南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

大模型网络结构

魏明强、宫丽娜
计算机科学与技术学院

目录

目录

Transformer

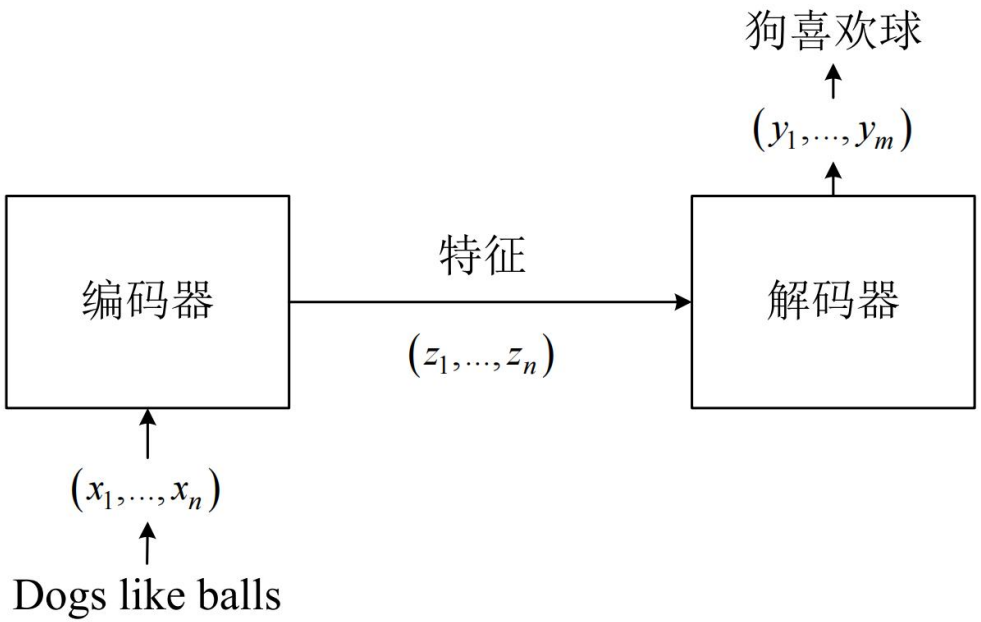
南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **面对问题：**记录输入序列中的长期依赖关系
- Transformer 利用注意力机制完成对源语言序列和目标语言序列全局依赖的建模

