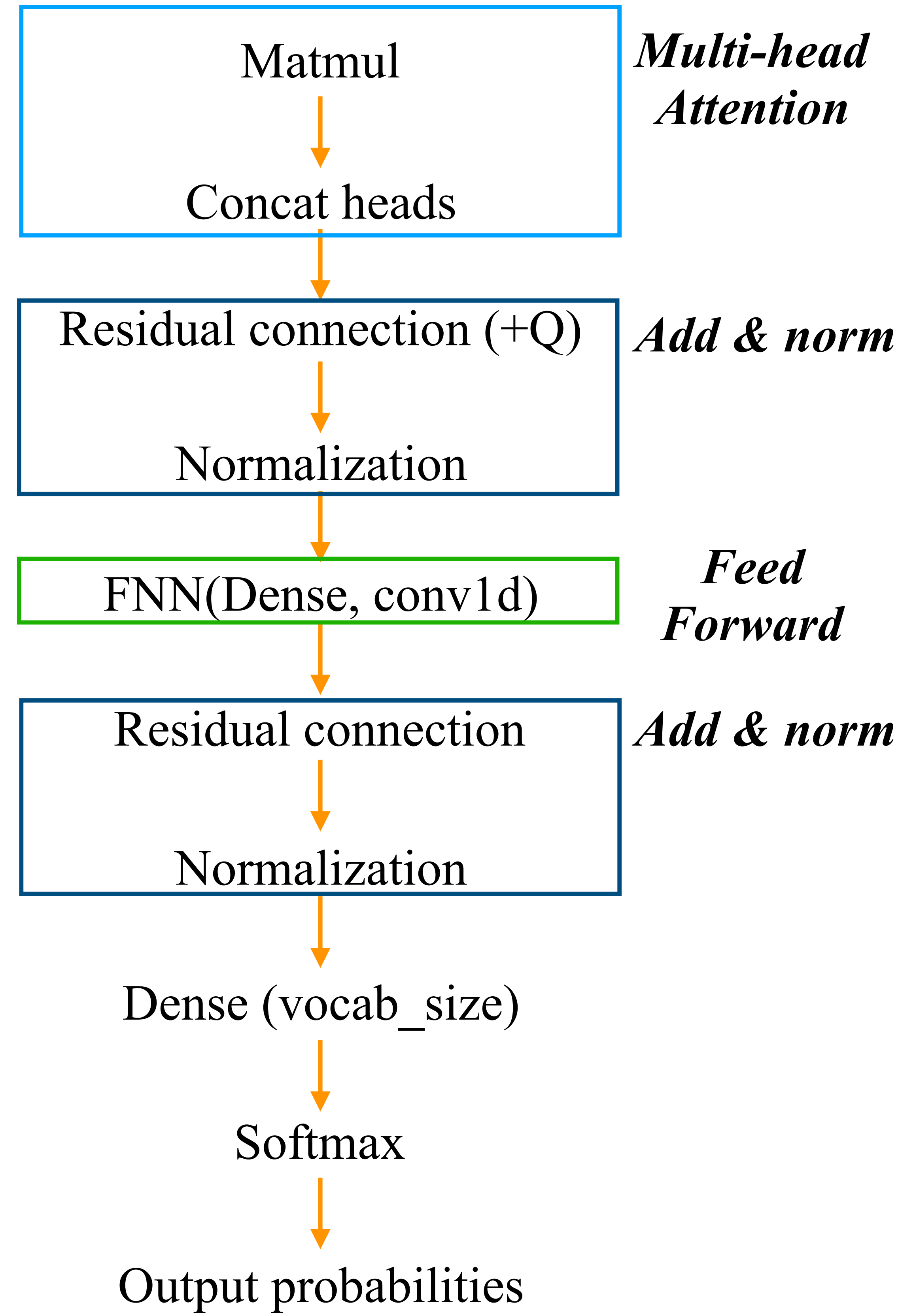
- **ByteNet:** 2 convolution layers
- **Deep-Att + PosUnk:** 2 Bi-LSTM layers(Encoder) + 1 LSTM layer (Decoder)
- **GNMT + RL:** 7 LSTM layers + 1 Bi-LSTM layer(Encoder) + 8 LSTM layers(Decoder)
- **Transformer(base):** training 100,000 steps(12 hours), 0.4 seconds per steps
- **Transformer(big):** training 300,000 steps(3.5 days), 1.0 seconds per steps

Evaluation(5/5)