Transformer 是自然语言处理领域的颠覆者，为后续大模型网络结构（BERT、GPT）的发展奠定了基础

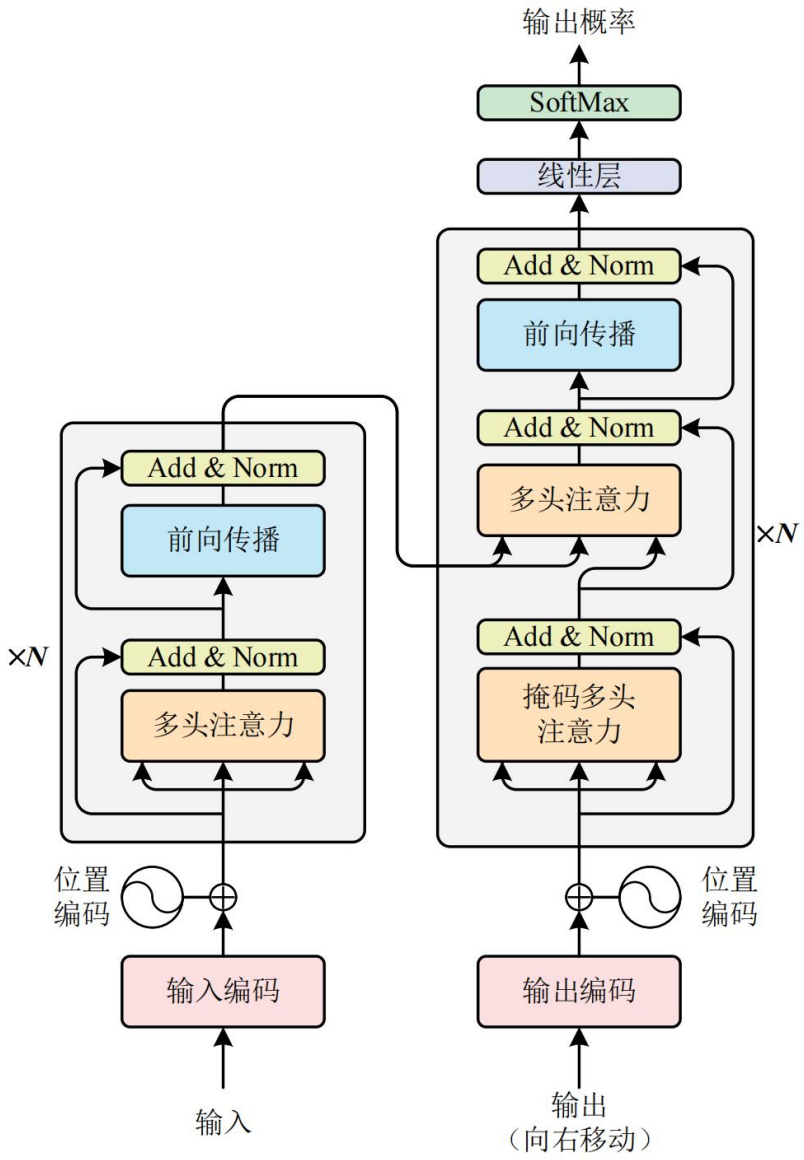
Transformer

- **基本结构**：编码器-解码器结构
- 编码器输入，解码器输出



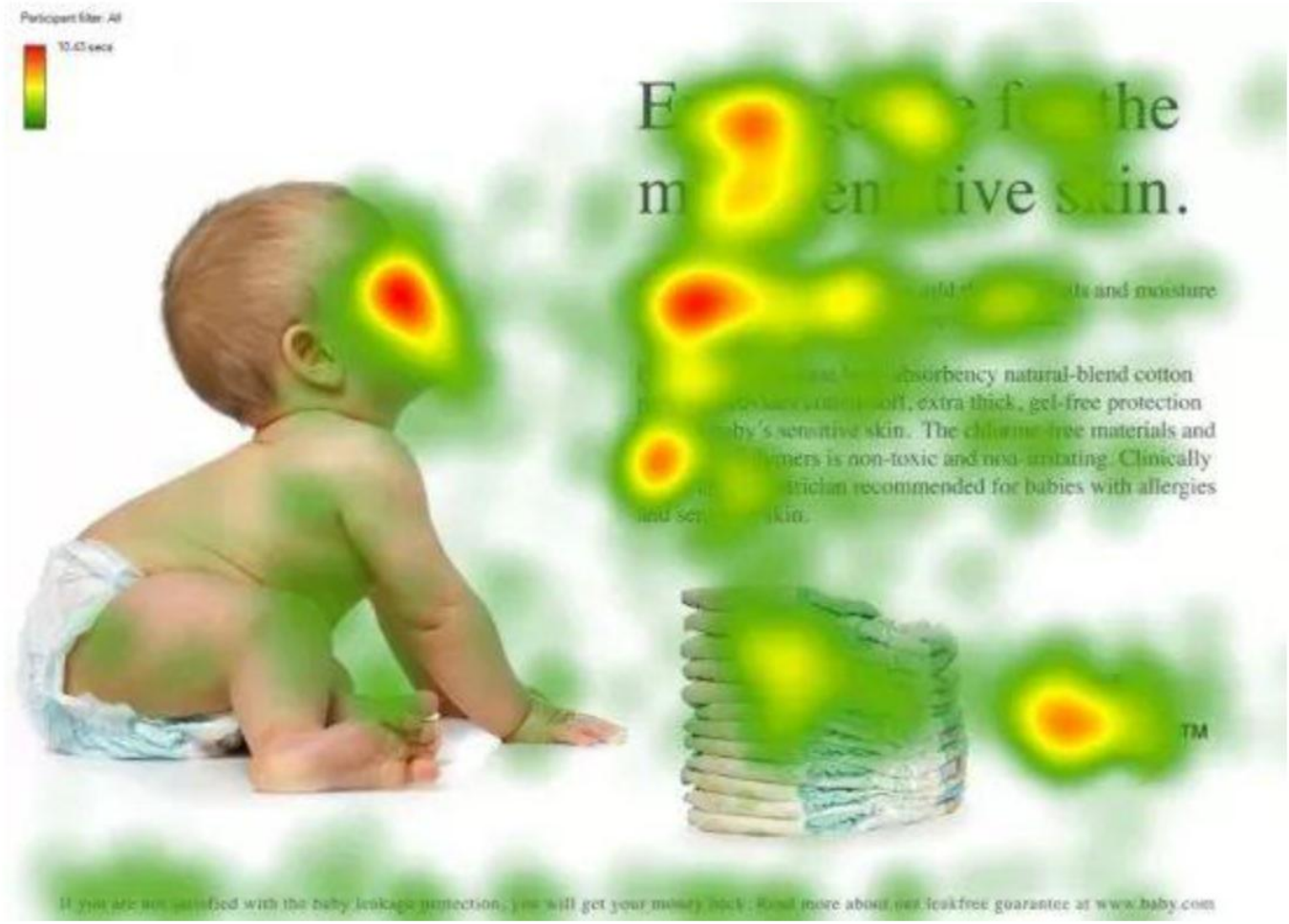
Transformer

- 嵌入表示层
- 注意力层
- 位置前馈感知层
- 残差连接
- 层归一化



目录

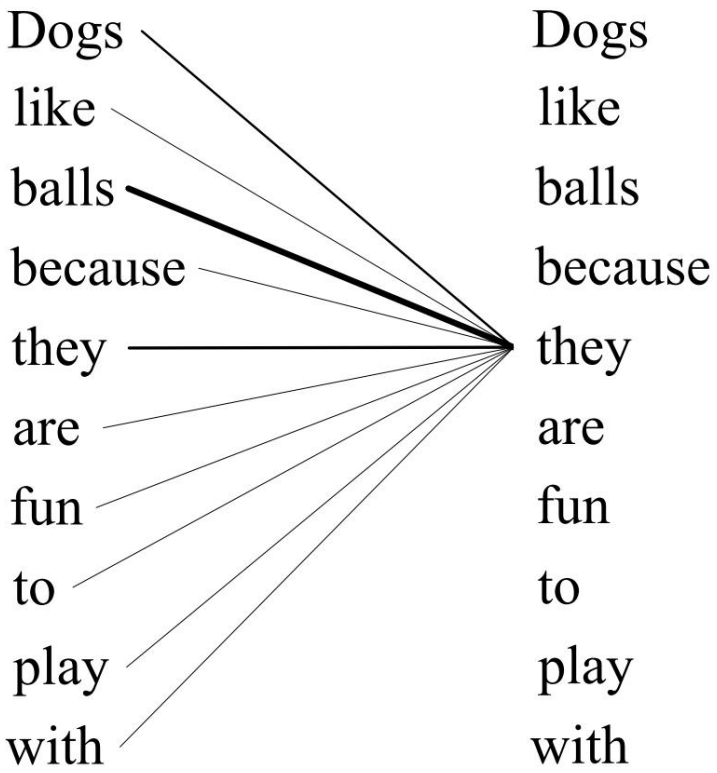
注意力机制



注意力机制是对人类行为的一种仿生，起源于对人类视觉注意机制的研究

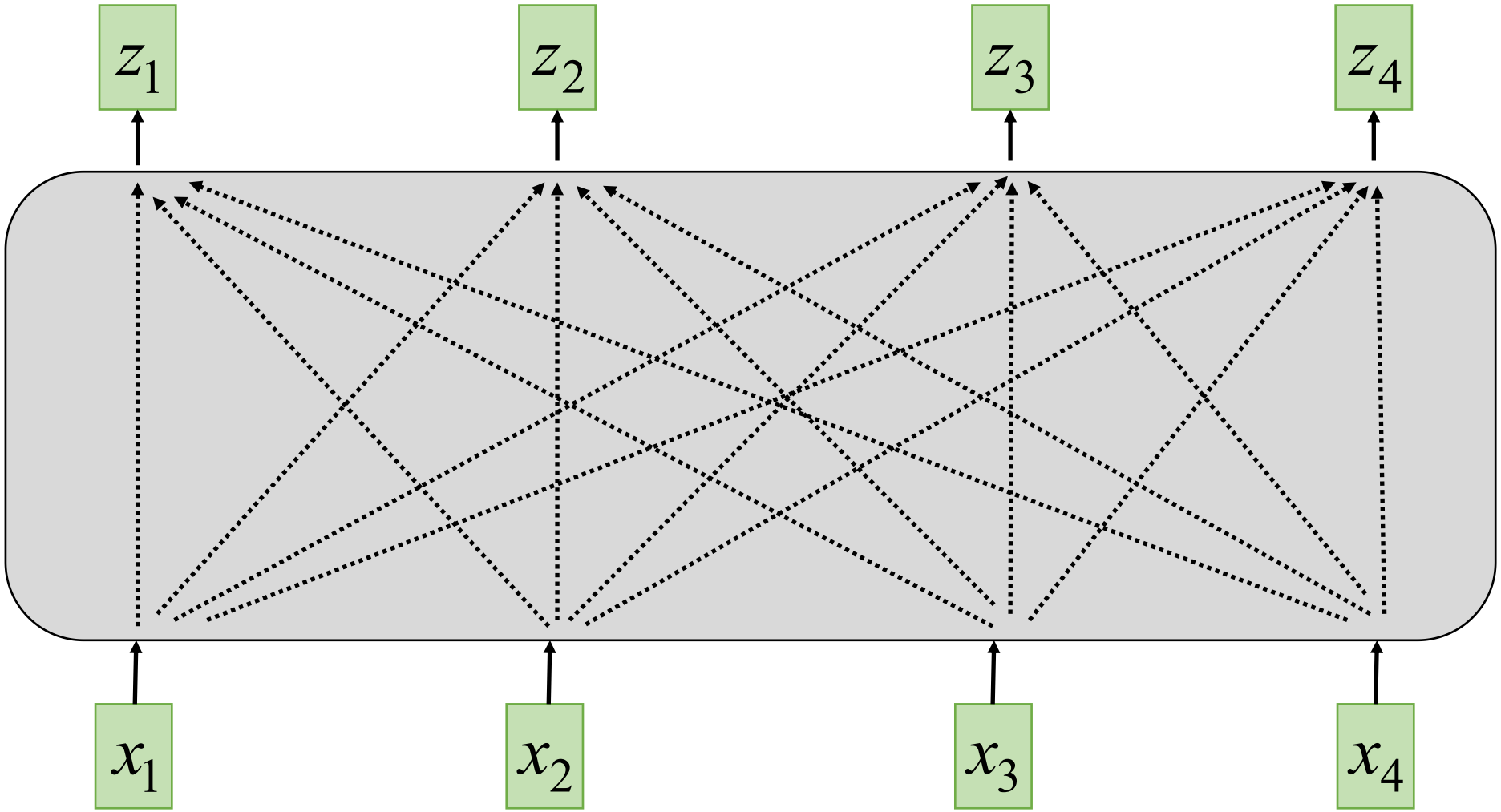
注意力机制

1.自注意力模块

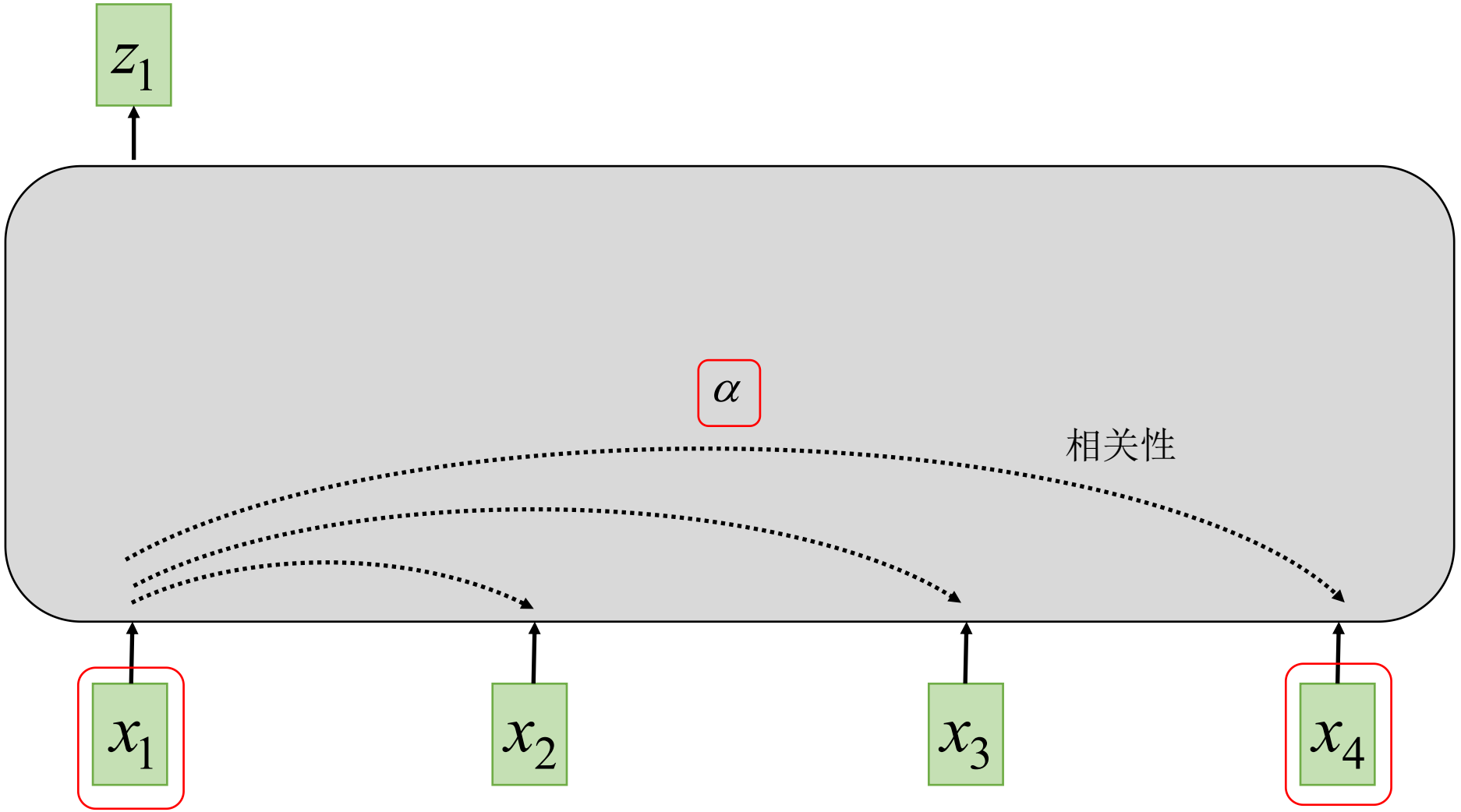


不同的单词通过不同权重计算影响

注意力机制

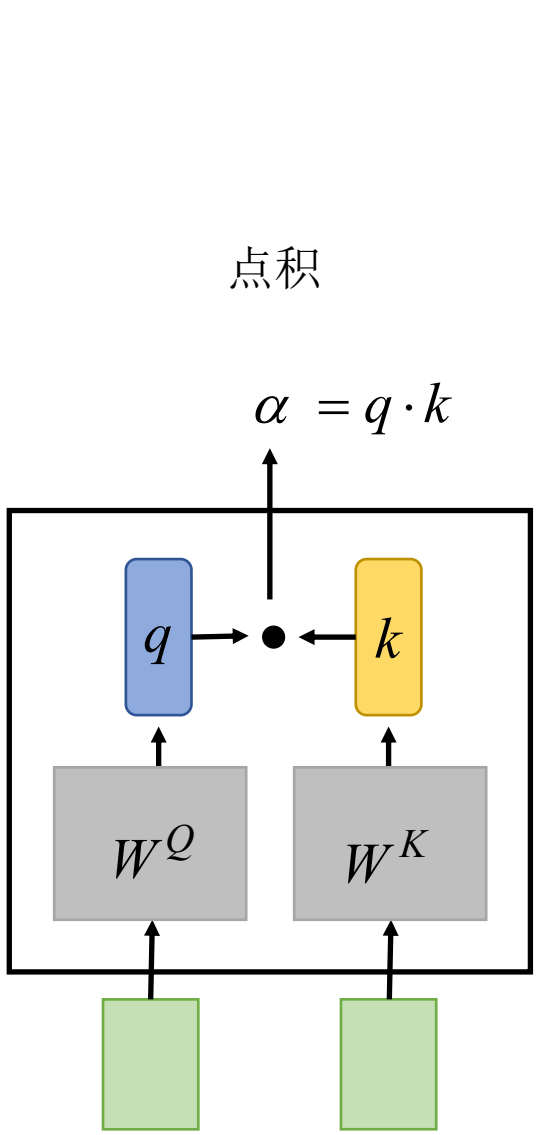


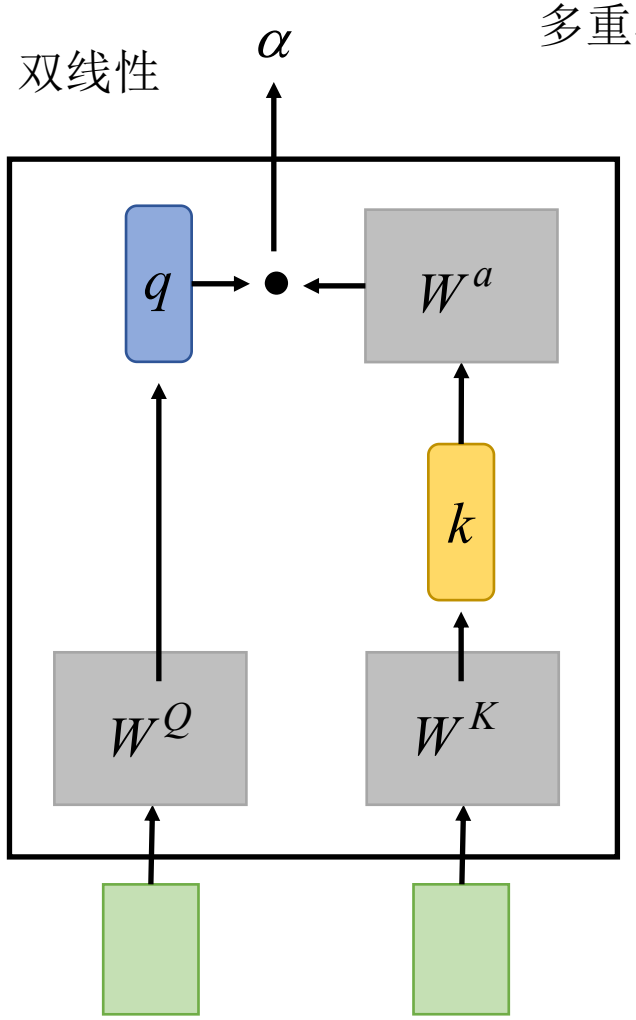
注意力机制

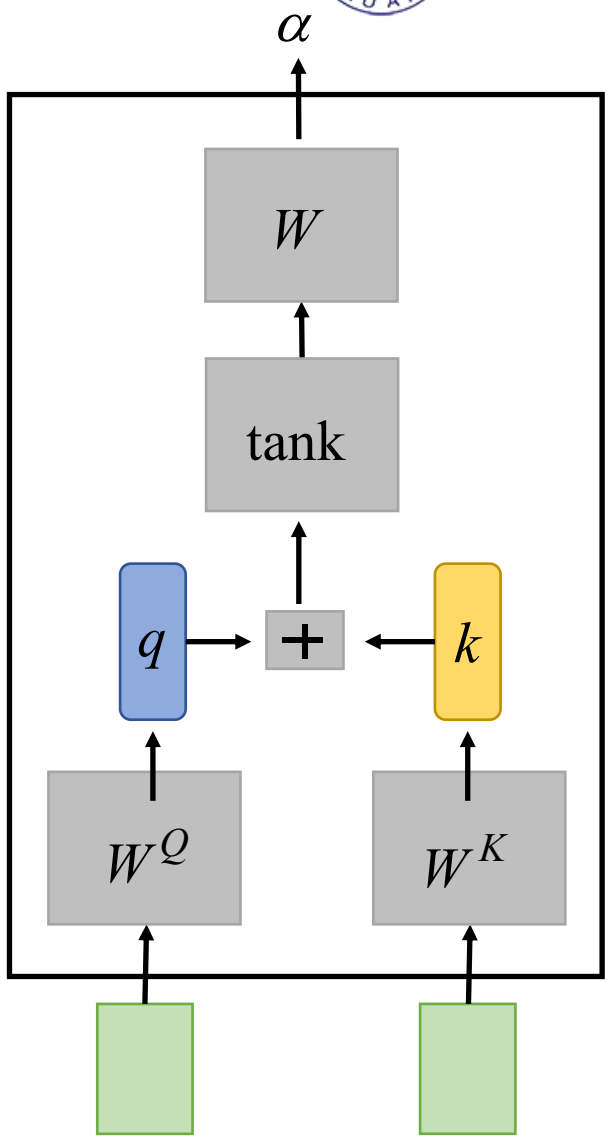


通过词与词之间的关系来更好地理解当前词的意思

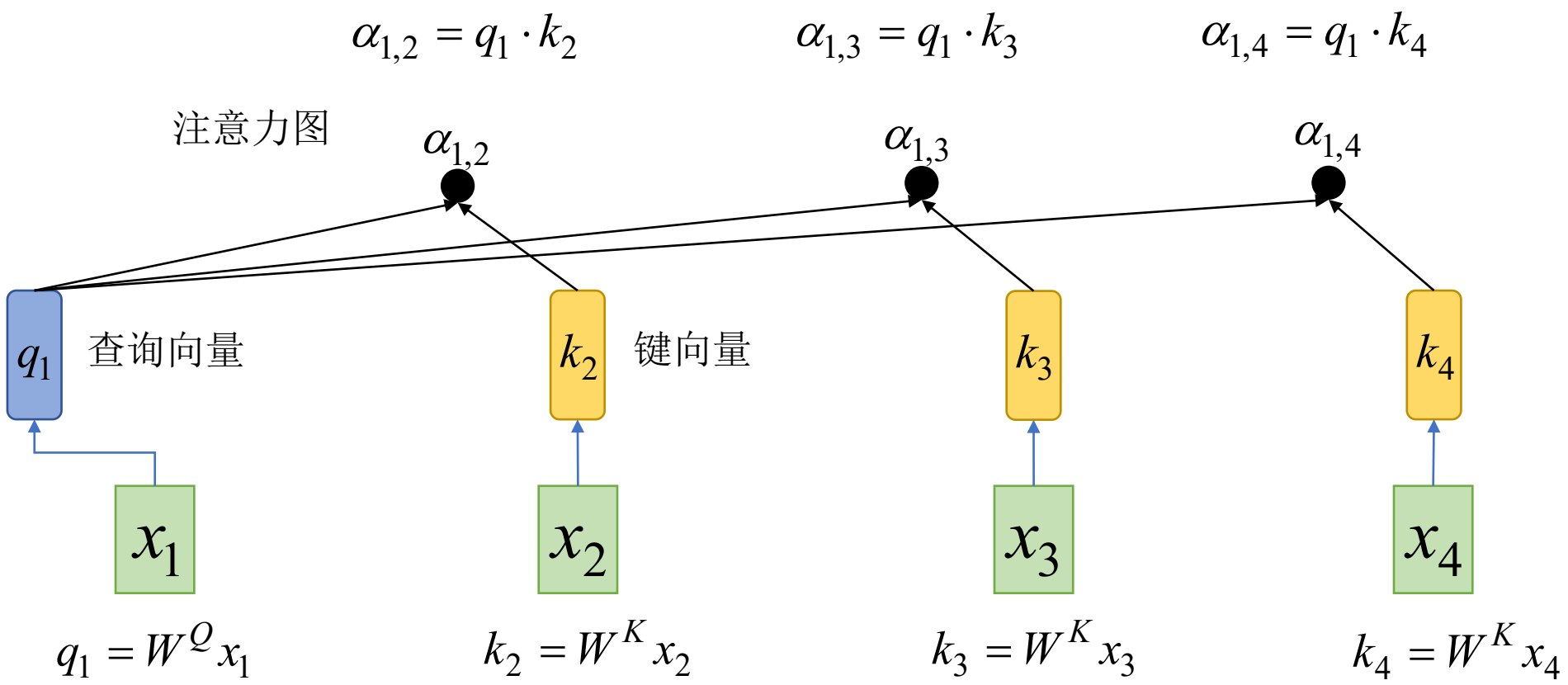
注意力机制


点积
$\alpha = q \cdot k$
q
k
W^Q
W^K


双线性
α
q
W^a
k
W^Q
W^K

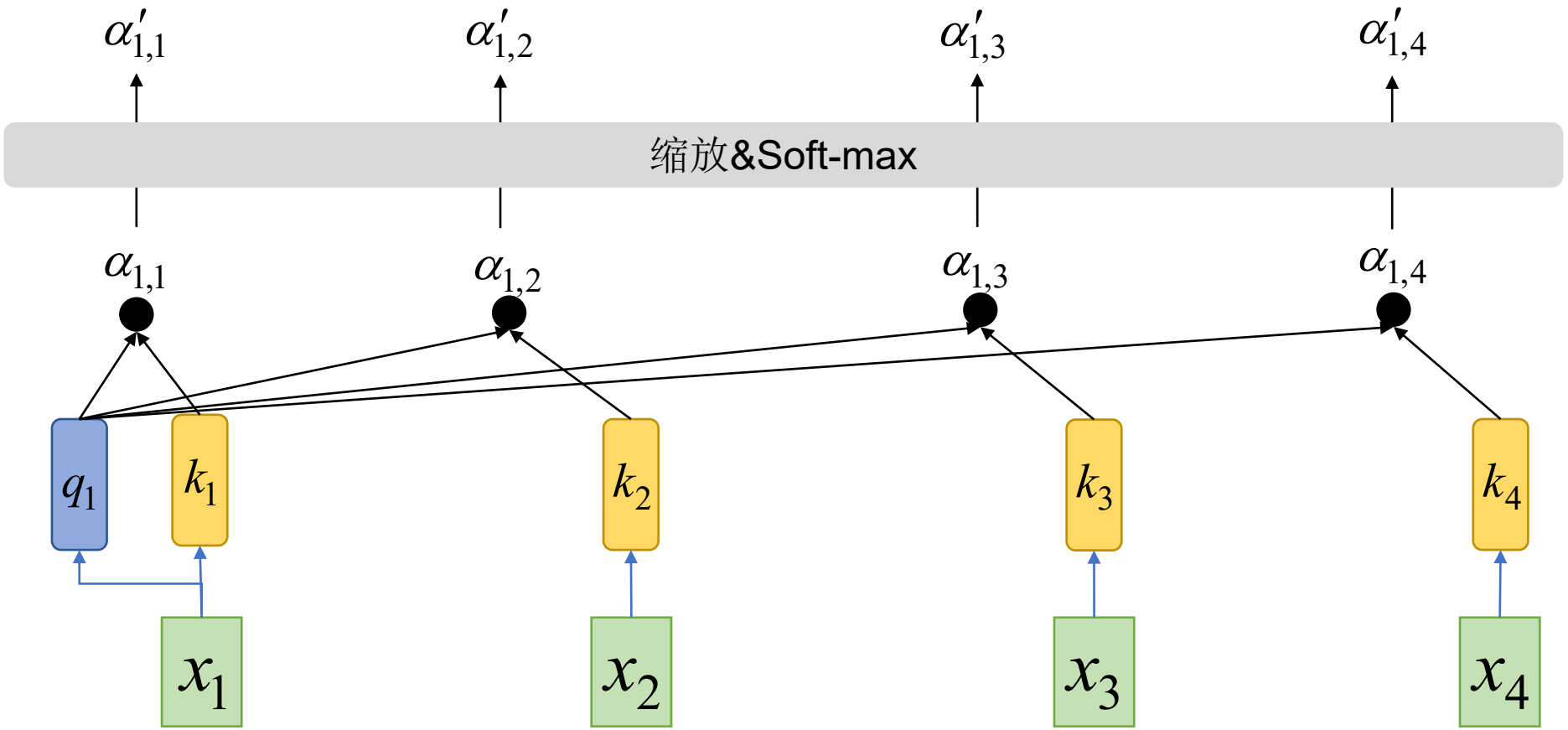

多重感知机
α
W
tank
q
+
k
W^Q
W^K

注意力机制


$\alpha_{1,2} = q_1 \cdot k_2$
$\alpha_{1,3} = q_1 \cdot k_3$
$\alpha_{1,4} = q_1 \cdot k_4$
注意力图
$\alpha_{1,2}$
$\alpha_{1,3}$
$\alpha_{1,4}$
q_1
查询向量
k_2
键向量
k_3
k_4
x_1
x_2
x_3
x_4
$q_1 = W^Q x_1$
$k_2 = W^K x_2$
$k_3 = W^K x_3$
$k_4 = W^K x_4$

注意力机制

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

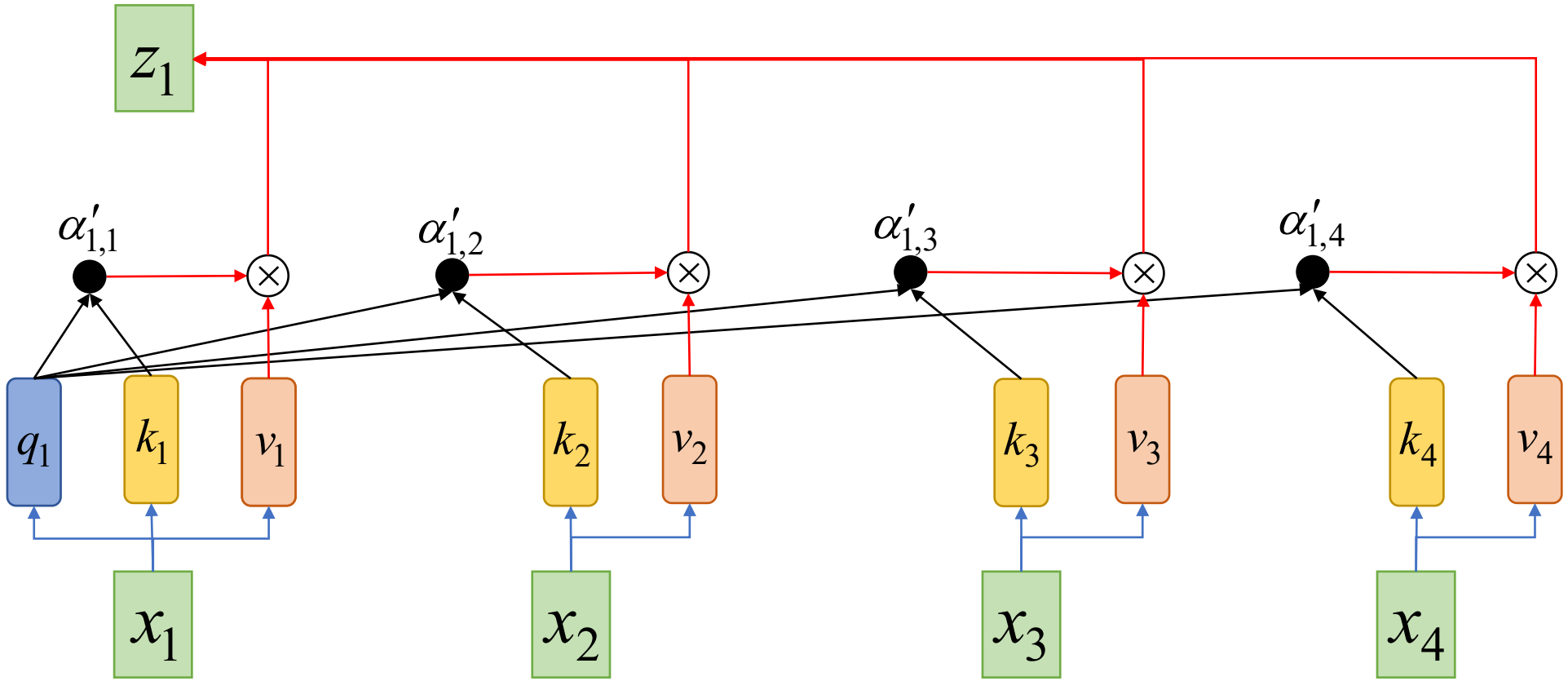


$$q_1 = W^Q x_1 \qquad k_2 = W^K x_2 \qquad k_3 = W^K x_3 \qquad k_4 = W^K x_4$$

$$k_1 = W^K x_1$$

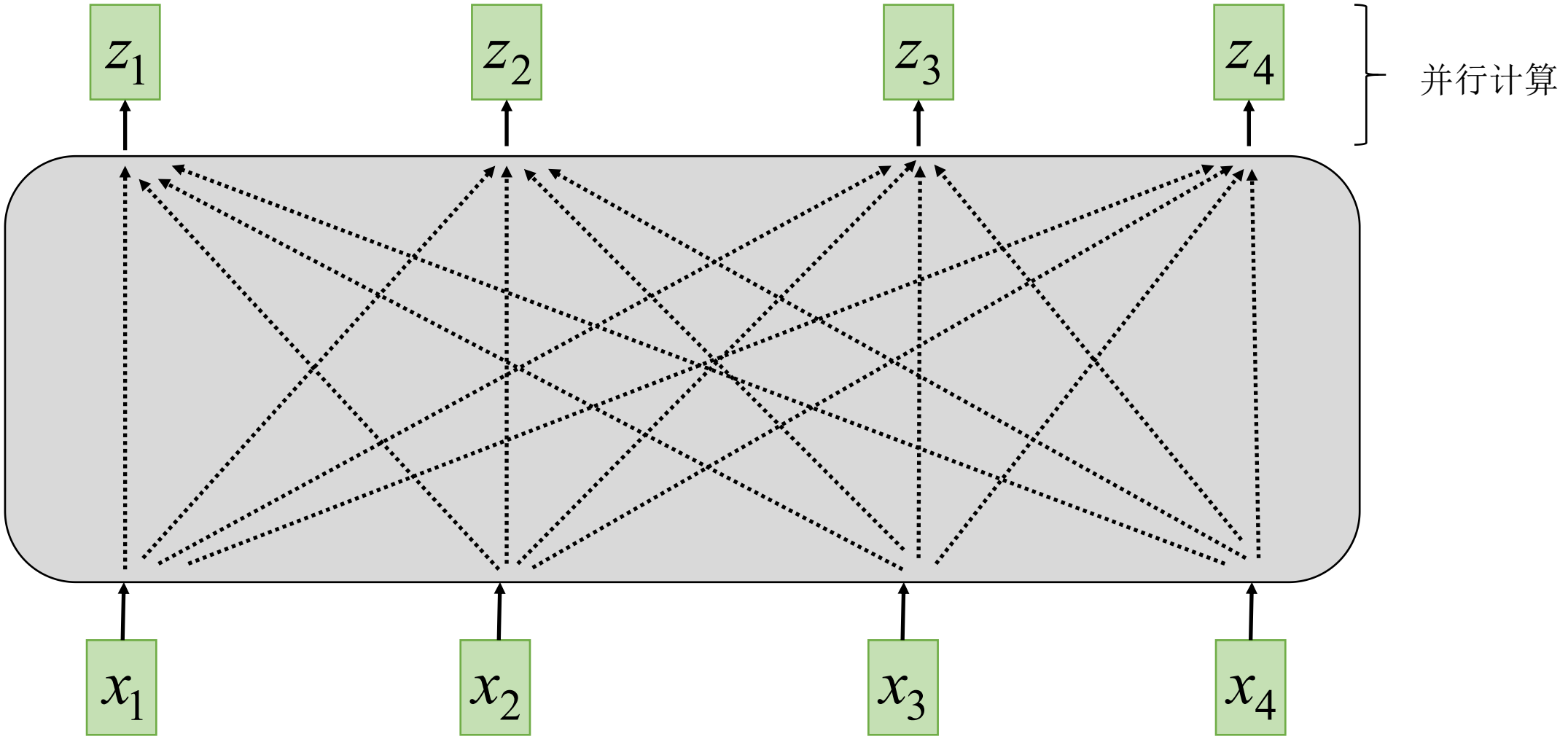
注意力机制

$$z_1 = \sum_i \alpha'_{1,i} v_i$$



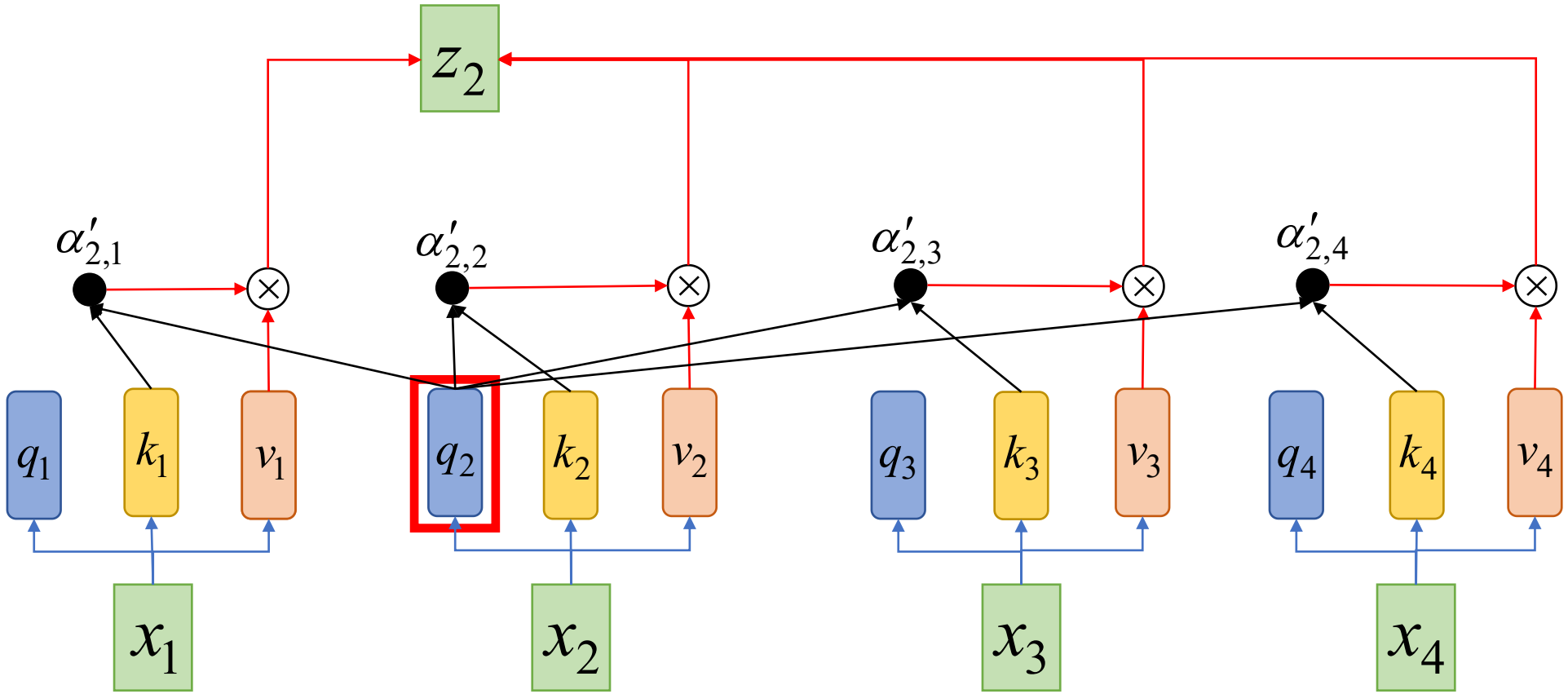
$v_1 = W^V x_1$ $v_2 = W^V x_2$ $v_3 = W^V x_3$ $v_4 = W^V x_4$

注意力机制



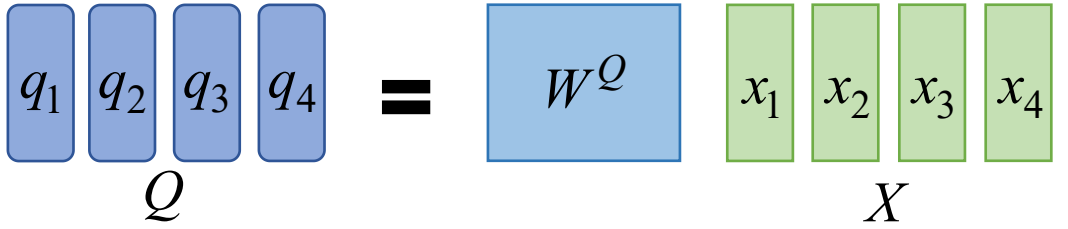
注意力机制

$$z_2 = \sum_i \alpha'_{2,i} v_i$$

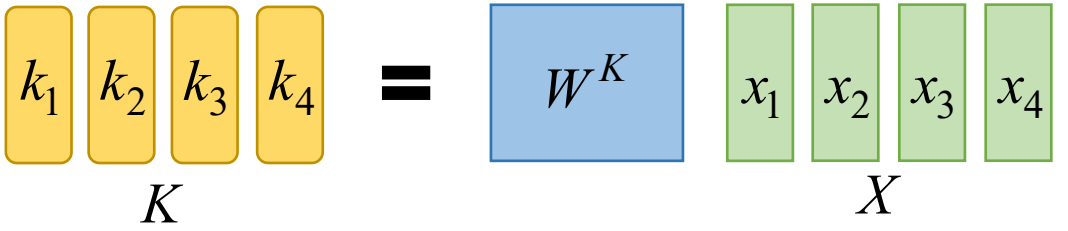


注意力机制

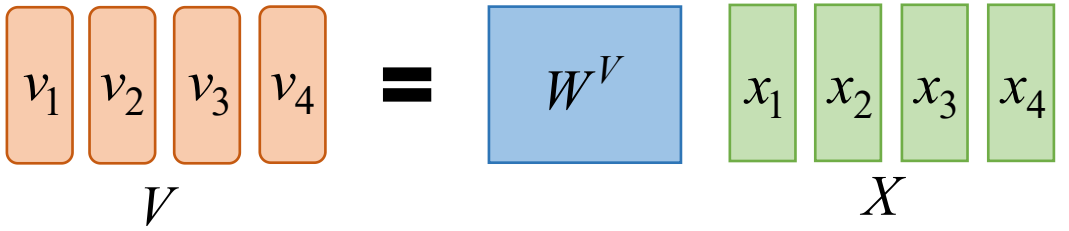
$$q_i = W^Q x_i$$

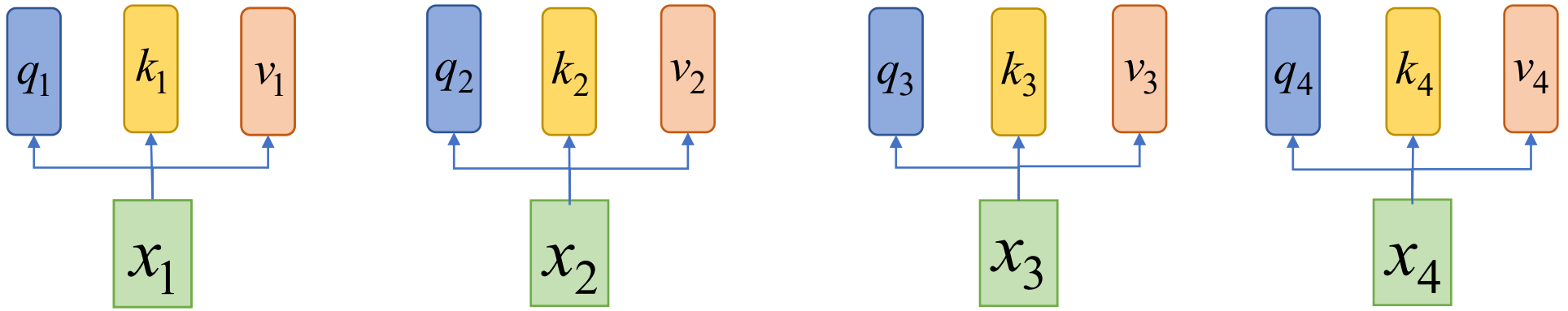


$$k_i = W^K x_i$$

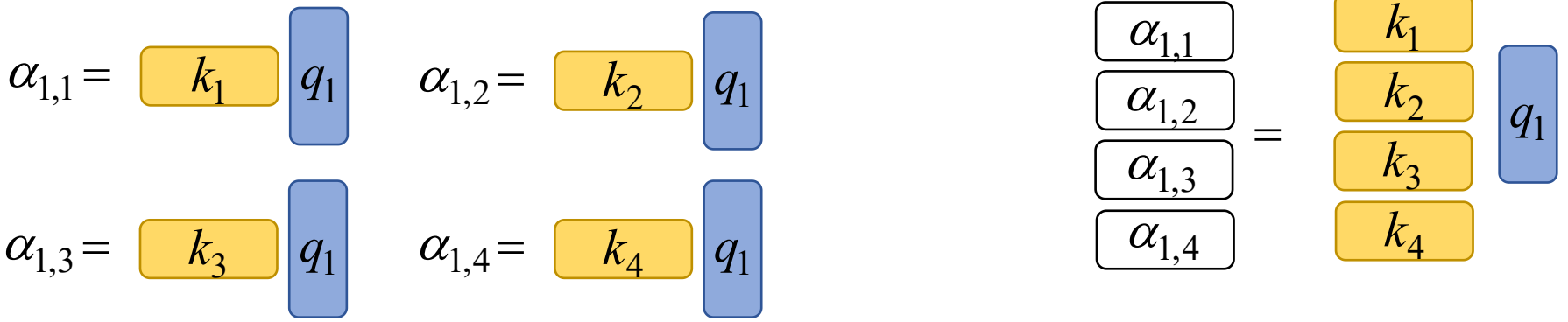


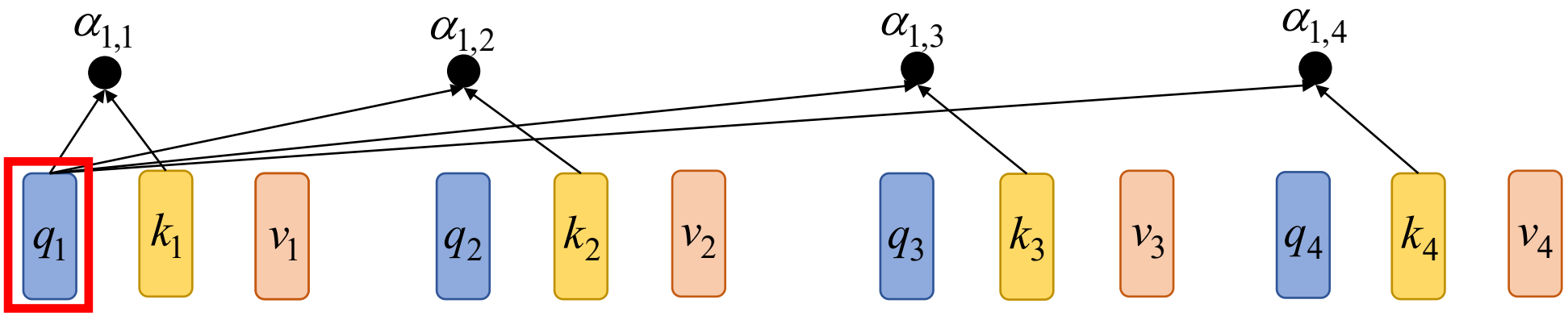
$$v_i = W^V x_i$$





注意力机制


$\alpha_{1,1}=$ k_1 q_1
$\alpha_{1,2}=$ k_2 q_1
$\alpha_{1,3}=$ k_3 q_1
$\alpha_{1,4}=$ k_4 q_1
$\alpha_{1,1}$ $\alpha_{1,2}$ $\alpha_{1,3}$ $\alpha_{1,4}$ = k_1 k_2 k_3 k_4 q_1

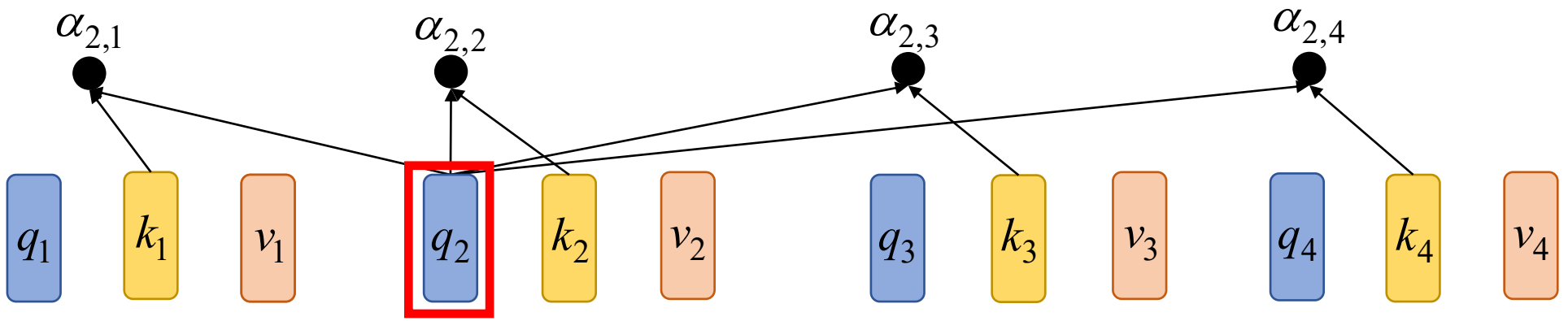

$\alpha_{1,1}$
$\alpha_{1,2}$
$\alpha_{1,3}$
$\alpha_{1,4}$
q_1 k_1 v_1 q_2 k_2 v_2 q_3 k_3 v_3 q_4 k_4 v_4

注意力机制

$\alpha_{1,1} = k_1 q_1$　　$\alpha_{1,2} = k_2 q_1$

$\alpha_{1,3} = k_3 q_1$　　$\alpha_{1,4} = k_4 q_1$

$$\begin{bmatrix}\alpha_{1,1}\\ \alpha_{1,2}\\ \alpha_{1,3}\\ \alpha_{1,4}\end{bmatrix} = \begin{bmatrix}k_1\\ k_2\\ k_3\\ k_4\end{bmatrix} q_1$$



$$\begin{bmatrix}\alpha'_{1,1} & \alpha'_{2,1} & \alpha'_{3,1} & \alpha'_{4,1}\\ \alpha'_{1,2} & \alpha'_{2,2} & \alpha'_{3,2} & \alpha'_{4,2}\\ \alpha'_{1,3} & \alpha'_{2,3} & \alpha'_{3,3} & \alpha'_{4,3}\\ \alpha'_{1,4} & \alpha'_{2,4} & \alpha'_{3,4} & \alpha'_{4,4}\end{bmatrix} = A'$$

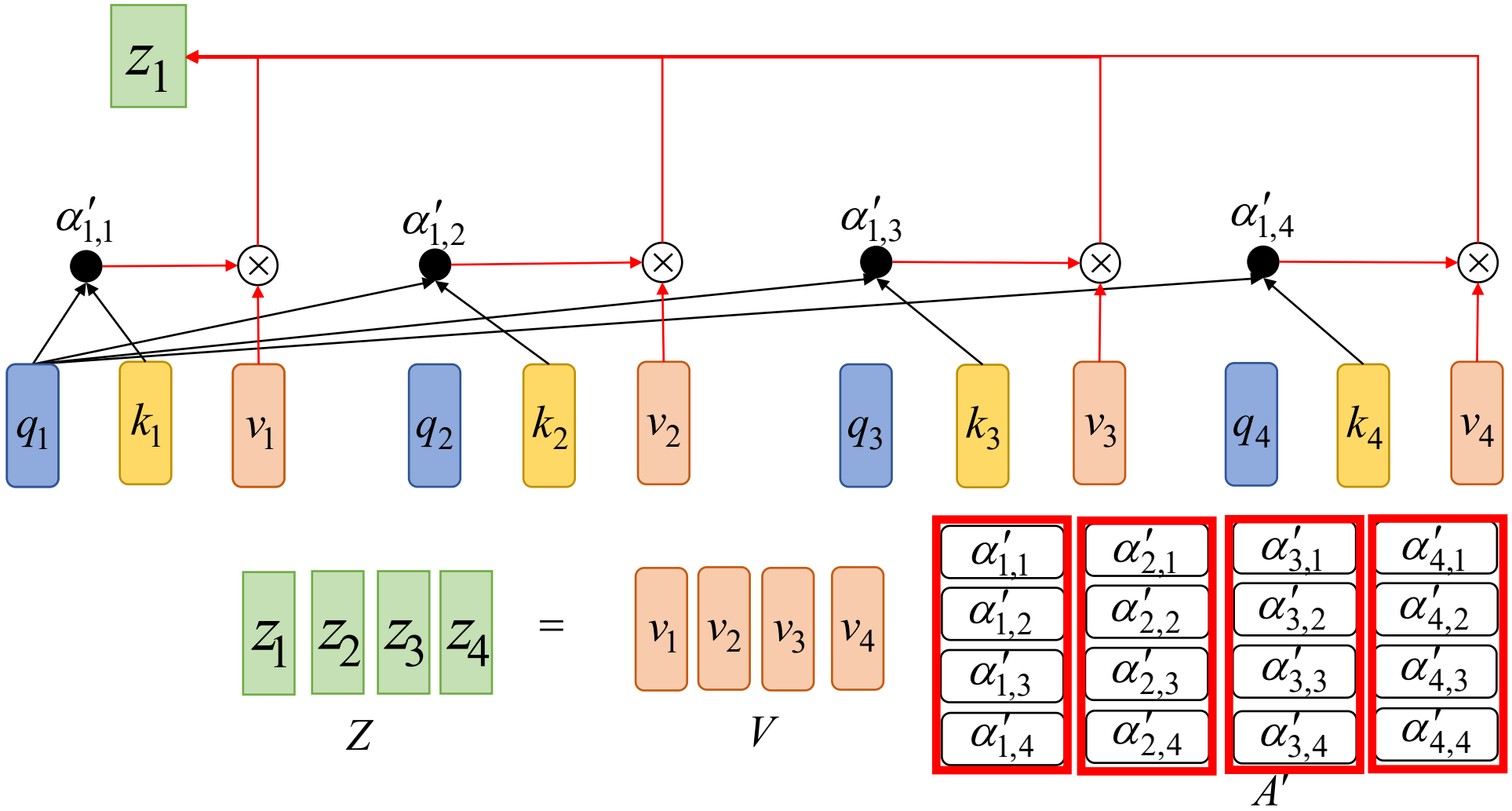
缩放

Soft-max

$$\begin{bmatrix}\alpha_{1,1} & \alpha_{2,1} & \alpha_{3,1} & \alpha_{4,1}\\ \alpha_{1,2} & \alpha_{2,2} & \alpha_{3,2} & \alpha_{4,2}\\ \alpha_{1,3} & \alpha_{2,3} & \alpha_{3,3} & \alpha_{4,3}\\ \alpha_{1,4} & \alpha_{2,4} & \alpha_{3,4} & \alpha_{4,4}\end{bmatrix} = A = K^{\mathrm{T}}Q = \begin{bmatrix}k_1\\ k_2\\ k_3\\ k_4\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}q_1 & q_2 & q_3 & q_4\end{bmatrix}$$

K^{T}　　Q

注意力机制


z_1
$\alpha'_{1,1}$
$\alpha'_{1,2}$
$\alpha'_{1,3}$
$\alpha'_{1,4}$
q_1 k_1 v_1 q_2 k_2 v_2 q_3 k_3 v_3 q_4 k_4 v_4
z_1 z_2 z_3 z_4 = v_1 v_2 v_3 v_4
Z
V
$\alpha'_{1,1}$ $\alpha'_{2,1}$ $\alpha'_{3,1}$ $\alpha'_{4,1}$
$\alpha'_{1,2}$ $\alpha'_{2,2}$ $\alpha'_{3,2}$ $\alpha'_{4,2}$
$\alpha'_{1,3}$ $\alpha'_{2,3}$ $\alpha'_{3,3}$ $\alpha'_{4,3}$
$\alpha'_{1,4}$ $\alpha'_{2,4}$ $\alpha'_{3,4}$ $\alpha'_{4,4}$
A'

注意力机制

$$Z = \begin{matrix} \\ Dogs \\ like \\ balls \end{matrix} \begin{matrix} \begin{matrix} Dogs & like & balls \end{matrix} \\ \begin{bmatrix} 0.90 & 0.03 & 0.07 \\ 0.02 & 0.95 & 0.03 \\ 0.02 & 0.01 & 0.97 \end{bmatrix} \end{matrix} \times \begin{matrix} \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1d'} \\ v_{21} & v_{22} & \cdots & v_{2d'} \\ v_{31} & v_{32} & \cdots & v_{3d'} \end{bmatrix} \begin{matrix} v_1 \\ v_2 \\ v_3 \end{matrix} \\ V \end{matrix}$$

$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

$$QK^T = \begin{matrix} \\ Dogs \\ like \\ balls \end{matrix} \begin{matrix} \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1d'} \\ q_{21} & q_{22} & \cdots & q_{2d'} \\ q_{31} & q_{32} & \cdots & q_{3d'} \end{bmatrix} \begin{matrix} q_1 \\ q_2 \\ q_3 \end{matrix} \times \\ Q \end{matrix} \begin{matrix} \begin{matrix} Dogs & like & balls \end{matrix} \\ \begin{bmatrix} k_{11} & k_{21} & k_{31} \\ k_{12} & k_{22} & k_{32} \\ \vdots & \vdots & \vdots \\ k_{1d'} & k_{2d'} & k_{3d'} \end{bmatrix} \\ \begin{matrix} k_1 & k_2 & k_3 \end{matrix} \\ K^T \end{matrix}$$

需要学习的参数

$$Q = \begin{matrix} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ x_{31} & x_{32} & \cdots & x_{3d} \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \\ X \end{matrix} \times \begin{matrix} \begin{bmatrix} w_{11}^Q & w_{12}^Q & \cdots & w_{1d'}^Q \\ w_{21}^Q & w_{22}^Q & \cdots & w_{2d'}^Q \\ \vdots & \vdots & \ddots & \vdots \\ w_{d1}^Q & w_{d2}^Q & \cdots & w_{dd'}^Q \end{bmatrix} \\ W^Q \end{matrix}$$

$$K = \begin{matrix} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ x_{31} & x_{32} & \cdots & x_{3d} \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \\ X \end{matrix} \times \begin{matrix} \begin{bmatrix} w_{11}^K & w_{12}^K & \cdots & w_{1d'}^K \\ w_{21}^K & w_{22}^K & \cdots & w_{2d'}^K \\ \vdots & \vdots & \ddots & \vdots \\ w_{d1}^K & w_{d2}^K & \cdots & w_{dd'}^K \end{bmatrix} \\ W^K \end{matrix}$$

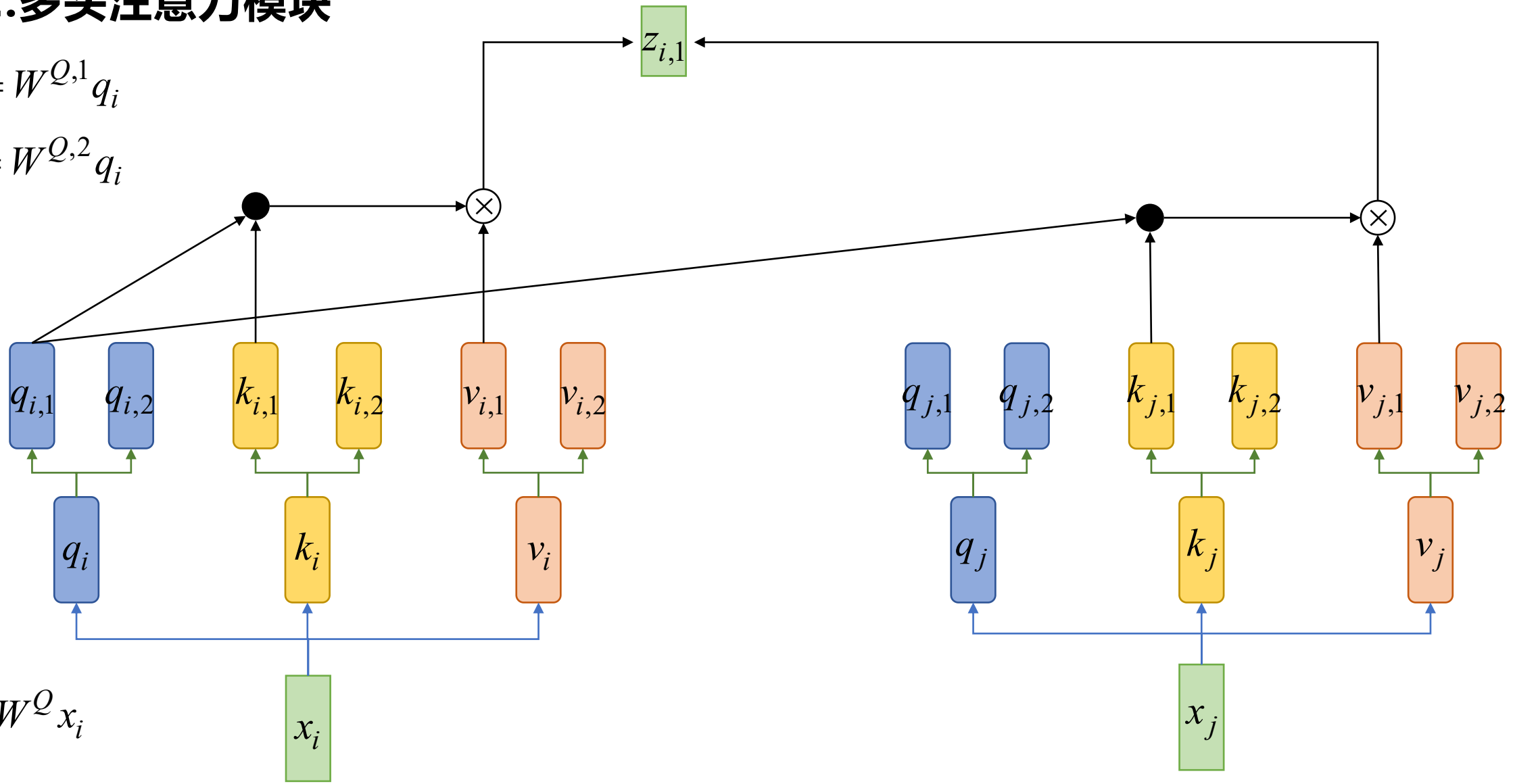
$$V = \begin{matrix} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ x_{31} & x_{32} & \cdots & x_{3d} \end{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \\ X \end{matrix} \times \begin{matrix} \begin{bmatrix} w_{11}^V & w_{12}^V & \cdots & w_{1d'}^V \\ w_{21}^V & w_{22}^V & \cdots & w_{2d'}^V \\ \vdots & \vdots & \ddots & \vdots \\ w_{d1}^V & w_{d2}^V & \cdots & w_{dd'}^V \end{bmatrix} \\ W^V \end{matrix}$$