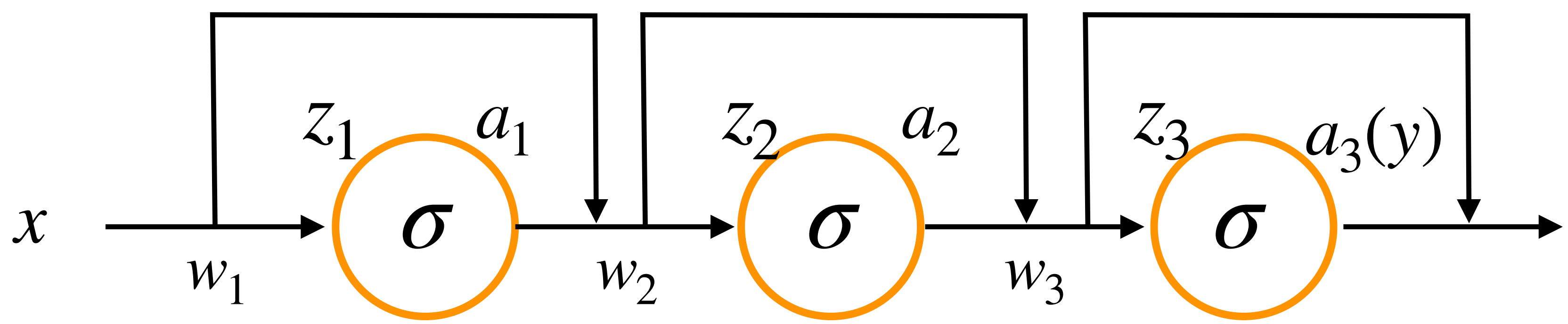



Encoder
Input
Embedding + scaled($\sqrt{d_{model}}$)
+
Positional Encoding
Dropout (rate=0.1)
Multi-head Attention
Dense
Dense
Dense
Q
K
V
Split head
Split head
Split head
Matmul
Scaled ($\sqrt{d_k}$)
Softmax
Dropout (rate=0.1)
Matmul
Decoder
Output(shifted right)
Embedding + scaled($\sqrt{d_{model}}$)
+
Positional Encoding
Dropout (rate=0.1)
Masked Multi-head Attention
Dense
Dense
Dense
Q
K
V
Split head
Split head
Split head
Matmul
Scaled ($\sqrt{d_k}$)
Masked
Softmax
Dropout (rate=0.1)
Matmul


Multi-head Attention
Matmul
Concat heads
Add & norm
Residual connection (+Q)
Normalization
Feed Forward
FNN(Dense, conv1d)
Add & norm
Residual connection
Normalization
Masked Multi-head Attention
Matmul
Concat heads
Add & norm
Residual connection (+Q)
Normalization
Multi-head Attention
Dense
Dense
Dense
V
K
Q
Split head
Split head
Split head
Matmul
Scaled ($\sqrt{d_k}$)
Softmax
Dropout (rate=0.1)
Matmul


Matmul
Concat heads
Multi-head Attention
Residual connection (+Q)
Normalization
Add & norm
FNN(Dense, conv1d)
Feed Forward
Residual connection
Normalization
Add & norm
Dense (vocab_size)
Softmax
Output probabilities

Residual Connection



$$
\begin{aligned}
z_1 &= w_1 x \\
a_1 &= \sigma(z_1) + a_0 \\
z_2 &= w_2 a_1 \\
a_2 &= \sigma(a_2) + a_1 \\
z_3 &= w_3 a_2 \\
a_3 &= \sigma(z_3) + a_2
\end{aligned}
$$

$$
\begin{aligned}
\text{Residual connection : } \frac{\partial C}{\partial z_1} &= \frac{\partial C}{\partial a_3}\frac{\partial a_3}{\partial z_3}\frac{\partial z_3}{\partial a_2}\frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial a_1}\frac{\partial a_1}{\partial z_1} \\
&= \frac{\partial C}{\partial a_3}[\sigma'(z_3) + \frac{1}{w_3}]w_3[\sigma'(z_2) + \frac{1}{w_2}]w_2[\sigma'(z_1) + \frac{1}{w_1}] \\
&= \frac{\partial C}{\partial a_3}\underbrace{[w_3\sigma'(z_3) + 1]}_{\geqslant 1}\underbrace{[w_2\sigma'(z_2) + 1]}_{\geqslant 1}\underbrace{[\sigma'(z_1) + \frac{1}{w_1}]}_{\geqslant 1}
\end{aligned}
$$