注意力机制

2.多头注意力模块

$$q_{i,1} = W^{Q,1} q_i$$

$$q_{i,2} = W^{Q,2} q_i$$

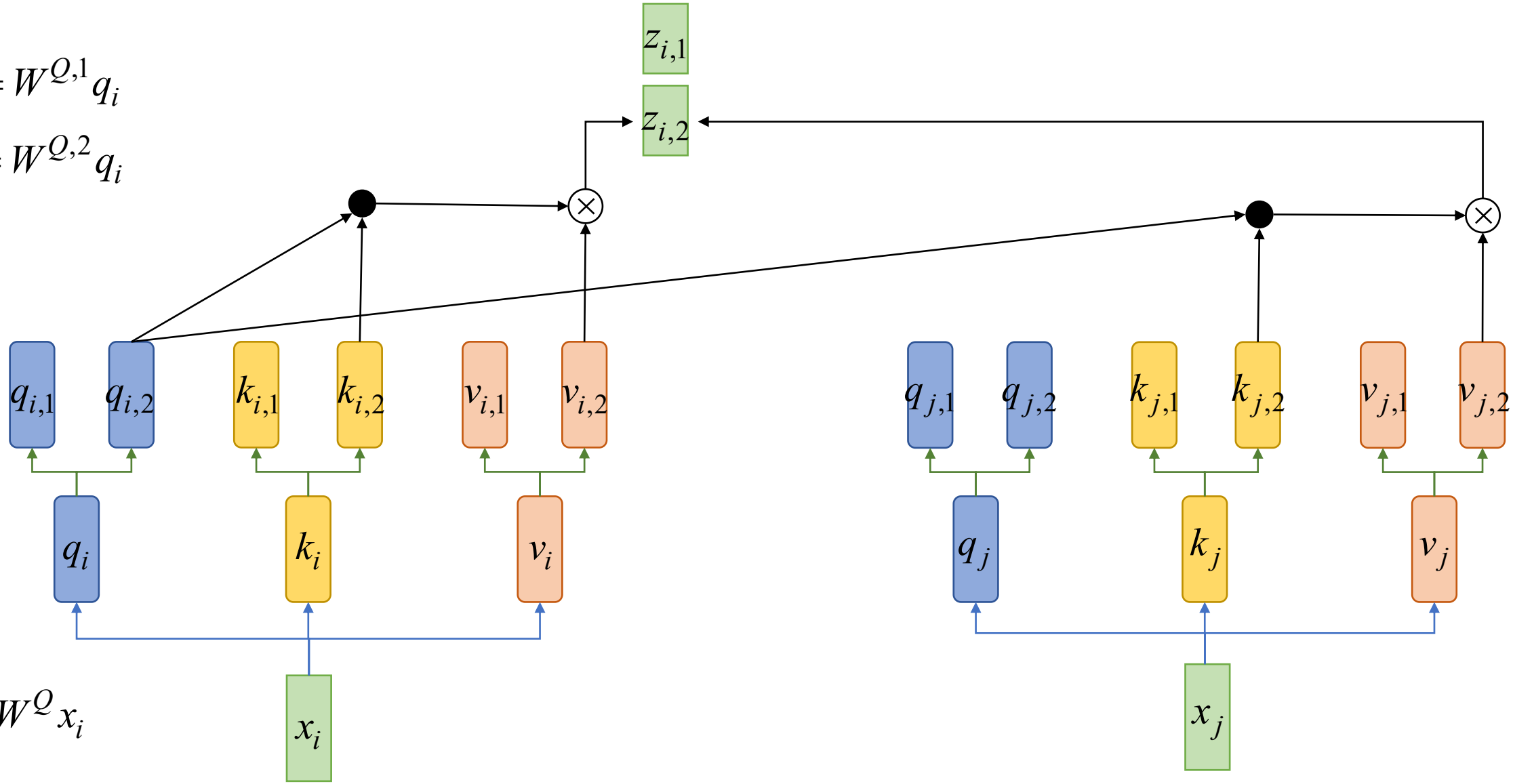


$$q_i = W^Q x_i$$

注意力机制

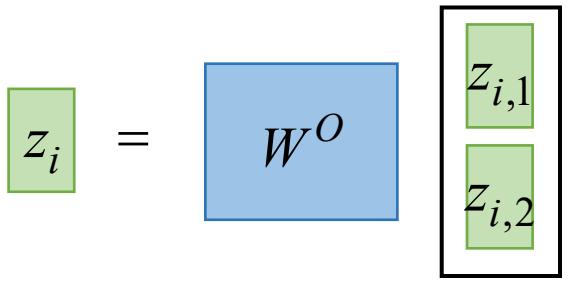
$$q_{i,1} = W^{Q,1} q_i$$

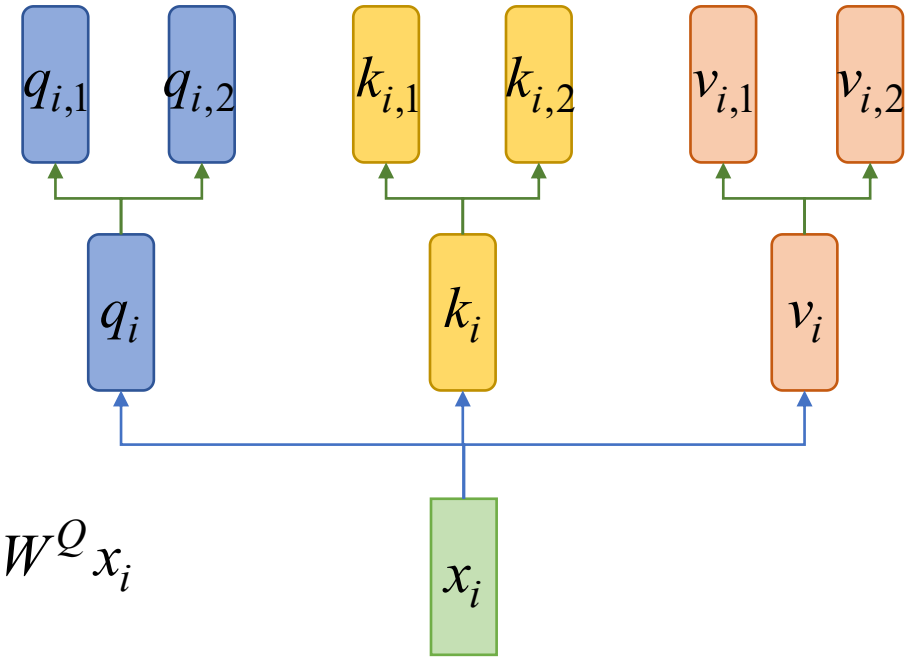
$$q_{i,2} = W^{Q,2} q_i$$



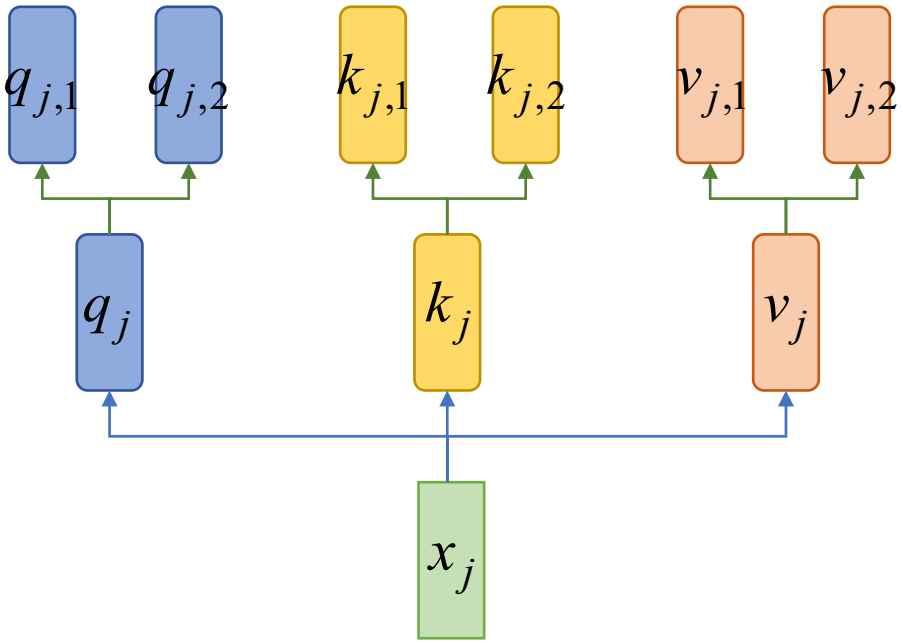
$$q_i = W^Q x_i$$

注意力机制



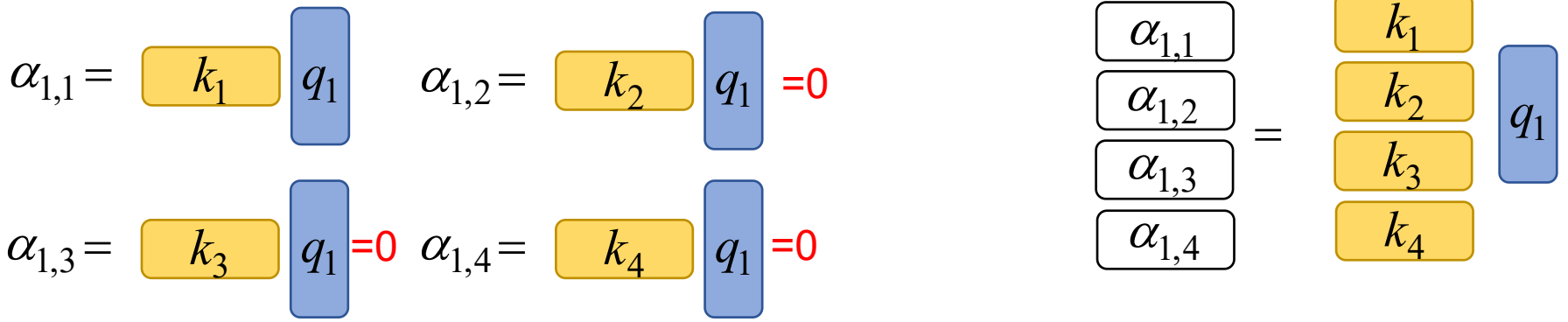


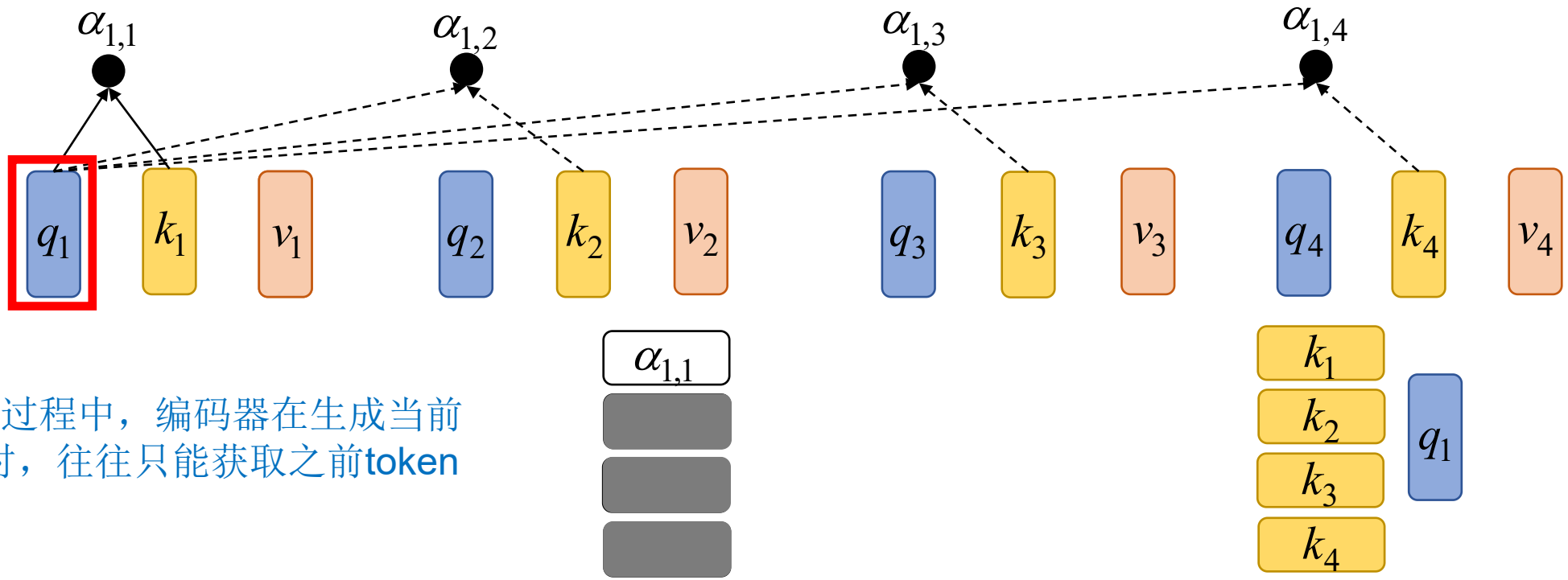
$q_i = W^Q x_i$



注意力机制

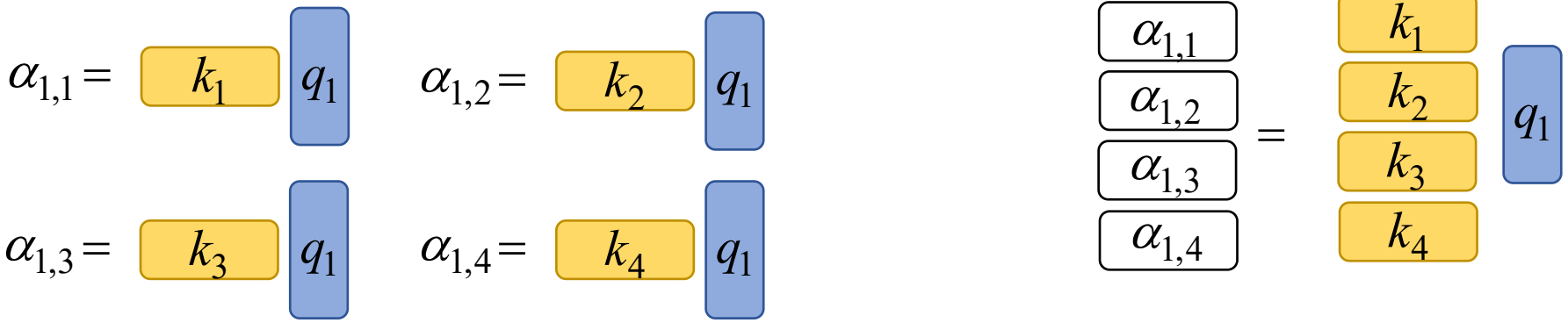
3.掩码多头注意力模块



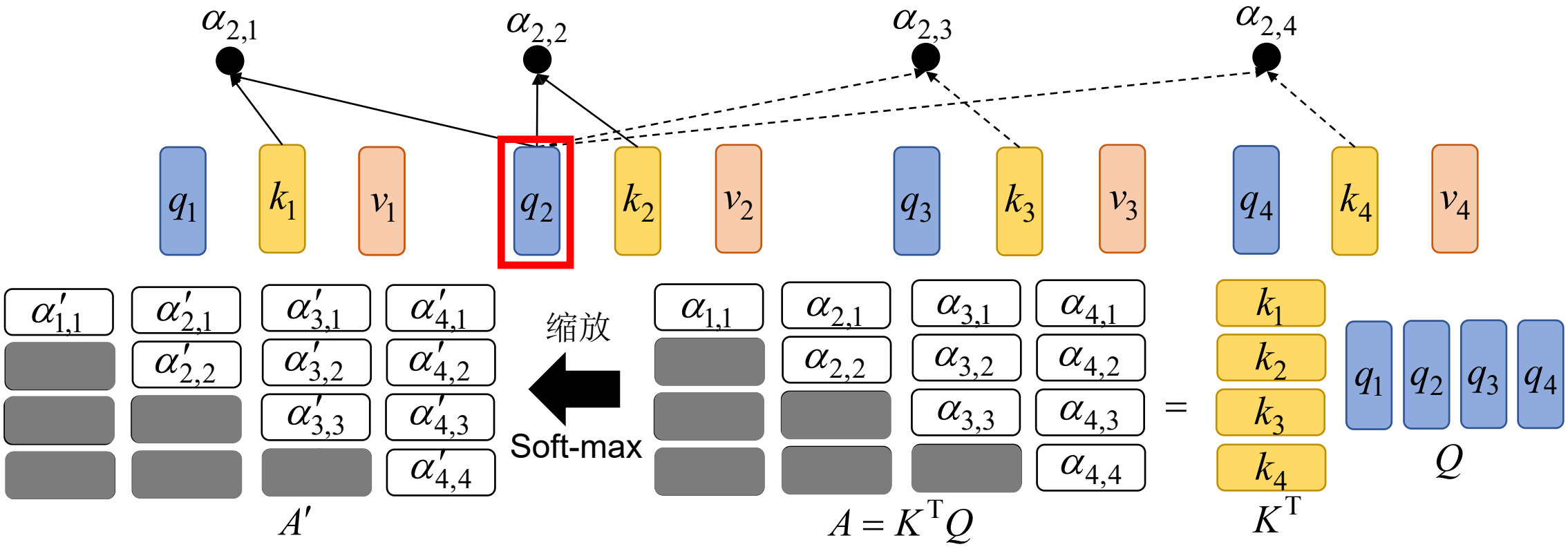


在推理过程中，编码器在生成当前token时，往往只能获取之前token的信息

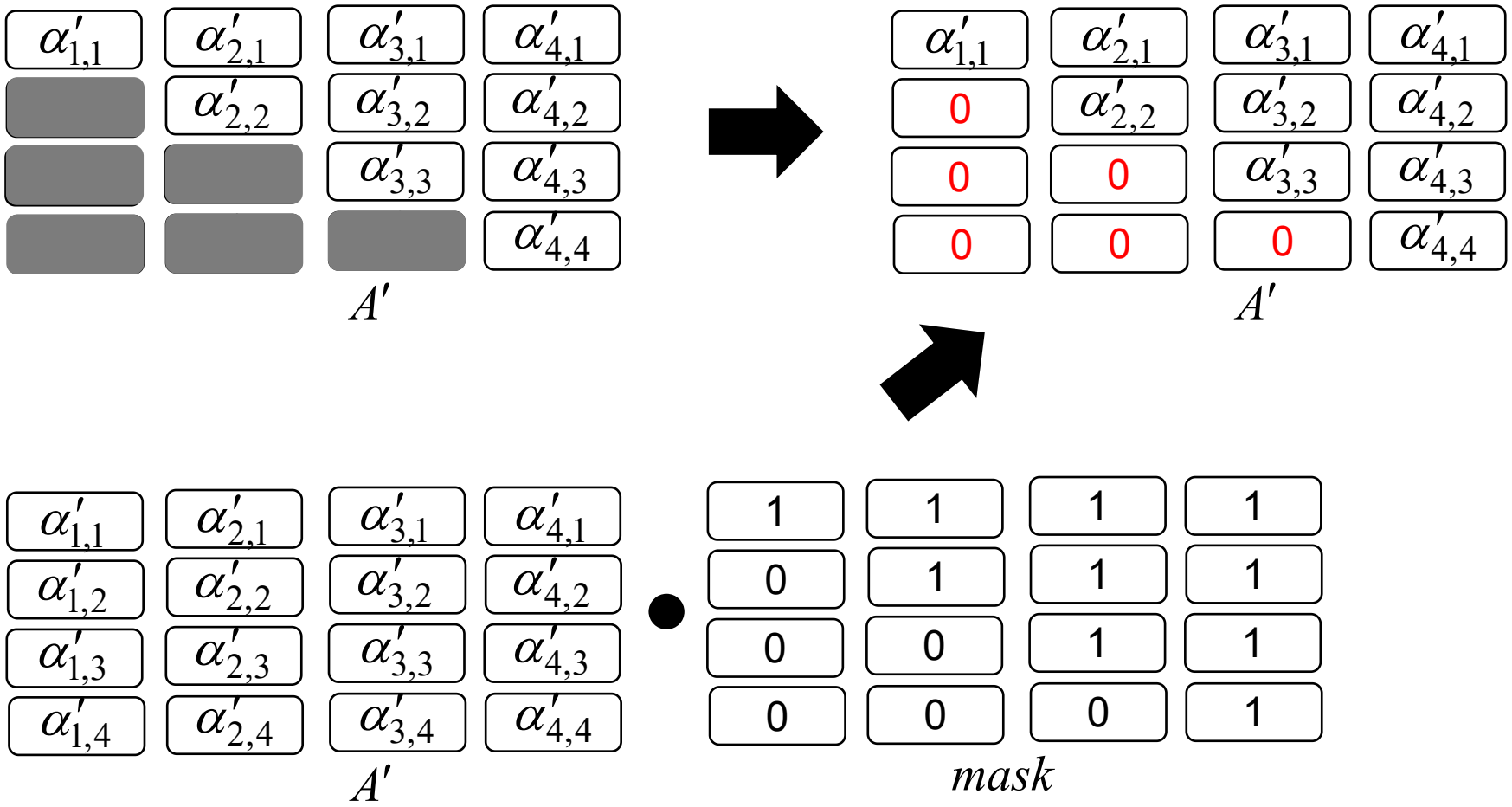
注意力机制


$\alpha_{1,1}=$ k_1 q_1
$\alpha_{1,2}=$ k_2 q_1
$\alpha_{1,3}=$ k_3 q_1
$\alpha_{1,4}=$ k_4 q_1

$\alpha_{1,1}$
$\alpha_{1,2}$
$\alpha_{1,3}$
$\alpha_{1,4}$
=
k_1
k_2
k_3
k_4
q_1

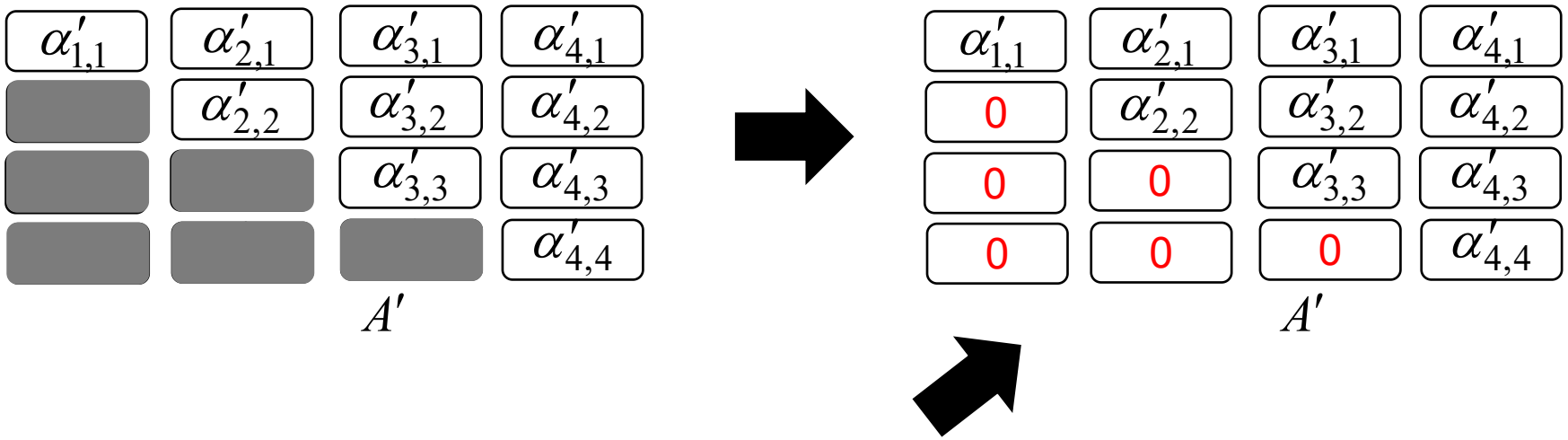

$\alpha_{2,1}$ $\alpha_{2,2}$ $\alpha_{2,3}$ $\alpha_{2,4}$
q_1 k_1 v_1 q_2 k_2 v_2 q_3 k_3 v_3 q_4 k_4 v_4
缩放
Soft-max
A'
$A = K^{\mathrm{T}}Q$
K^{T}
Q

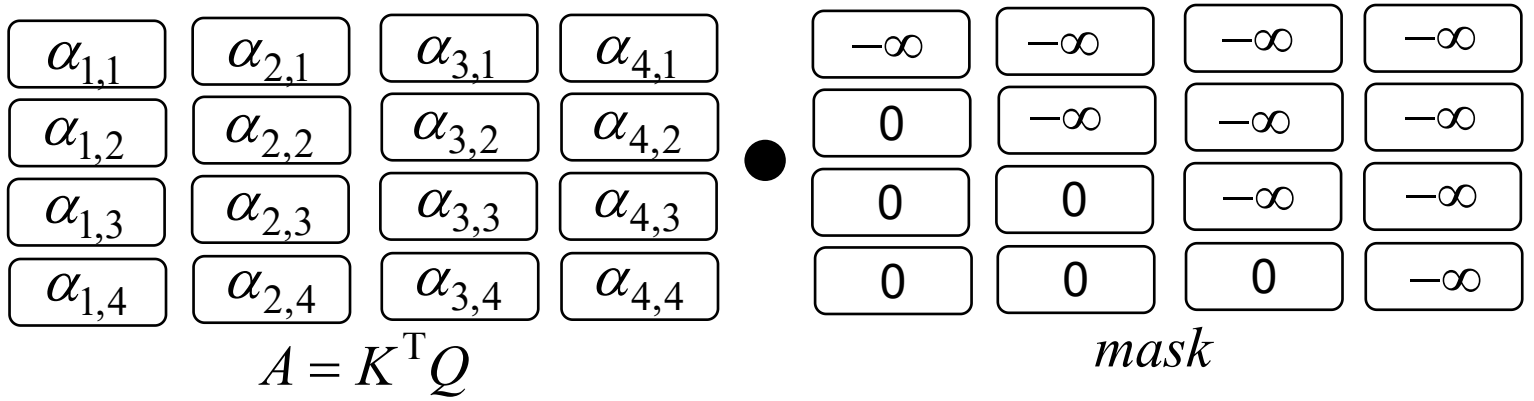
注意力机制



构造掩码矩阵

注意力机制


$\alpha'_{1,1}$ $\alpha'_{2,1}$ $\alpha'_{3,1}$ $\alpha'_{4,1}$
$\alpha'_{2,2}$ $\alpha'_{3,2}$ $\alpha'_{4,2}$
$\alpha'_{3,3}$ $\alpha'_{4,3}$
$\alpha'_{4,4}$
A'
$\alpha'_{1,1}$ $\alpha'_{2,1}$ $\alpha'_{3,1}$ $\alpha'_{4,1}$
0 $\alpha'_{2,2}$ $\alpha'_{3,2}$ $\alpha'_{4,2}$
0 0 $\alpha'_{3,3}$ $\alpha'_{4,3}$
0 0 0 $\alpha'_{4,4}$
A'


$\alpha_{1,1}$ $\alpha_{2,1}$ $\alpha_{3,1}$ $\alpha_{4,1}$
$\alpha_{1,2}$ $\alpha_{2,2}$ $\alpha_{3,2}$ $\alpha_{4,2}$
$\alpha_{1,3}$ $\alpha_{2,3}$ $\alpha_{3,3}$ $\alpha_{4,3}$
$\alpha_{1,4}$ $\alpha_{2,4}$ $\alpha_{3,4}$ $\alpha_{4,4}$
$A = K^{\mathrm{T}}Q$
$-\infty$ $-\infty$ $-\infty$ $-\infty$
0 $-\infty$ $-\infty$ $-\infty$
0 0 $-\infty$ $-\infty$
0 0 0 $-\infty$
$mask$

注意力机制

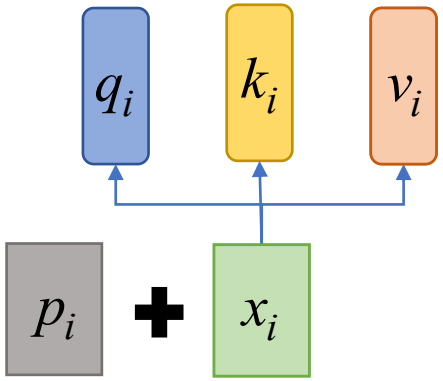
4.位置编码

在自注意力模块中缺乏句子词序

位置编码：每个位置有一个独一无二的位置向量

手工设计的

也可以是学习得到的



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

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

$$P = \begin{bmatrix} sin(\frac{pos}{10000^0}) & cos(\frac{pos}{10000^0}) & sin(\frac{pos}{10000^{2/3}}) \\ sin(\frac{pos}{10000^0}) & cos(\frac{pos}{10000^0}) & sin(\frac{pos}{10000^{2/3}}) \\ sin(\frac{pos}{10000^0}) & cos(\frac{pos}{10000^0}) & sin(\frac{pos}{10000^{2/3}}) \end{bmatrix}$$

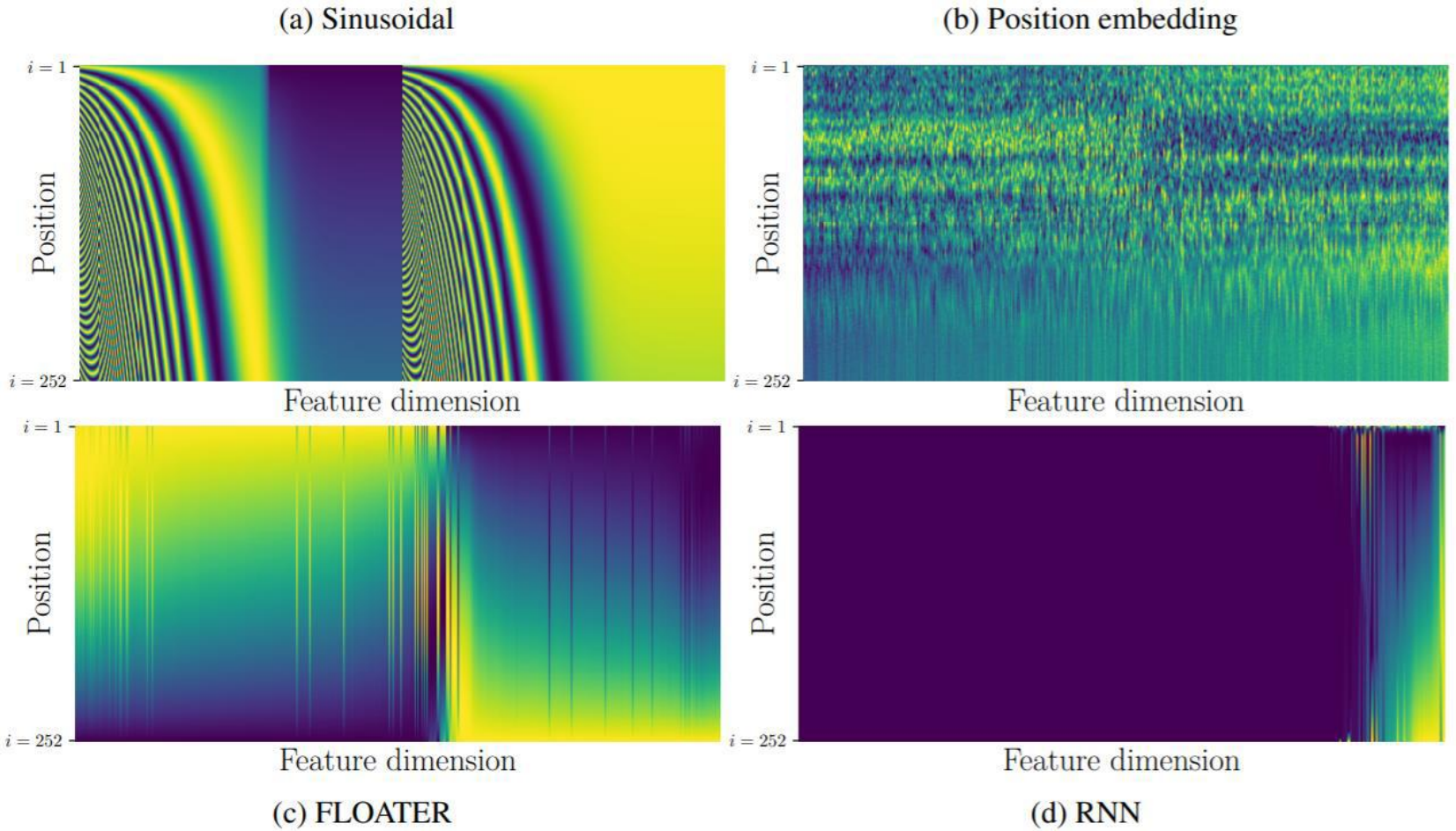
$$P = \begin{bmatrix} sin(0) & cos(0) & sin(\frac{0}{10000^{2/3}}) \\ sin(1) & cos(1) & sin(\frac{1}{10000^{2/3}}) \\ sin(2) & cos(2) & sin(\frac{2}{10000^{2/3}}) \end{bmatrix}$$

注意力机制

https://arxiv.org/abs/2003.09229

Table 1. Comparing position representation methods

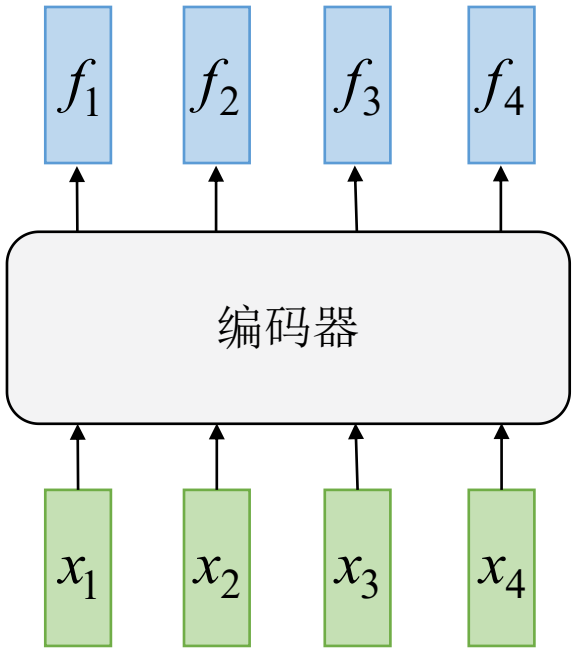
Methods	Inductive	Data-Driven	Parameter Efficient
Sinusoidal (Vaswani et al., 2017)	✓	✗	✓
Embedding (Devlin et al., 2018)	✗	✓	✗
Relative (Shaw et al., 2018)	✗	✓	✓
This paper	✓	✓	✓

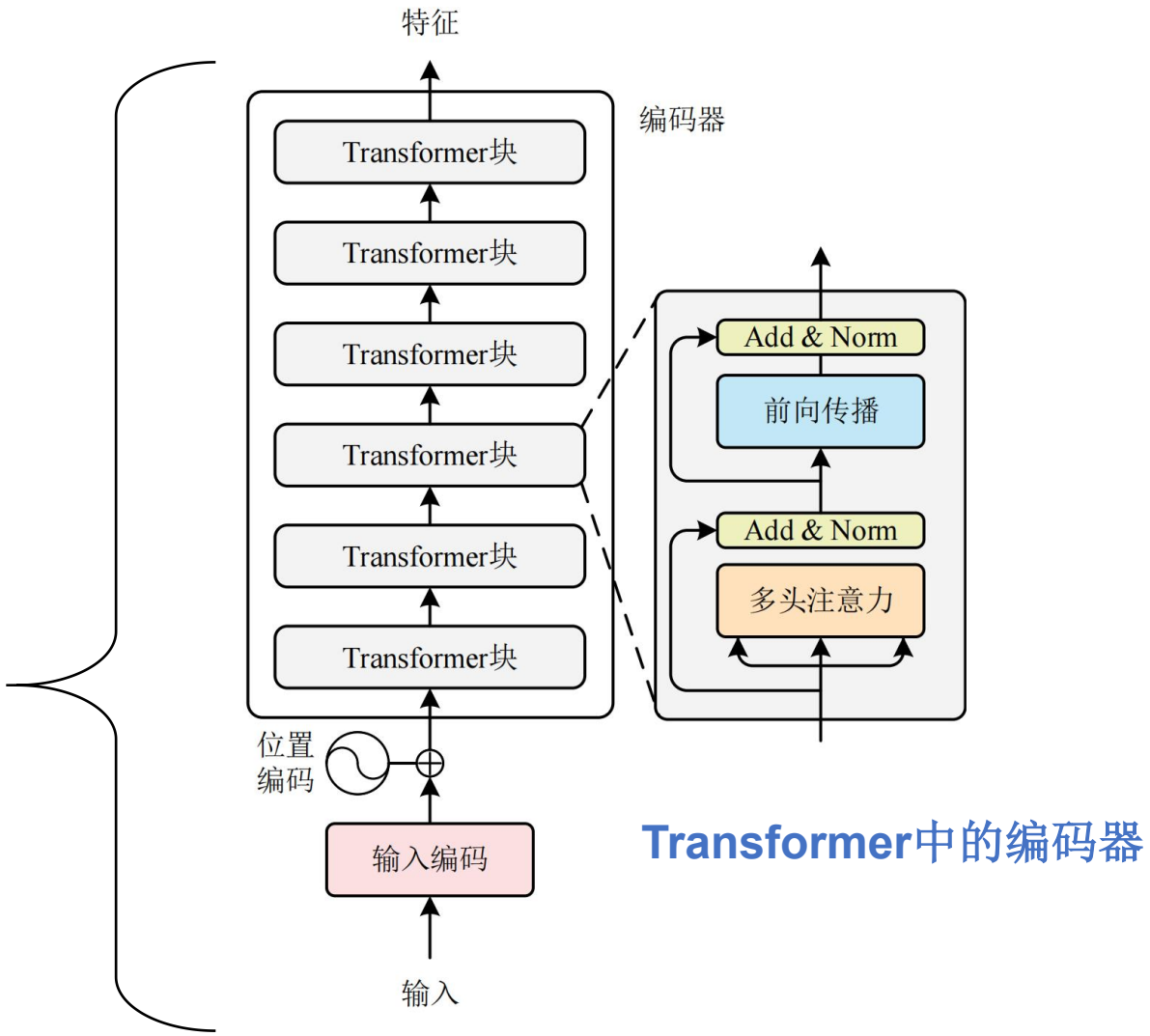


目录

编码器-解码器结构

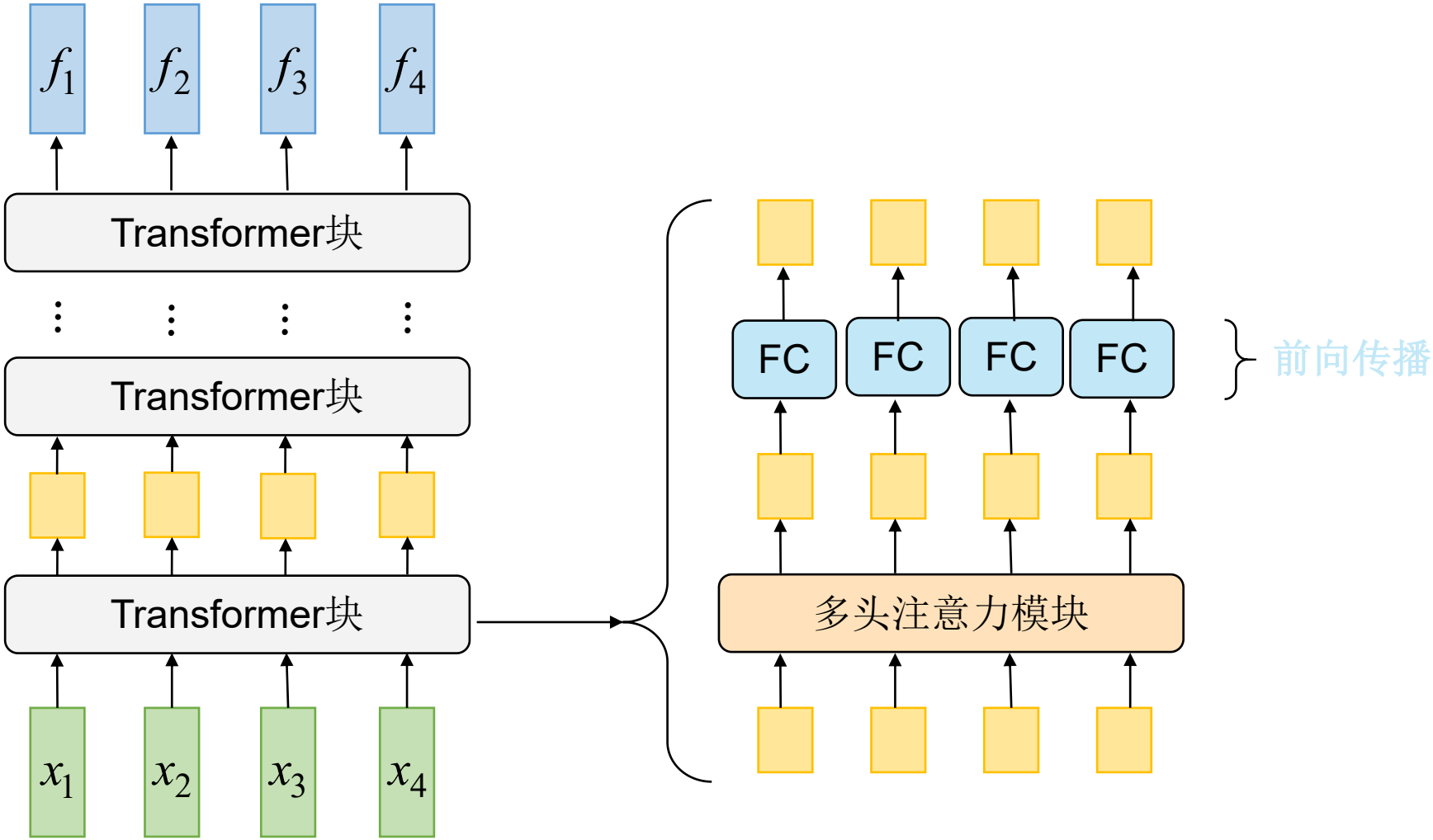
1.编码器



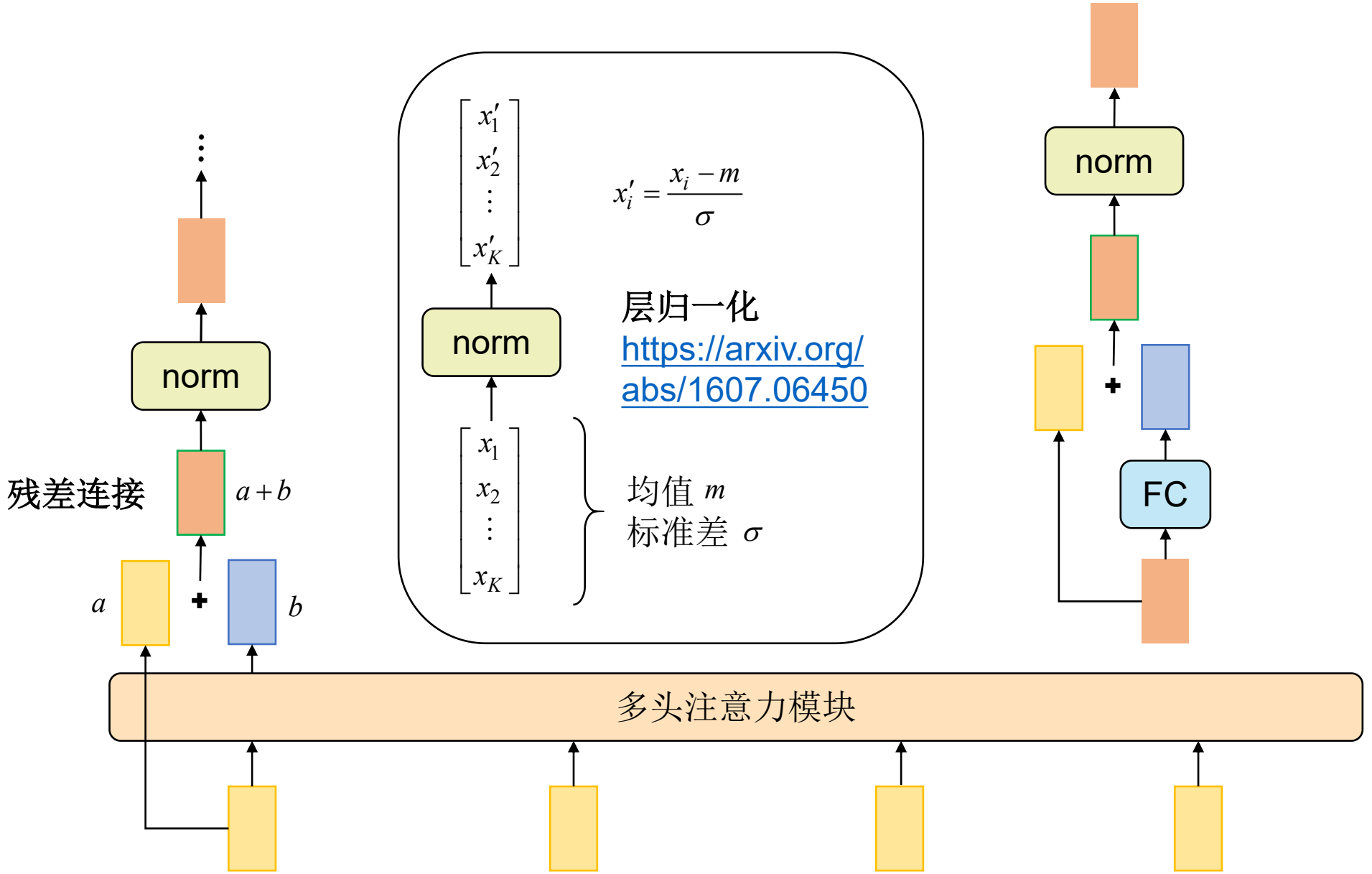


Transformer中的编码器

编码器-解码器结构

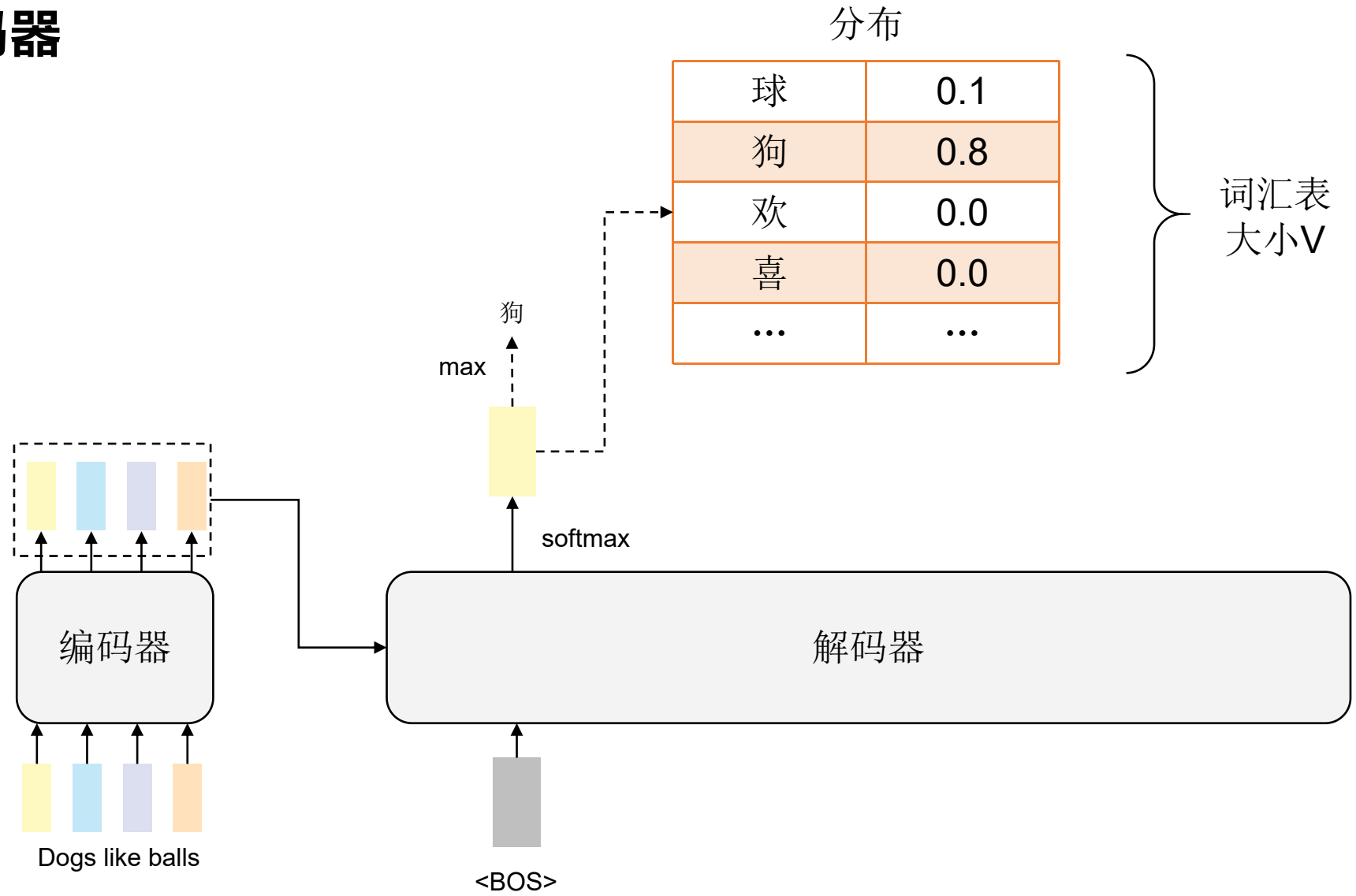

f_1
f_2
f_3
f_4
Transformer块
Transformer块
Transformer块
x_1
x_2
x_3
x_4
FC
FC
FC
FC
前向传播
多头注意力模块

编码器-解码器结构

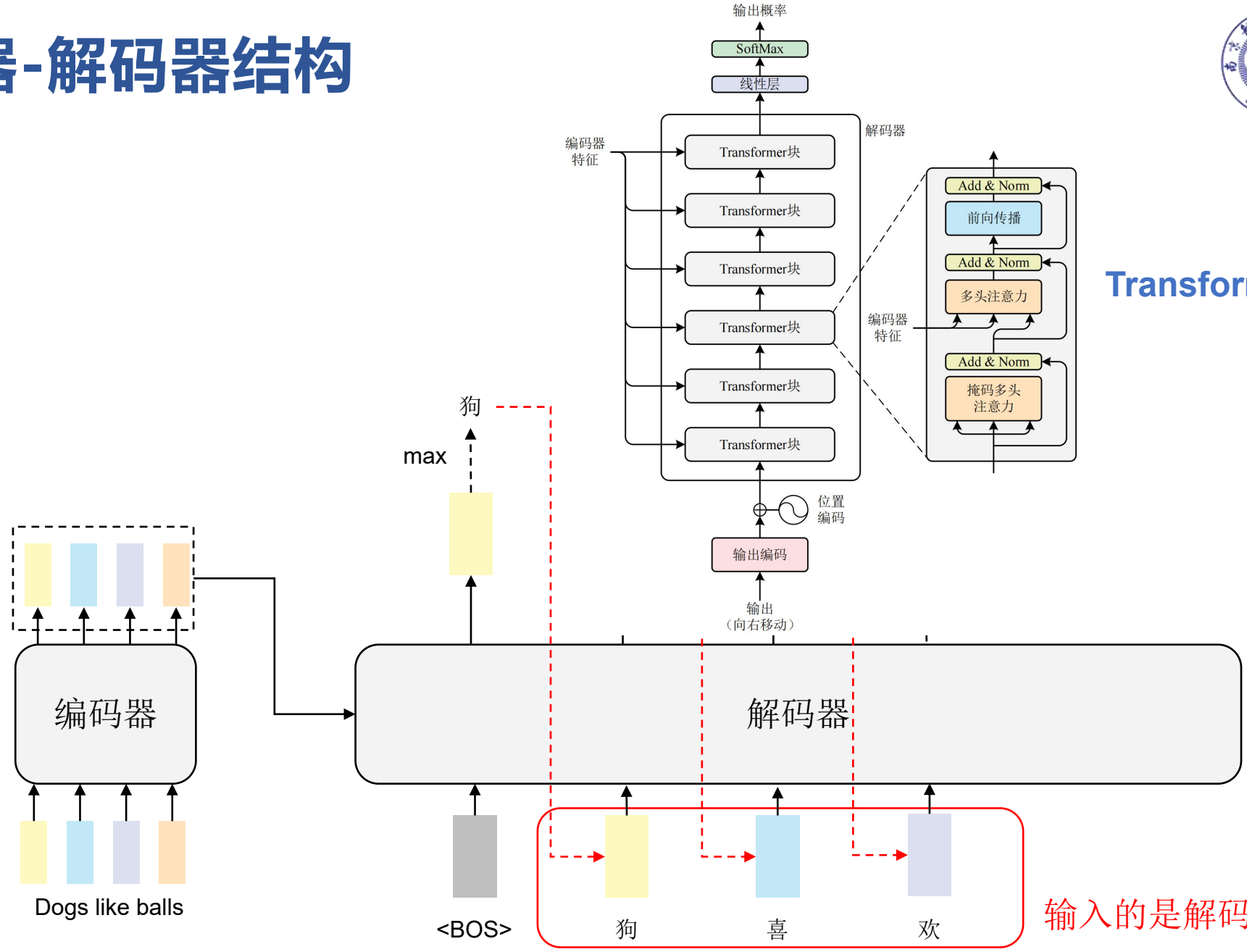

norm
残差连接
$a+b$
a
b
多头注意力模块
x'_1
x'_2
x'_K
x_1
x_2
x_K
$x'_i = \frac{x_i - m}{\sigma}$
层归一化
https://arxiv.org/abs/1607.06450
均值 m
标准差 σ
norm
FC

编码器-解码器结构

2.解码器



编码器-解码器结构

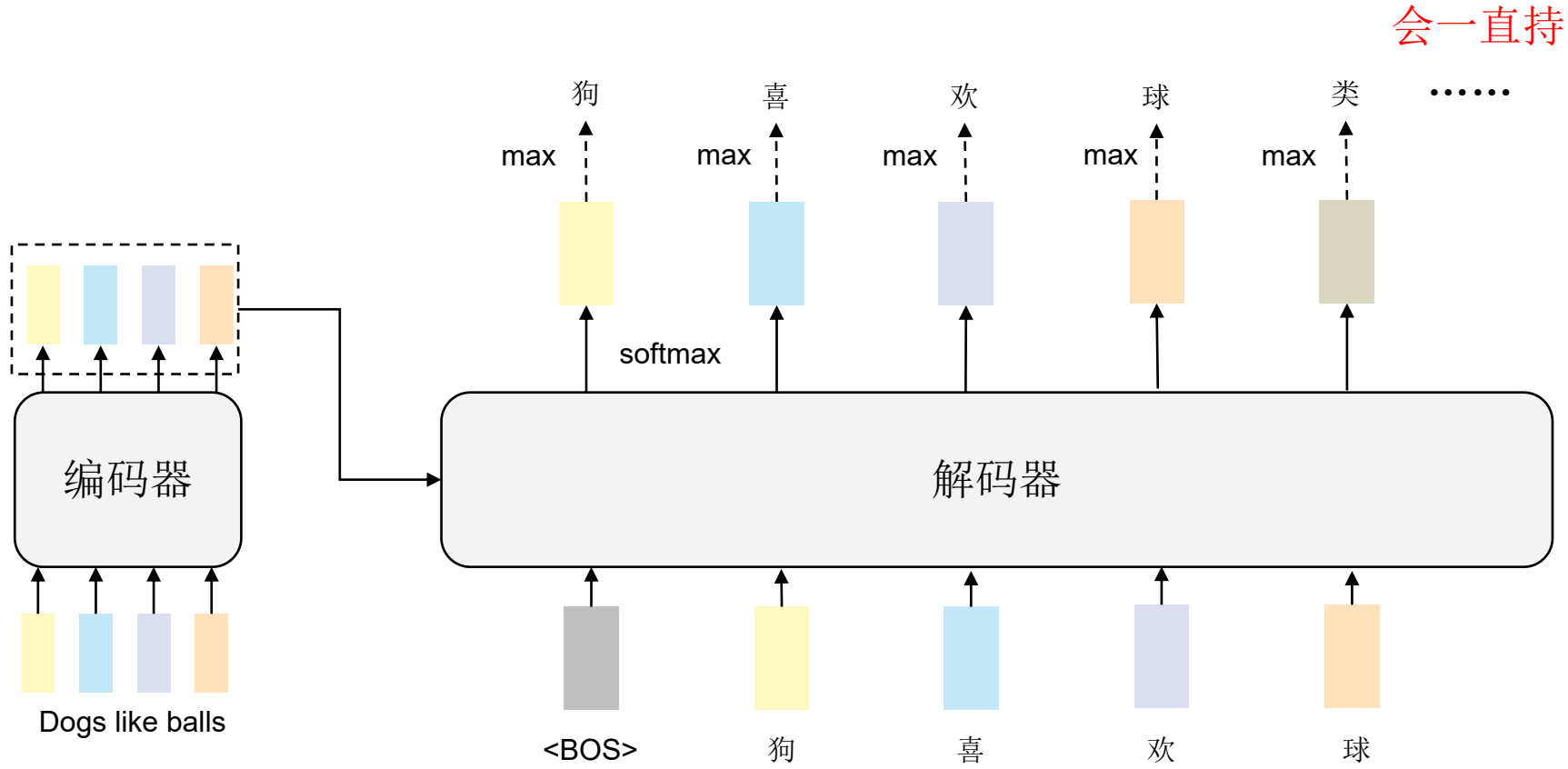


Transformer中的解码器

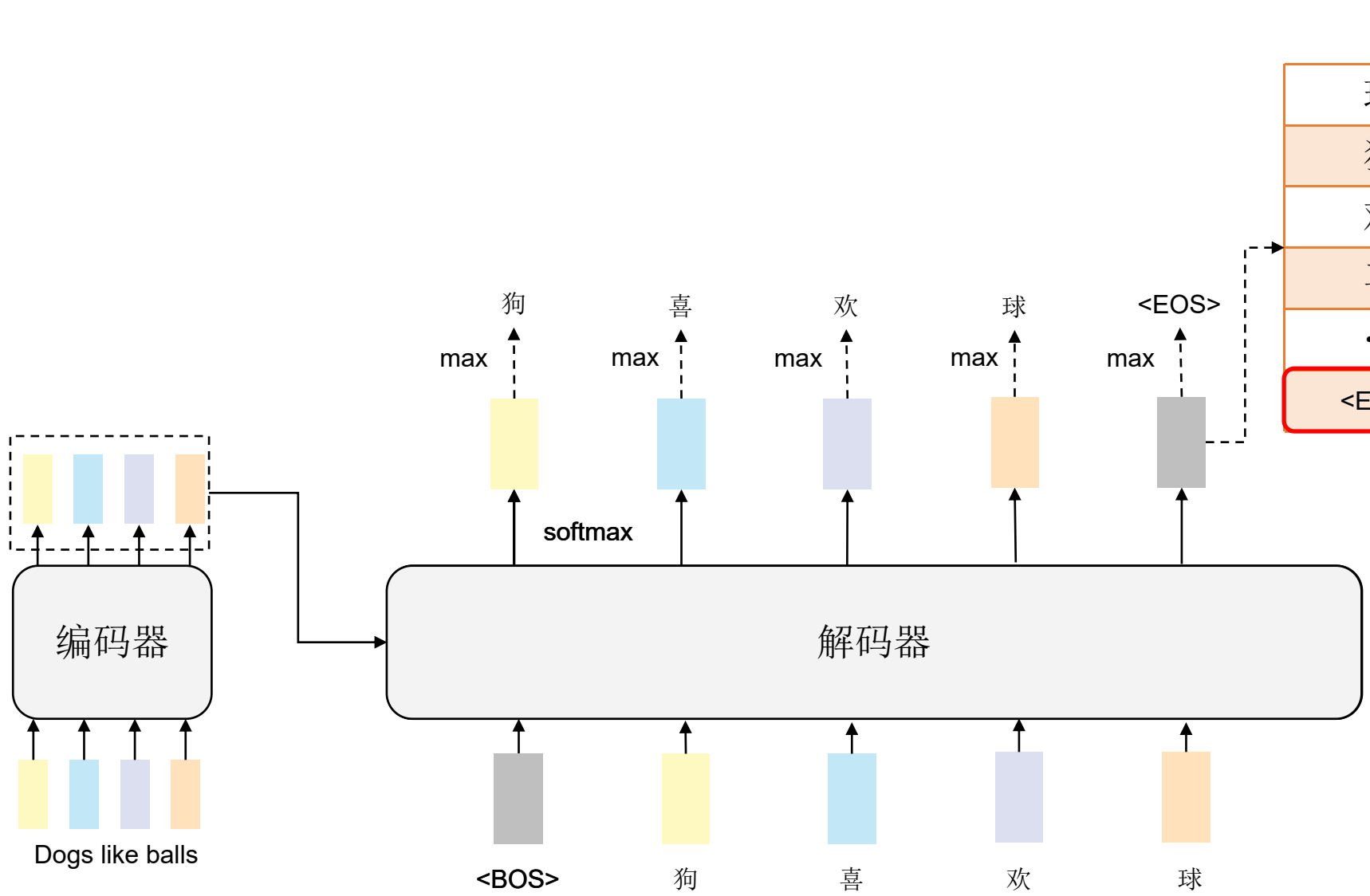
输入的是解码器的输出

编码器-解码器结构

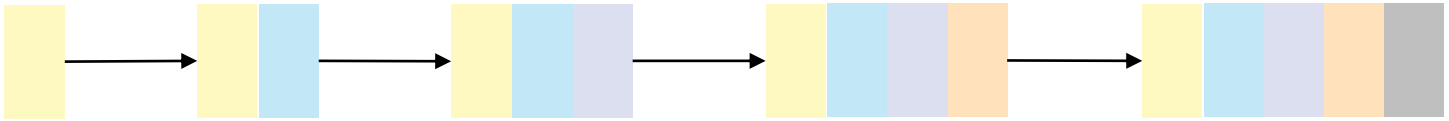
- **解码器需要自己决定生成的目标句的长度**
- **实际上：**机器并不能确定正确的输出长度



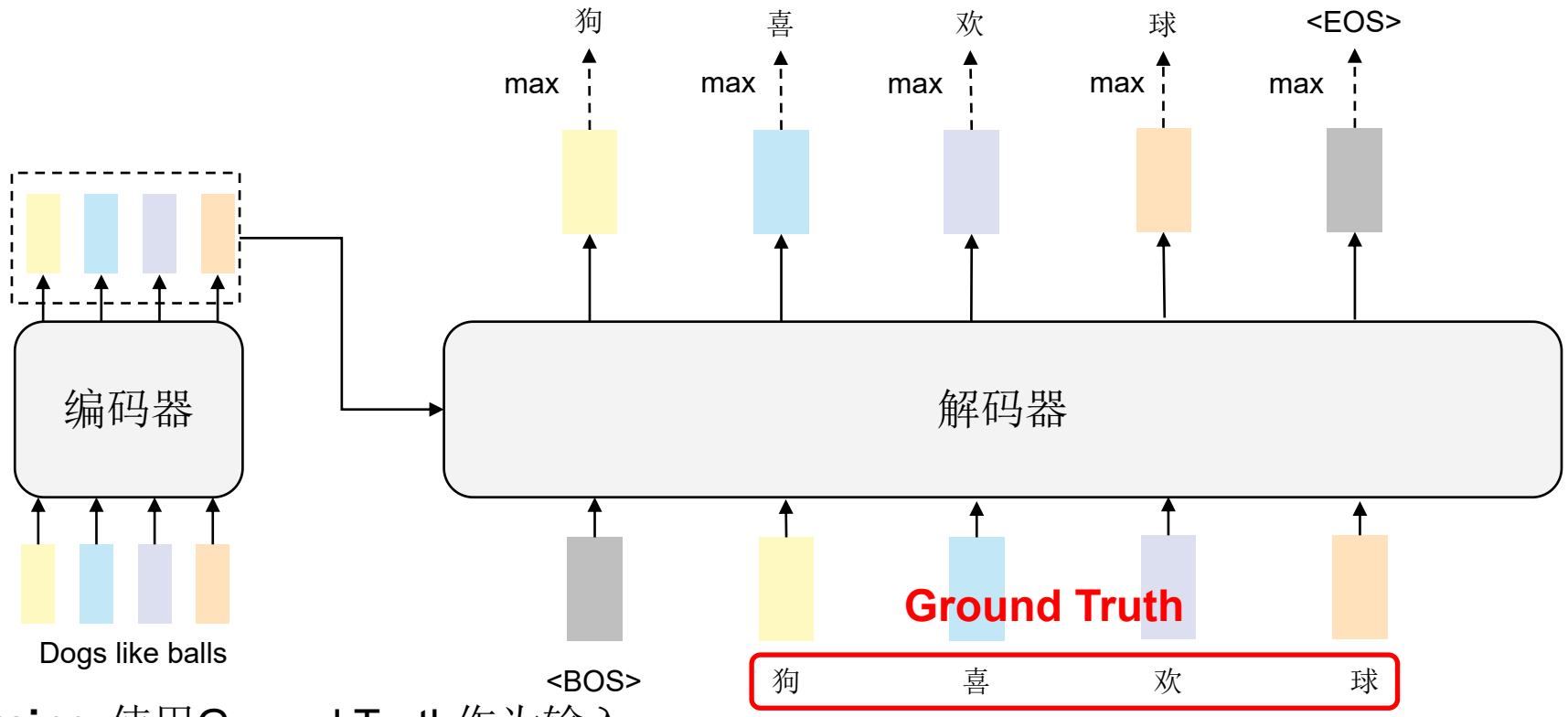
编码器-解码器结构


分布
球 0.1
狗 0.0
欢 0.0
喜 0.0
… …
<EOS> 0.8
狗
喜
欢
球
<EOS>
max
max
max
max
max
softmax
编码器
解码器
Dogs like balls
<BOS>
狗
喜
欢
球

编码器-解码器结构

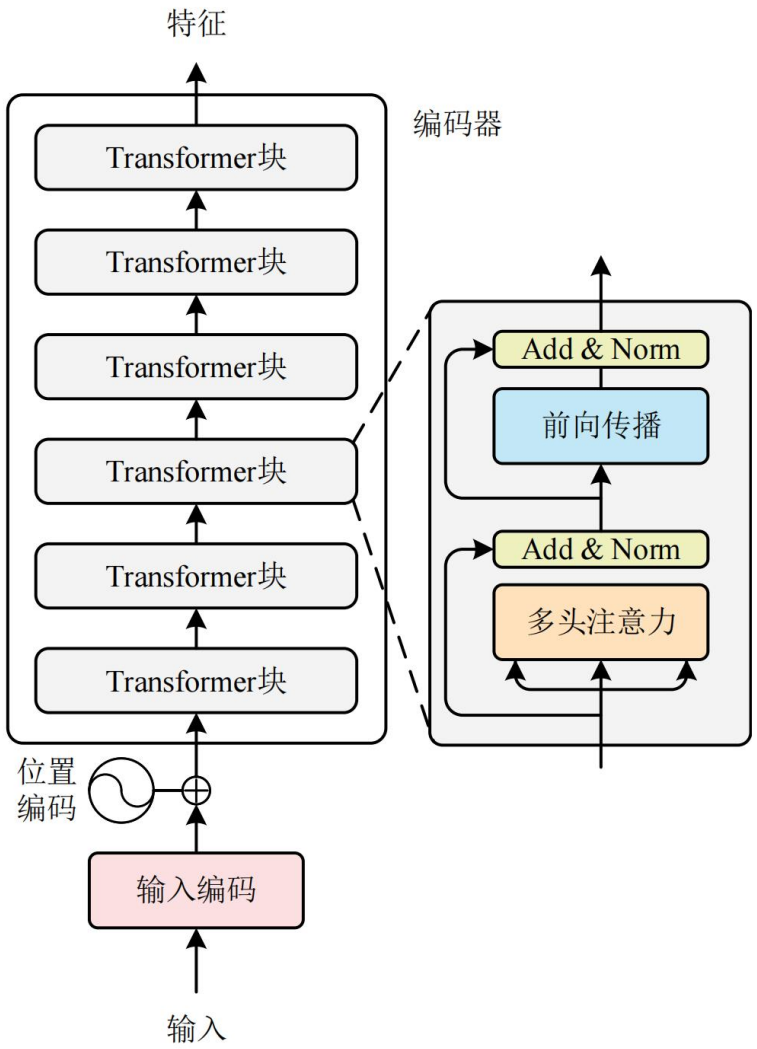


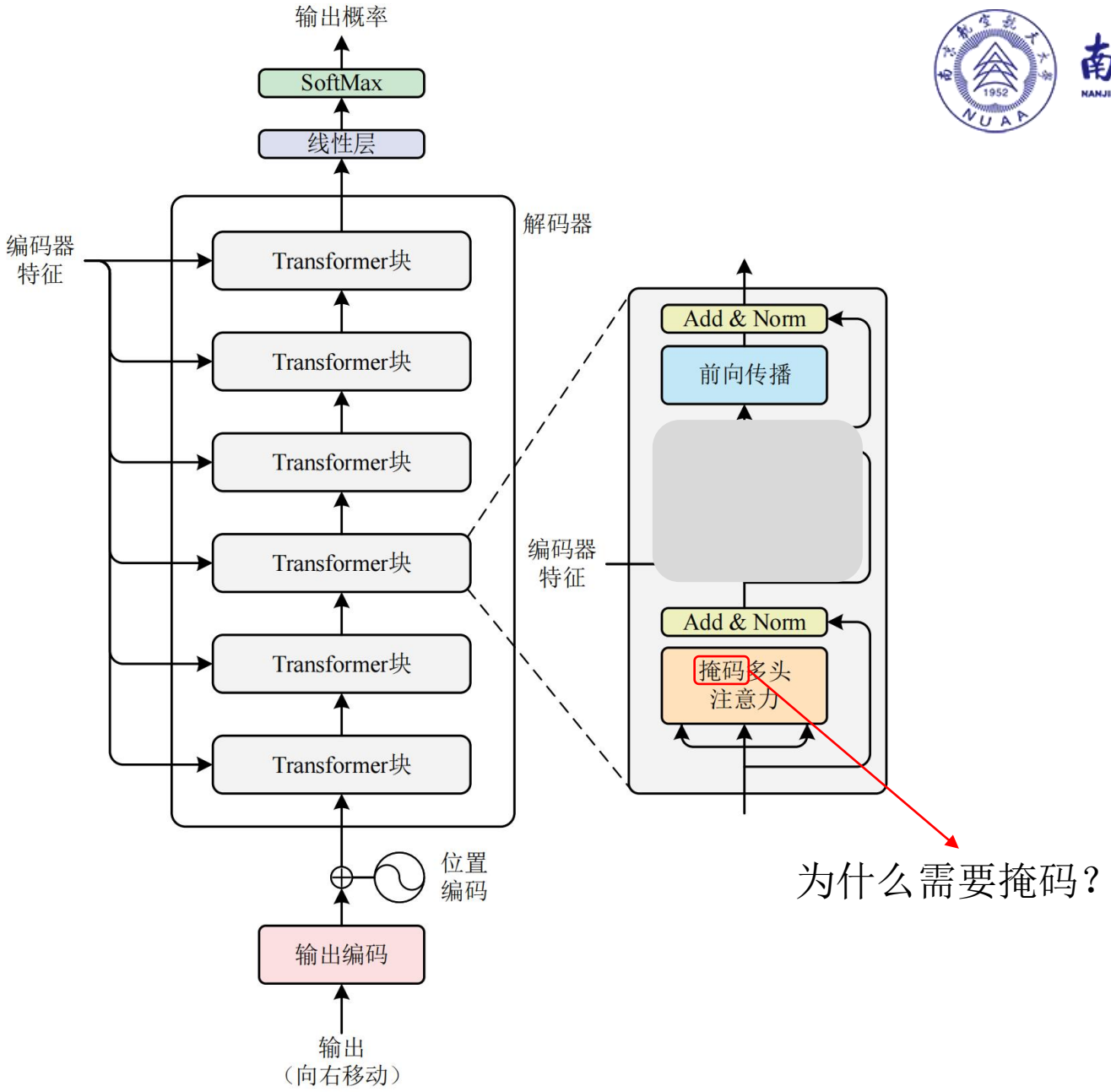
在训练的过程中，不仅降低模型并行程度，使得训练时间变长，而且训练更加困难



Teacher Forcing:使用Ground Truth作为输入

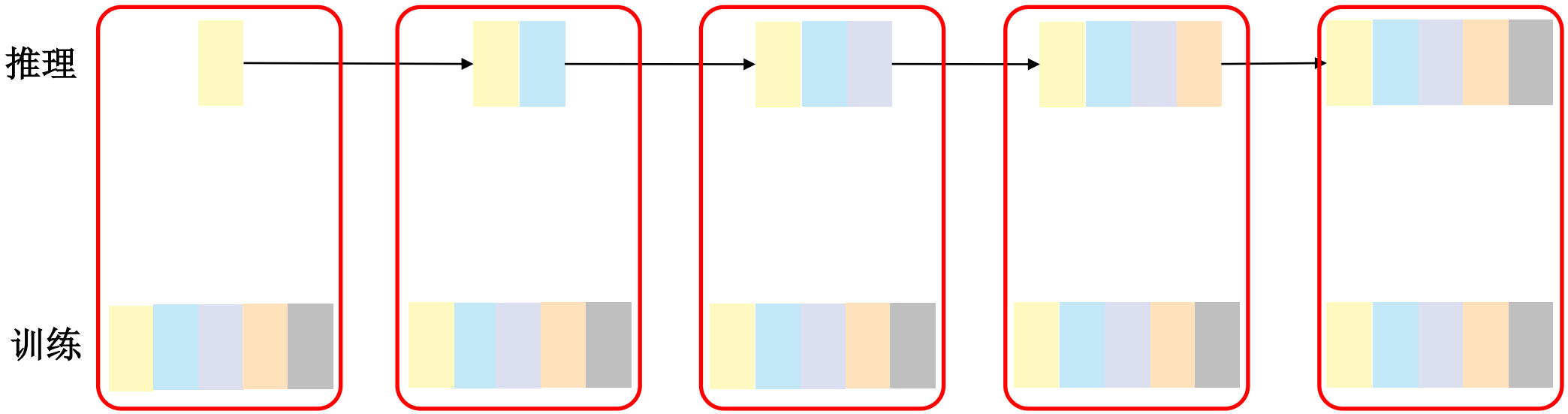
编码器-解码器结构





为什么需要掩码？

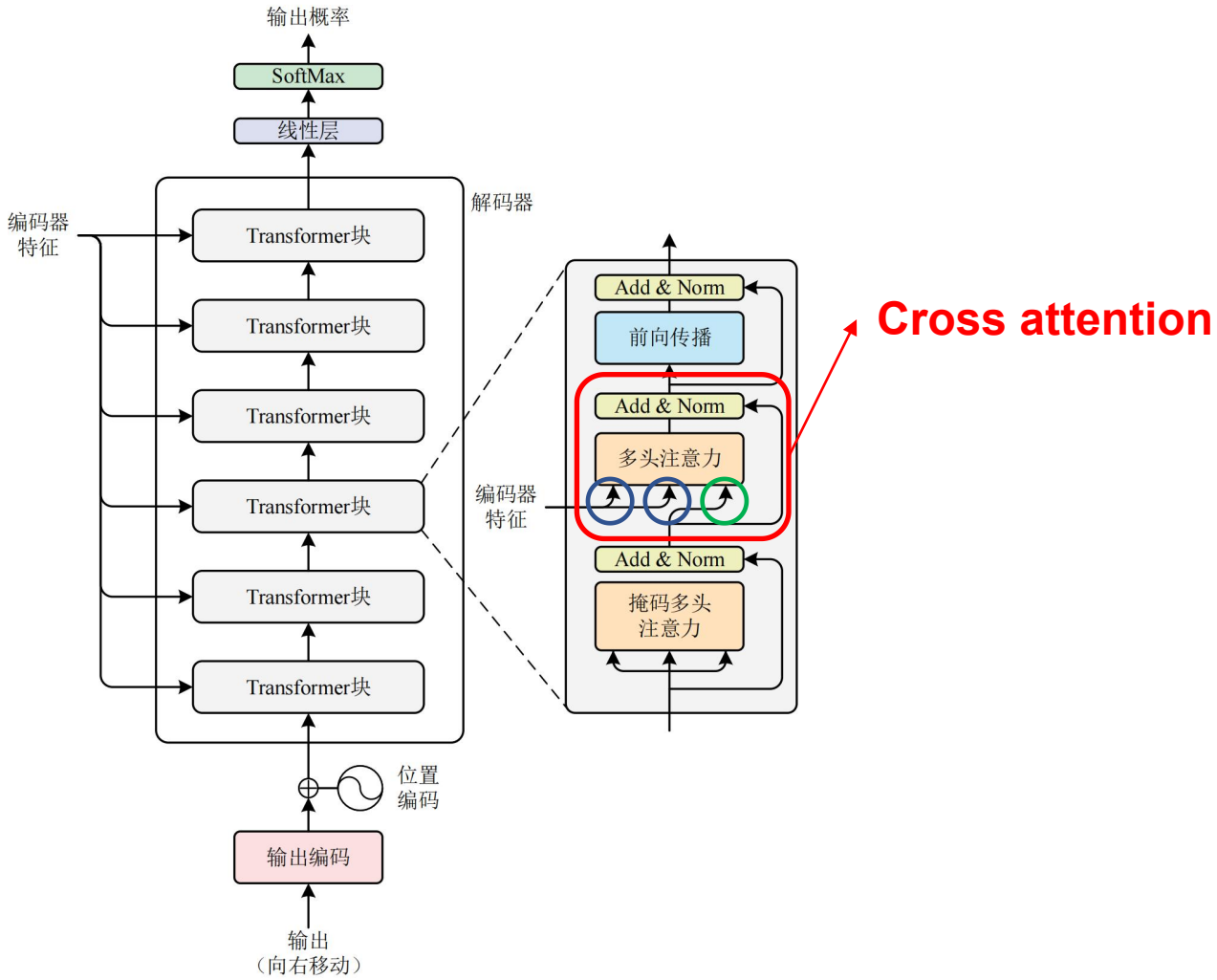
编码器-解码器结构



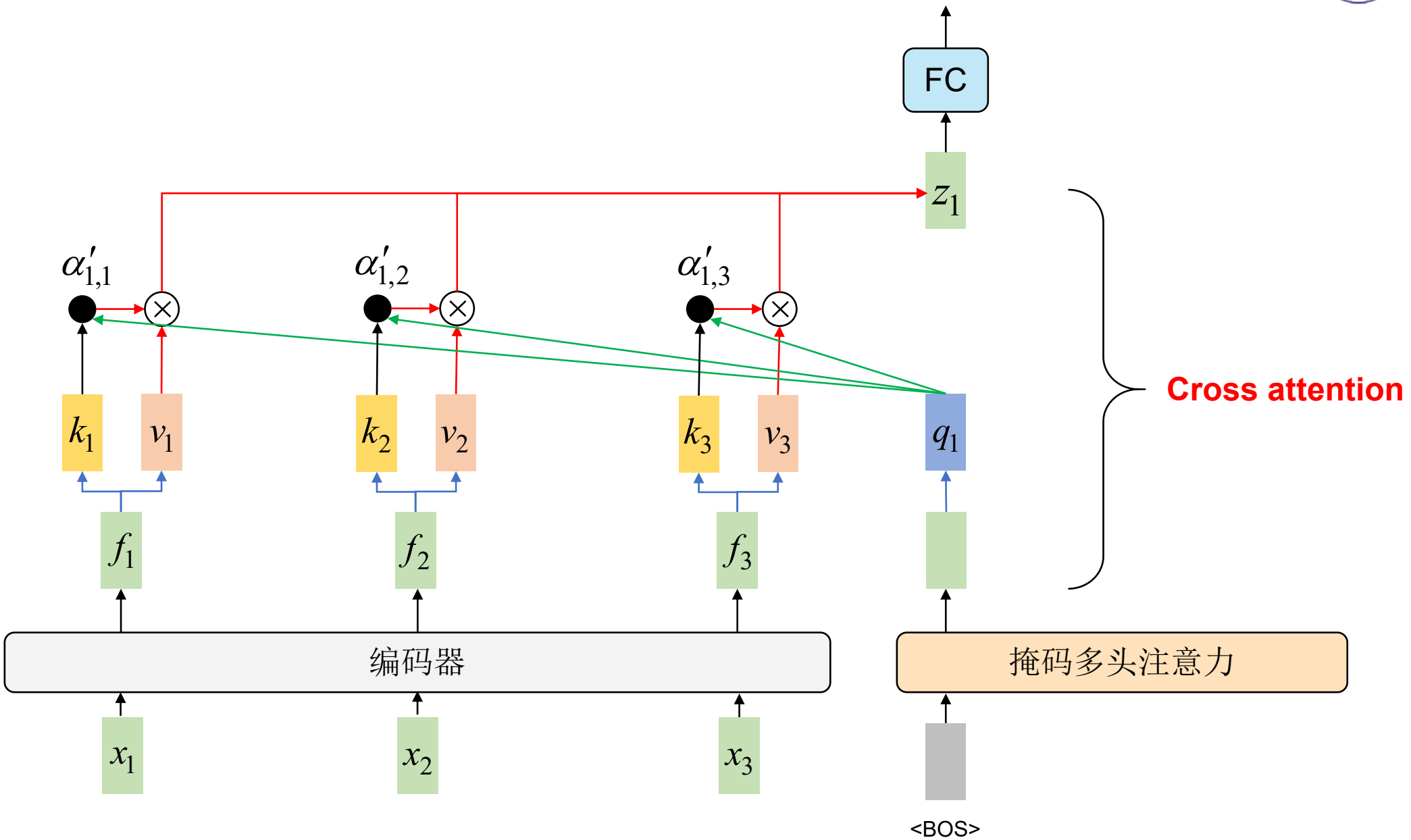
在训练的过程中，使用掩码多头注意力模块

编码器-解码器结构

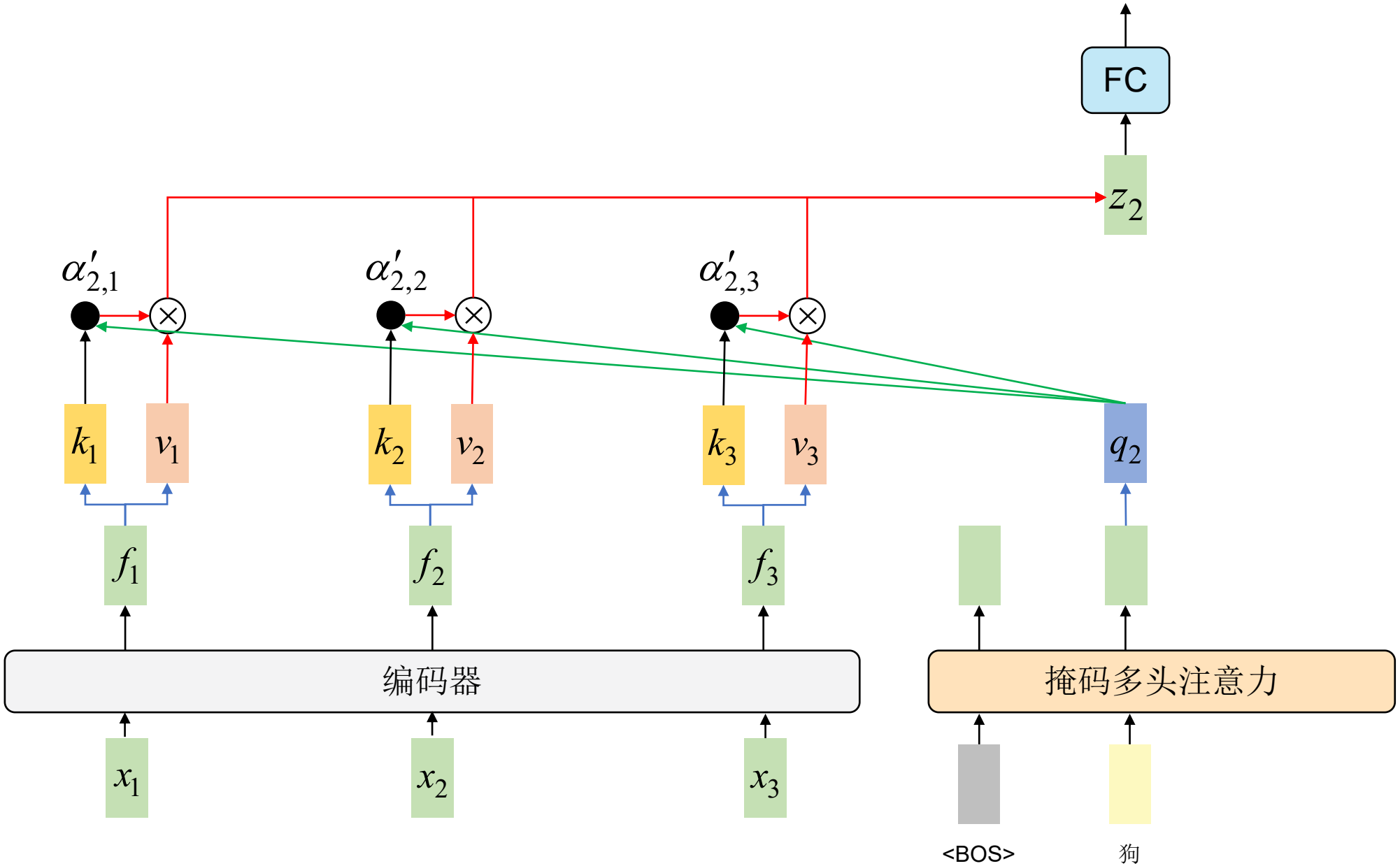
3.编码器与解码器之间的信息传递



编码器-解码器结构


FC
z_1
$\alpha'_{1,1}$
$\alpha'_{1,2}$
$\alpha'_{1,3}$
k_1
v_1
k_2
v_2
k_3
v_3
q_1
f_1
f_2
f_3
Cross attention
编码器
掩码多头注意力
x_1
x_2
x_3
<BOS>

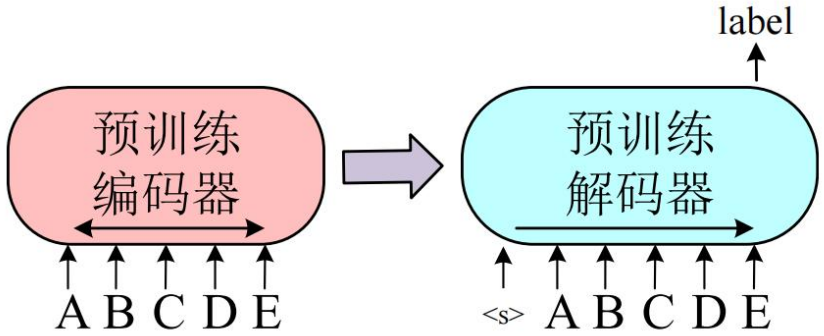
编码器-解码器结构

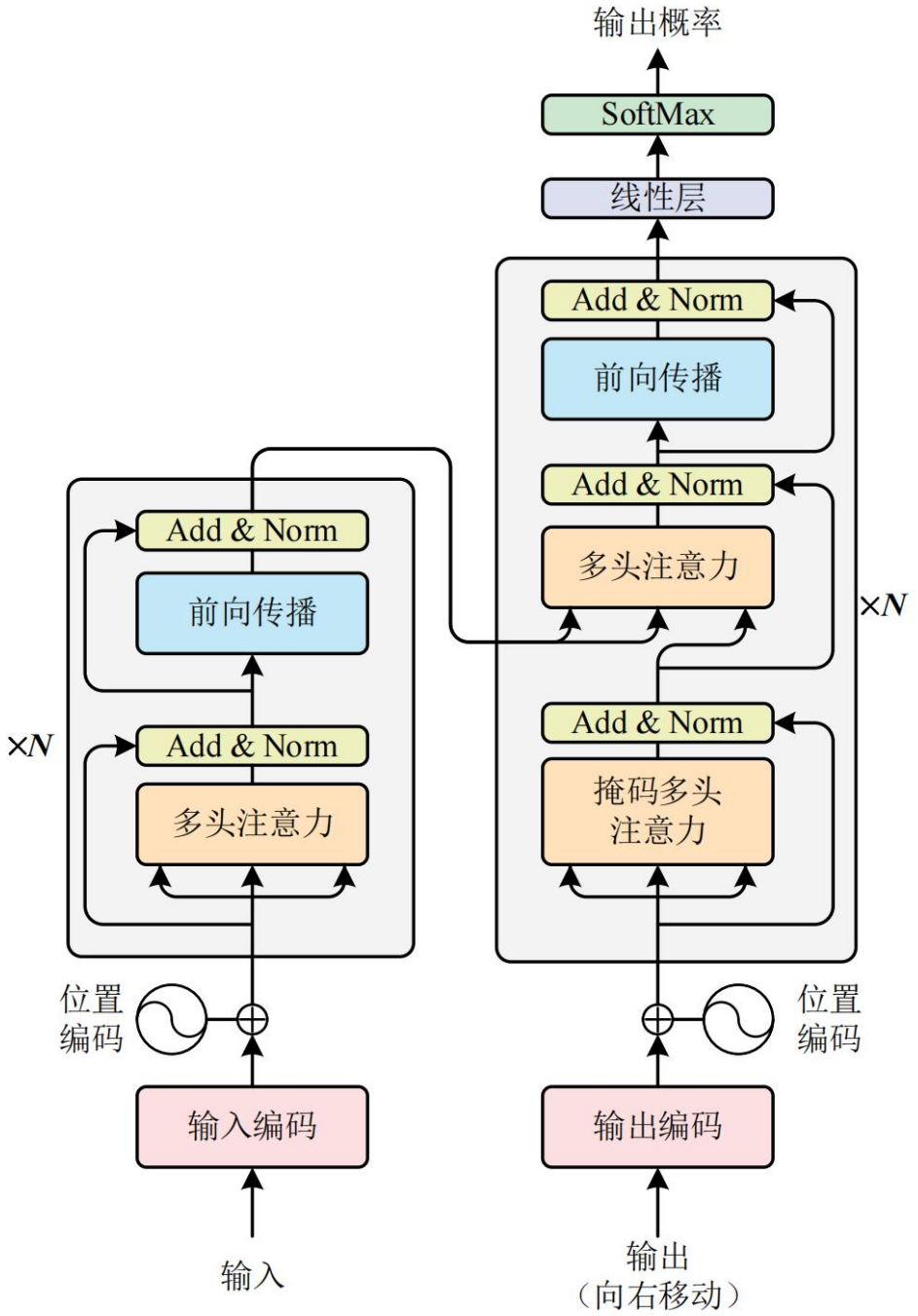

FC
z_2
$\alpha'_{2,1}$
$\alpha'_{2,2}$
$\alpha'_{2,3}$
k_1
v_1
k_2
v_2
k_3
v_3
q_2
f_1
f_2
f_3
编码器
掩码多头注意力
x_1
x_2
x_3
<BOS>
狗

目录

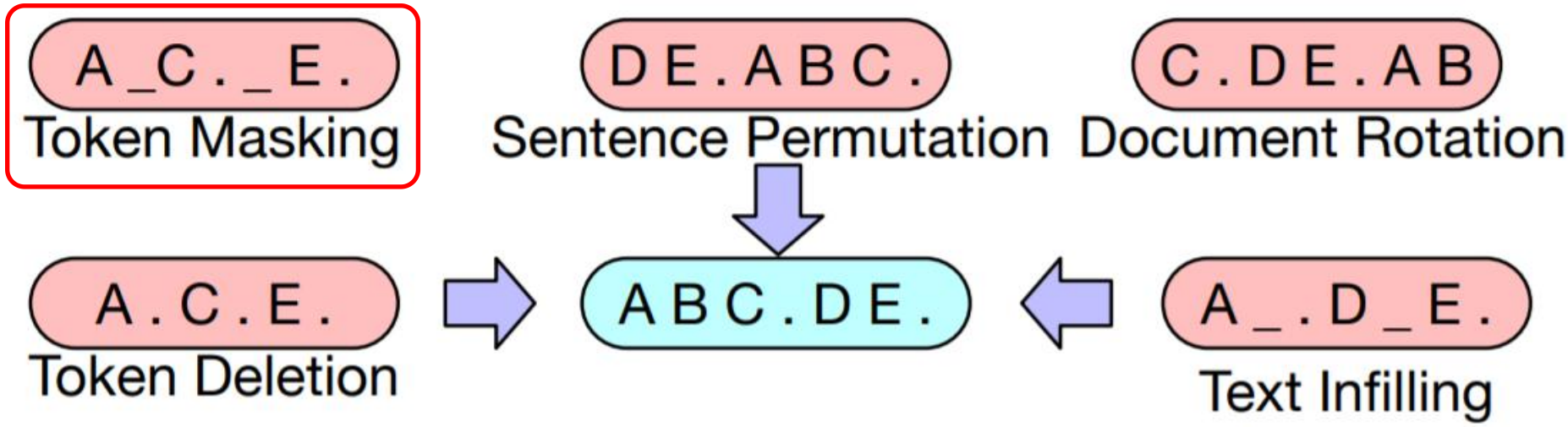
大模型中的编码器-解码器结构

1.BART

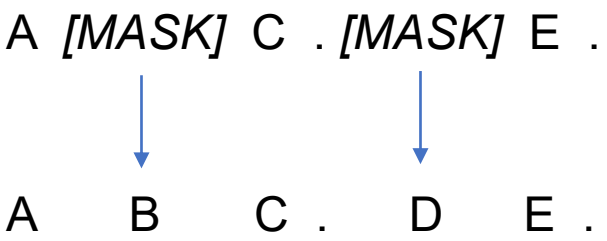




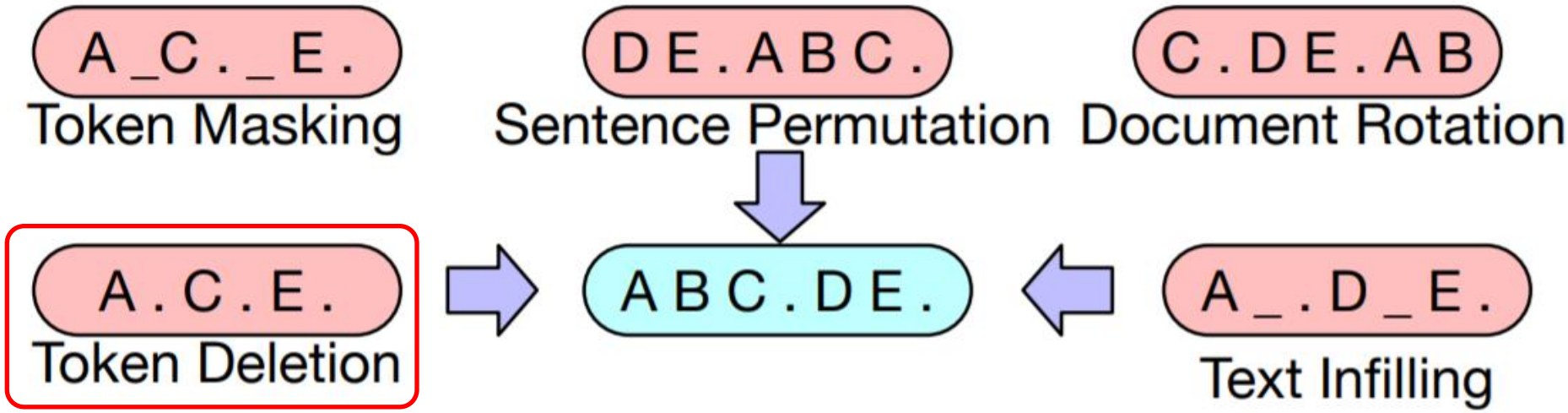
大模型中的编码器-解码器结构



Token Masking

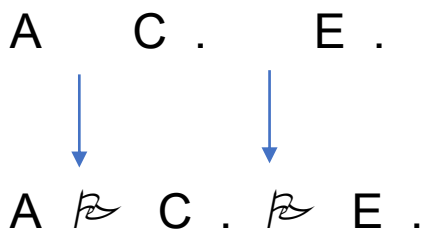


大模型中的编码器-解码器结构

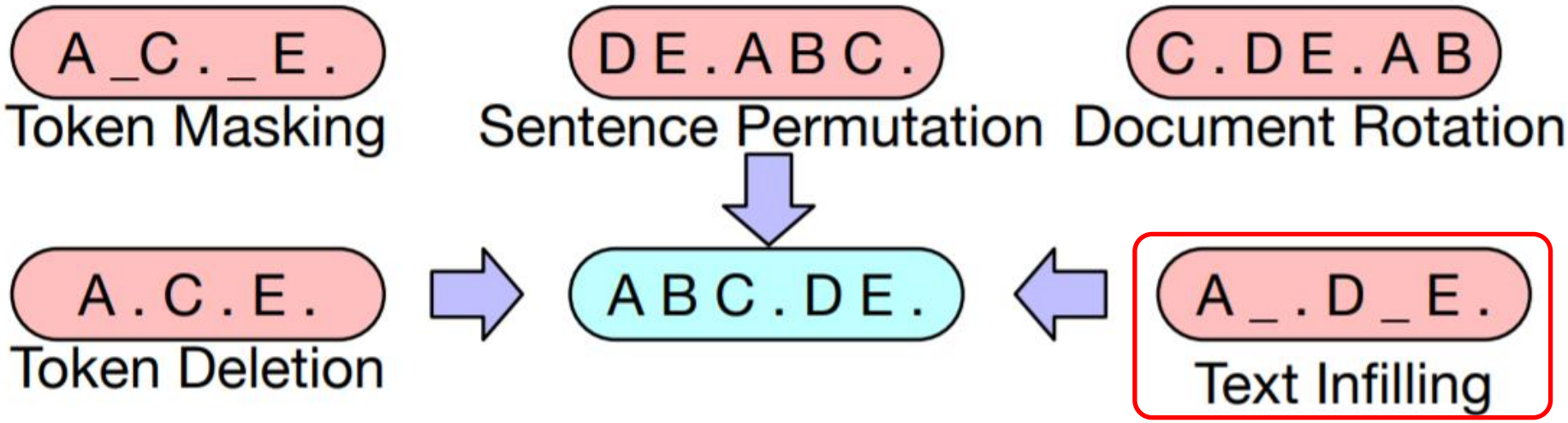


Token Masking

Token Deletion



大模型中的编码器-解码器结构



Token Masking

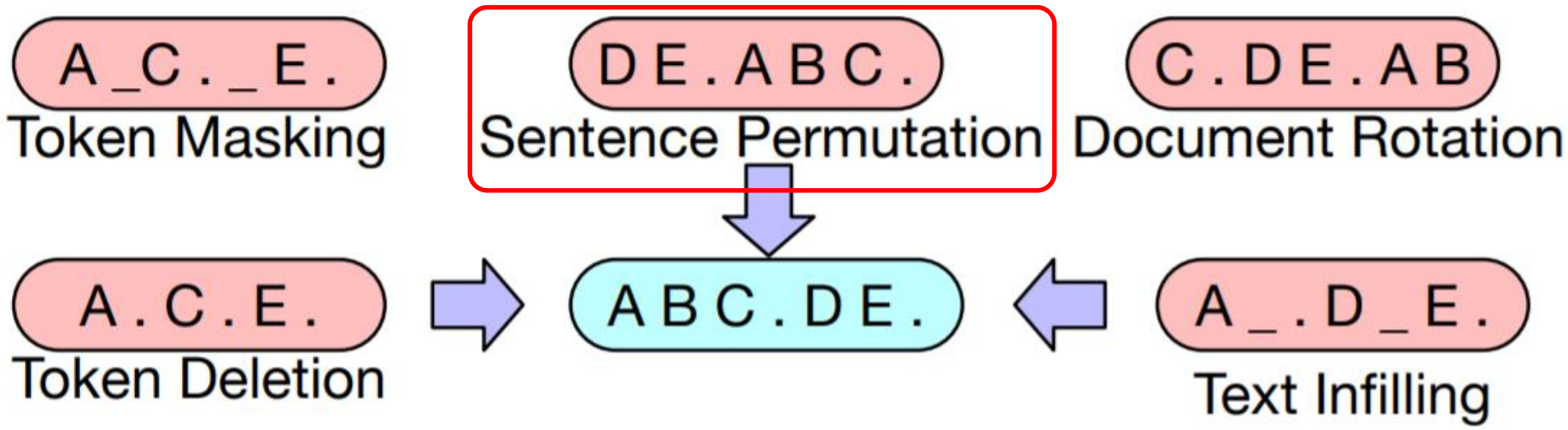
Token Deletion

Token Infilling

A *[MASK]* . D *[MASK]* E .

A 2 . D 0 E .

大模型中的编码器-解码器结构

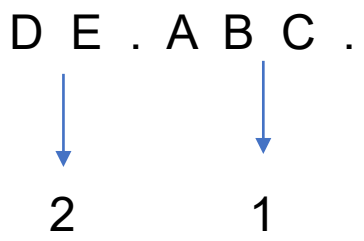


Token Masking

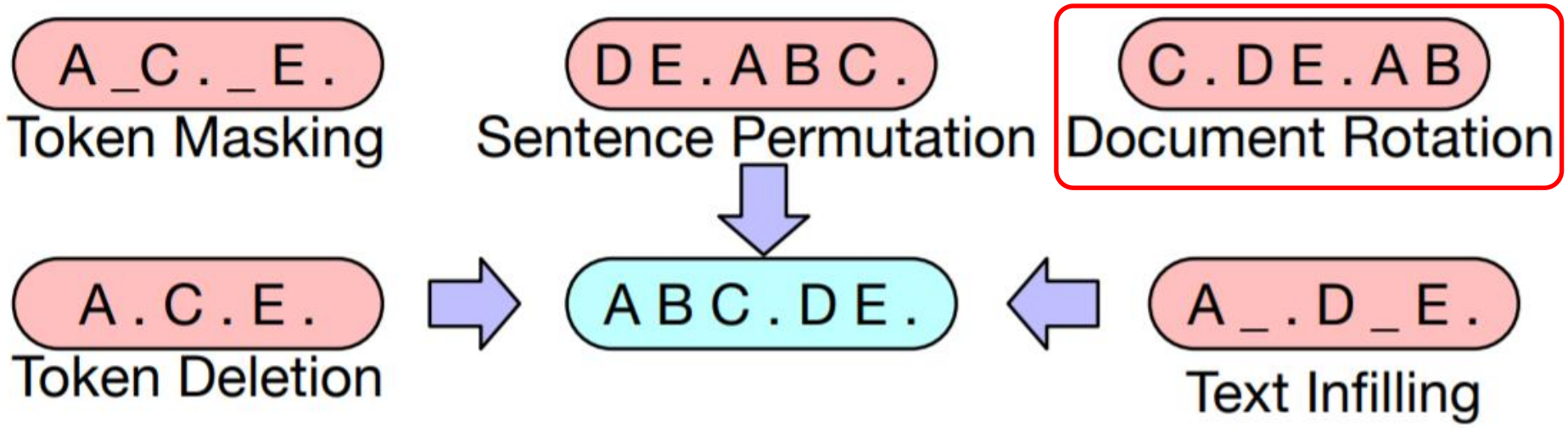
Token Deletion

Token Infilling

Sentence Permutation



大模型中的编码器-解码器结构



Token Masking

Token Deletion

Token Infilling

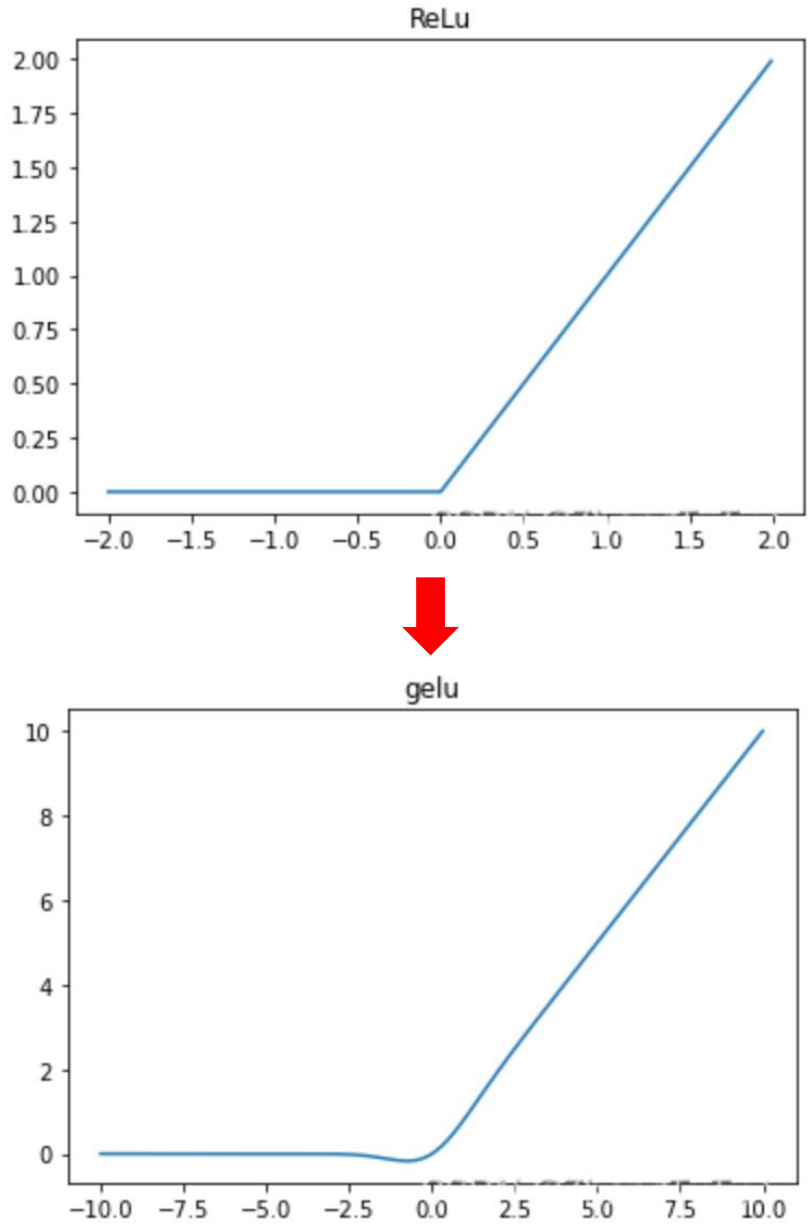
Sentence Permutation

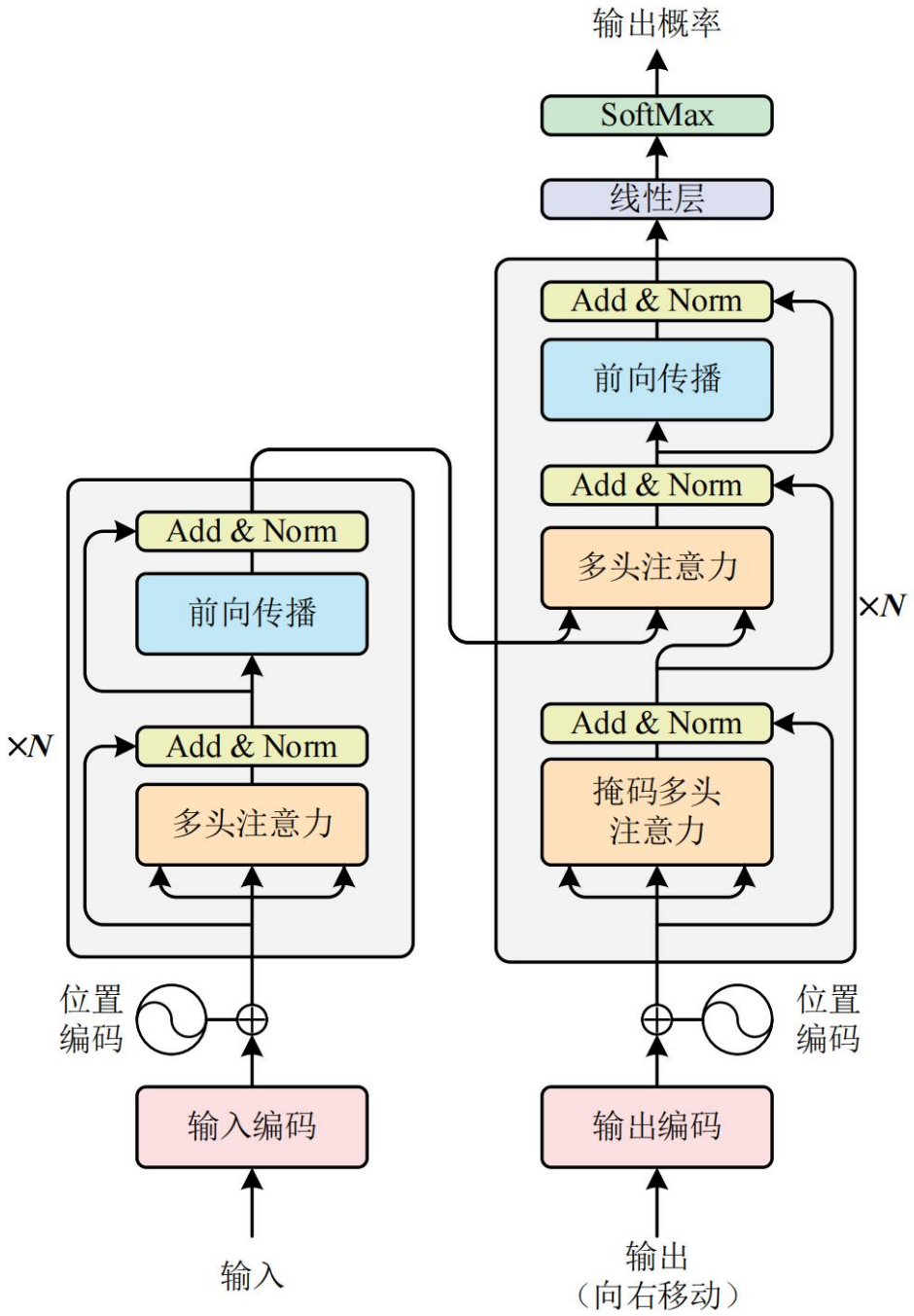
Document Rotation

C . D E . A B

Begin

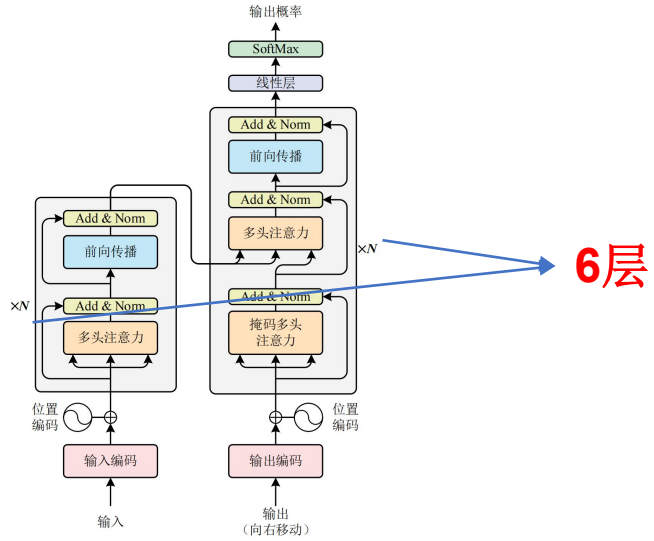
大模型中的编码器-解码器结构





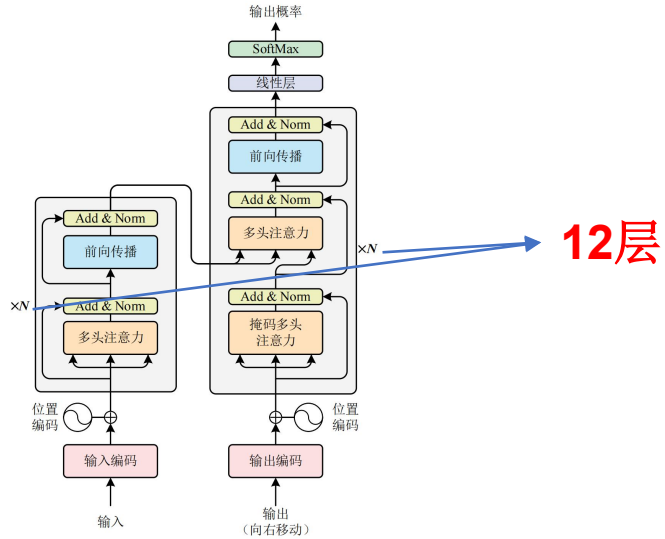
大模型中的编码器-解码器结构

Base

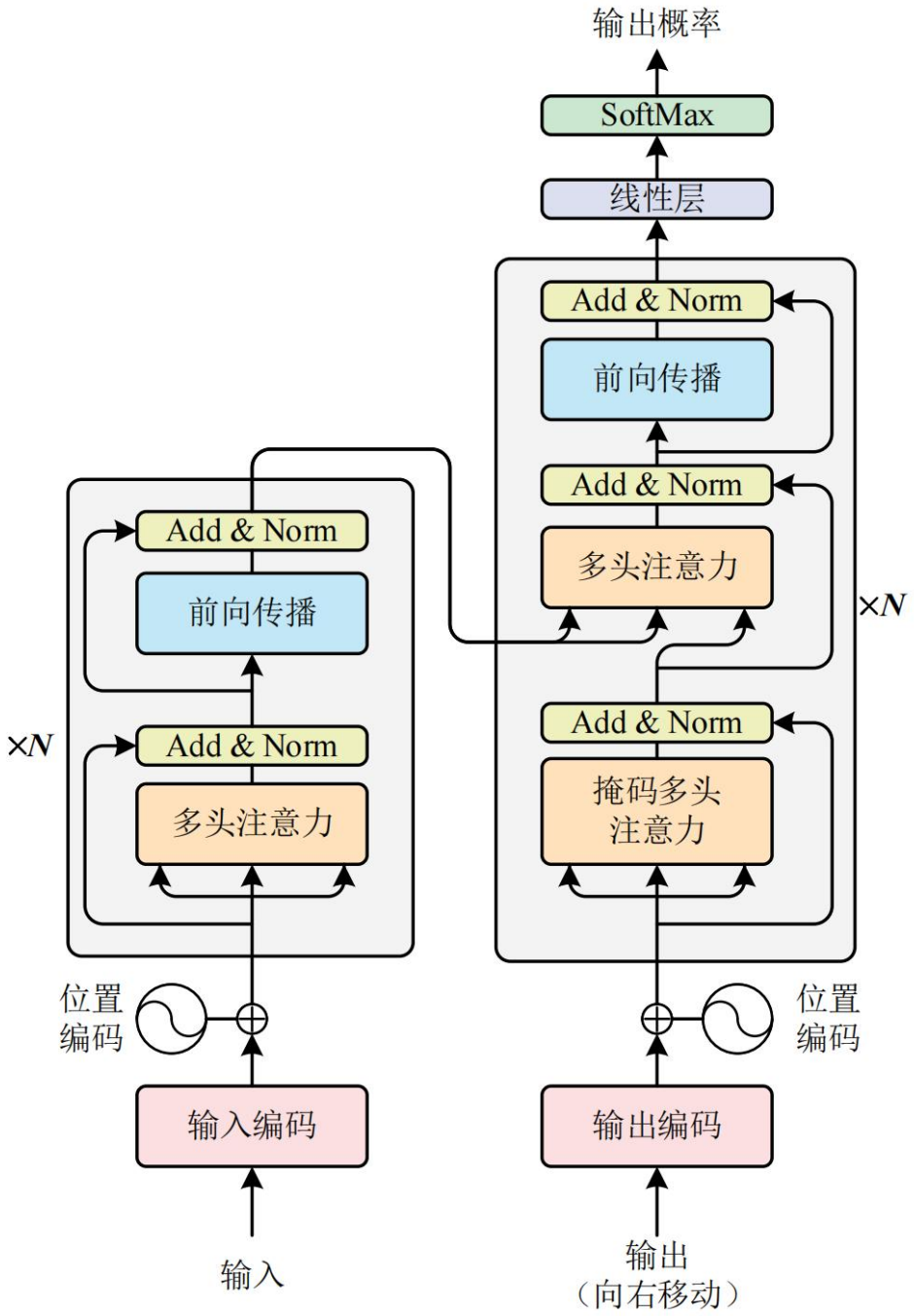


6层

Large

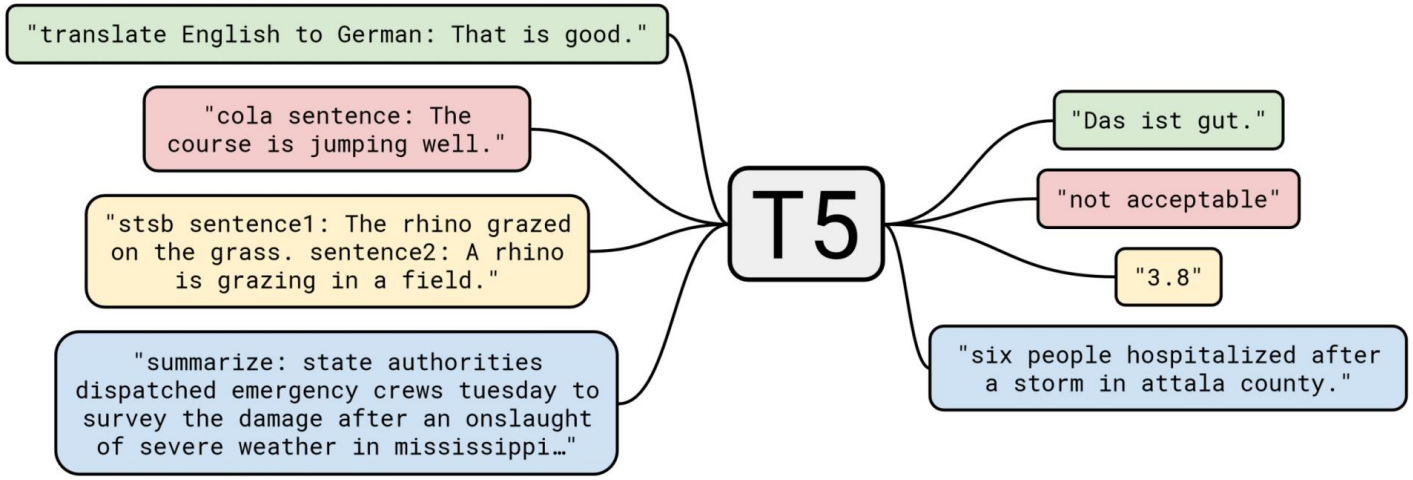


12层

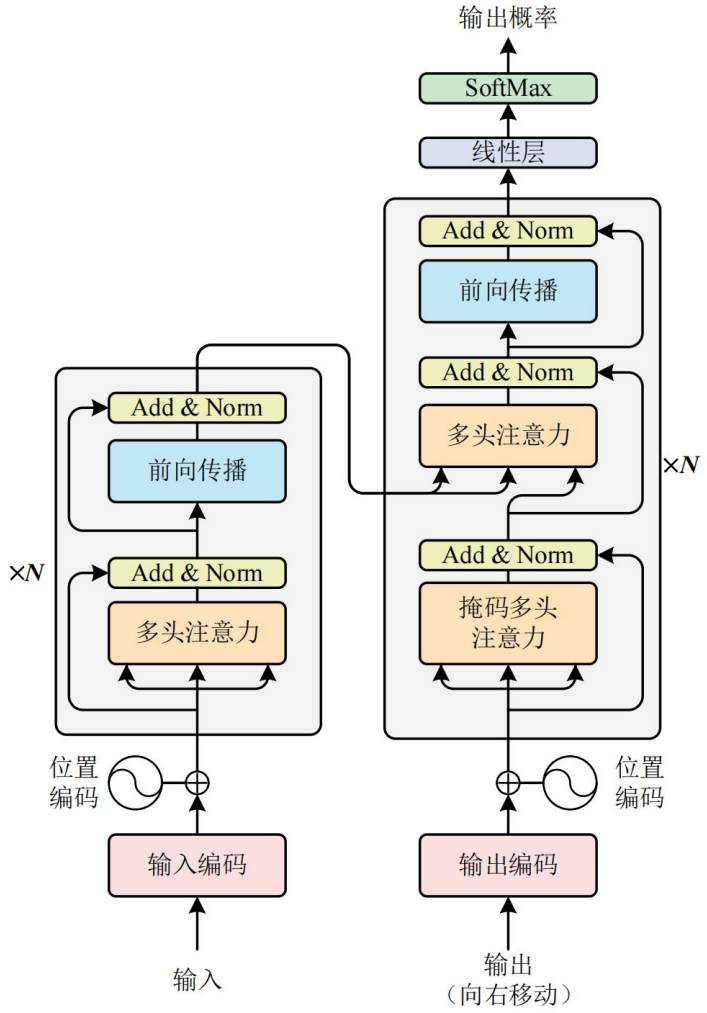


大模型中的编码器-解码器结构

1.T5

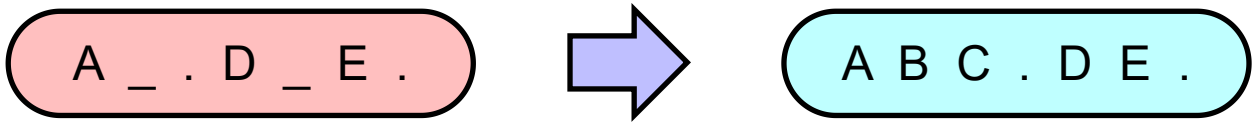


将所有文本处理问题转化为“文本到文本”的问题



大模型中的编码器-解码器结构

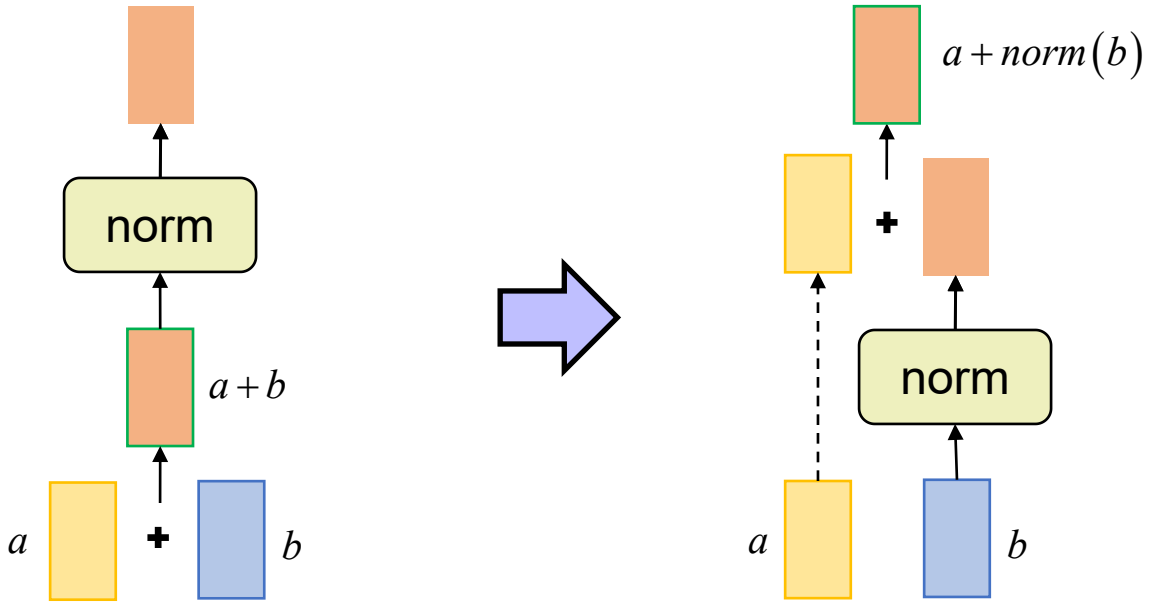
Replace Span



层归一化

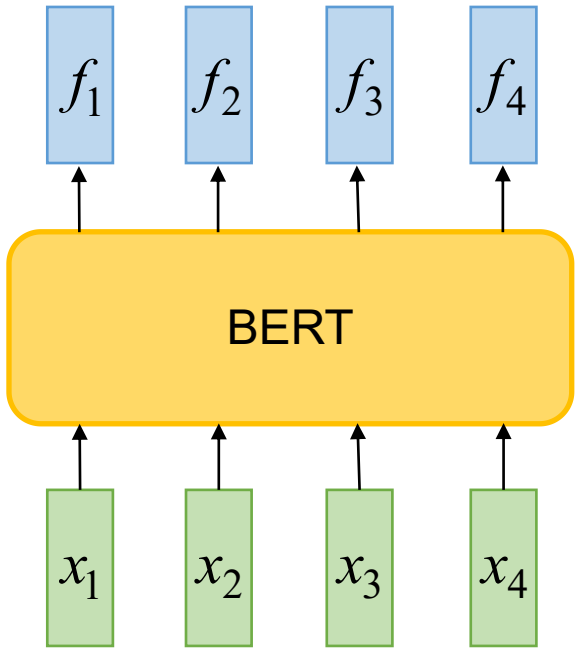
$$LayerNorm(x)=\frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}}*\gamma+\beta \quad \Rightarrow \quad LayerNorm(x)=\frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}}*\gamma$$

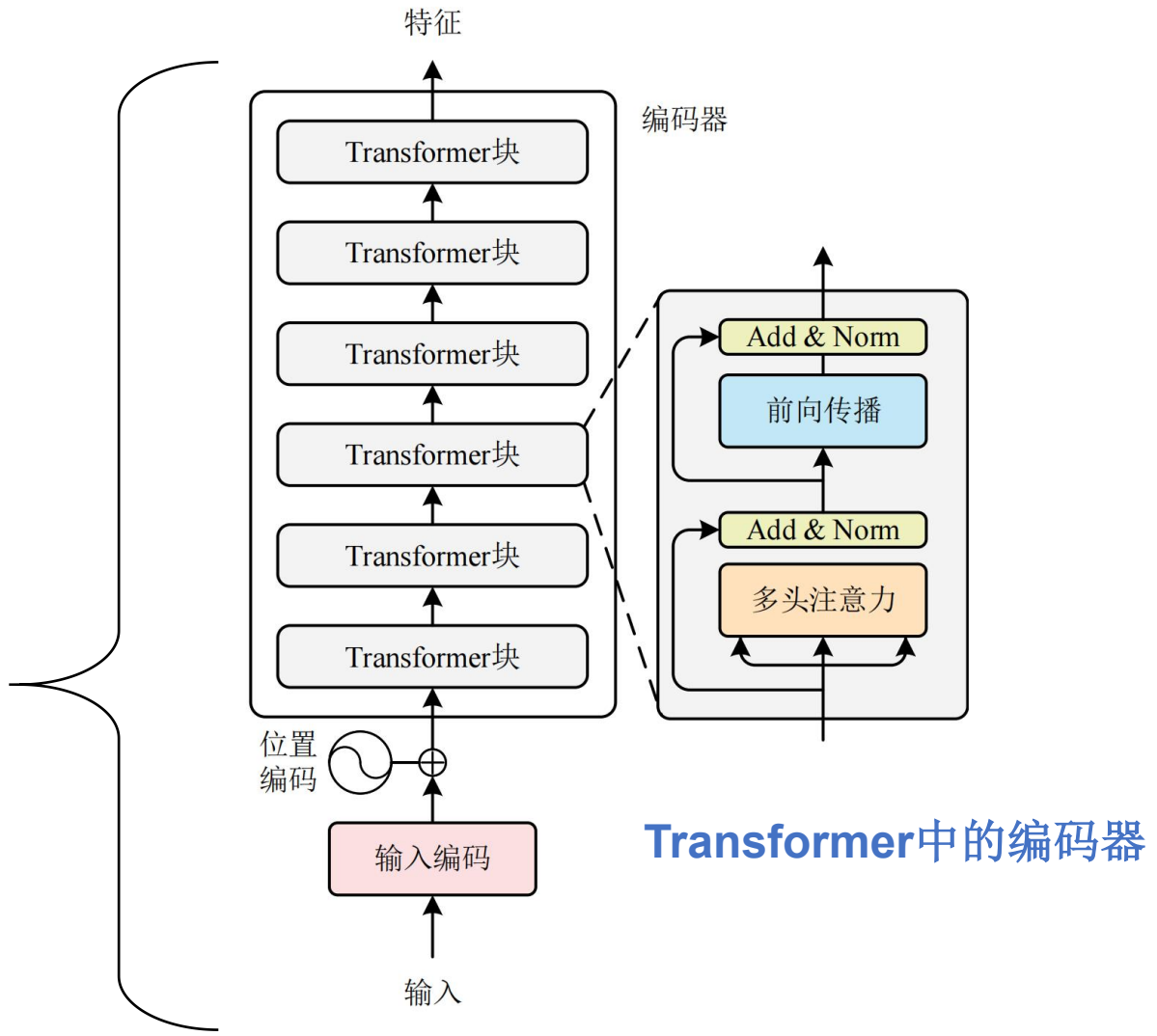
残差连接



目录

编码器结构-BERT家族





Transformer中的编码器

编码器结构-BERT家族

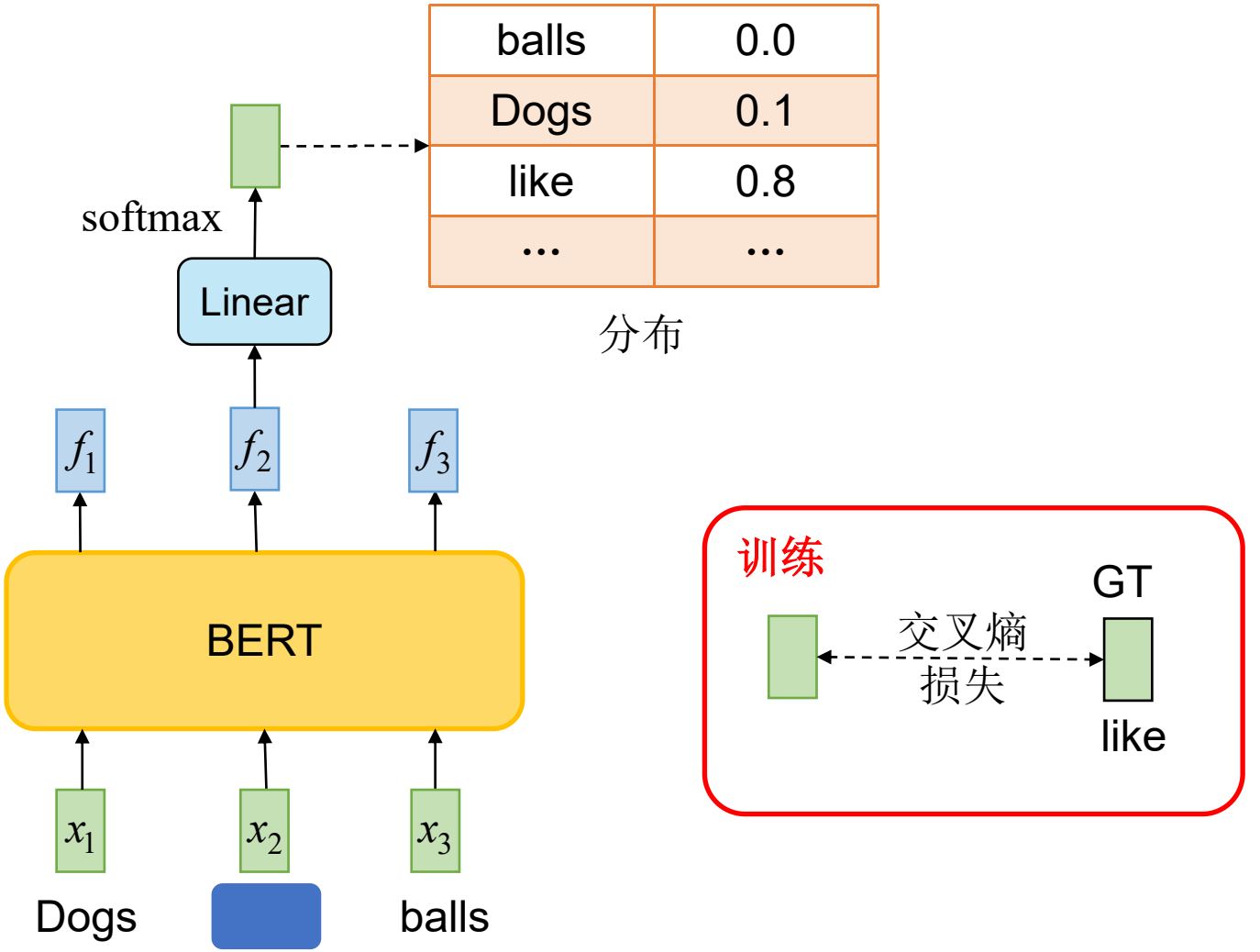
2.预训练策略

自监督学习策略的提出使得BERT的训练成为可能

“掩码语言建模”

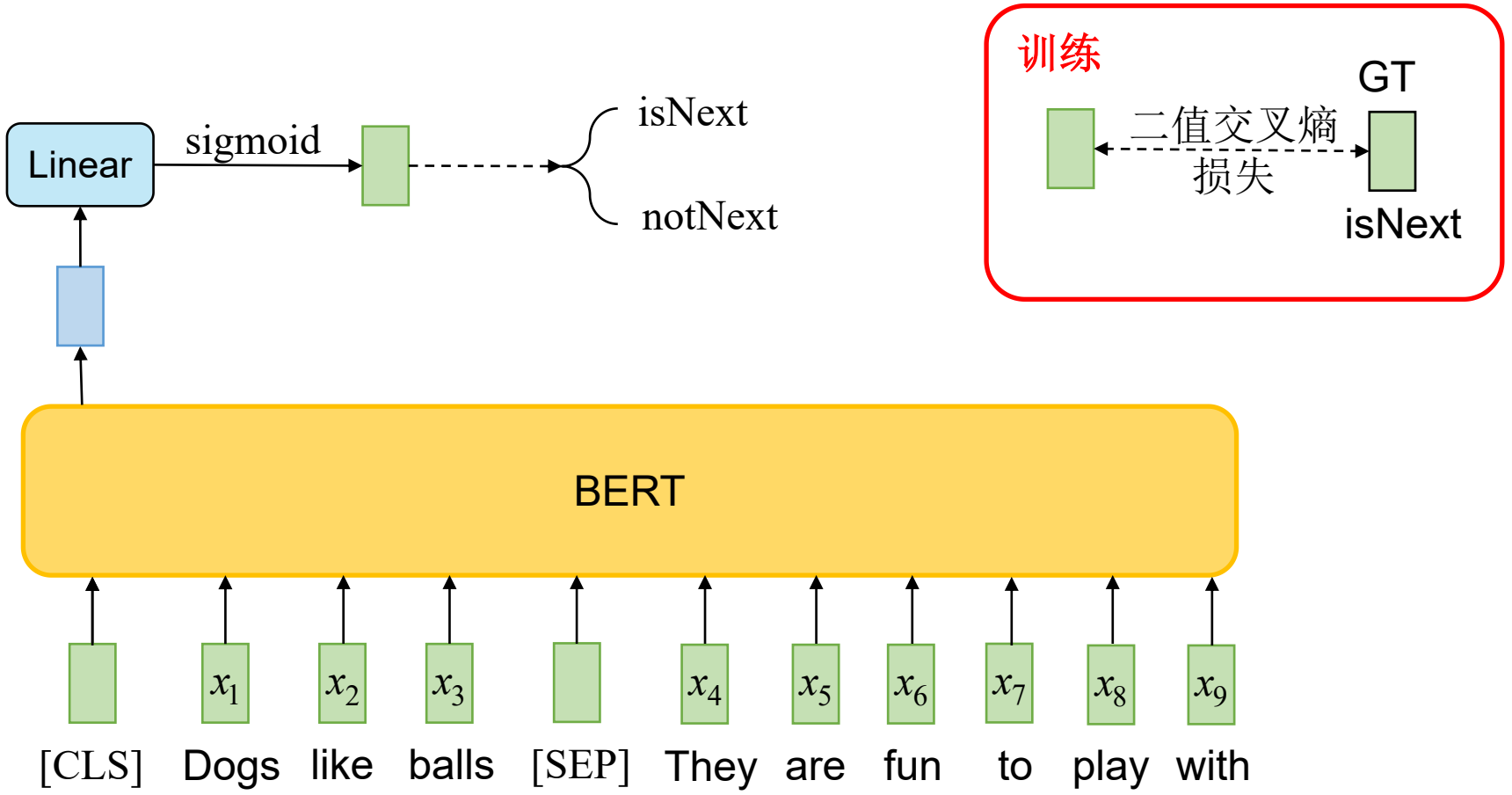
= “*[MASK]*”

随机掩盖其中一些token



编码器结构-BERT家族

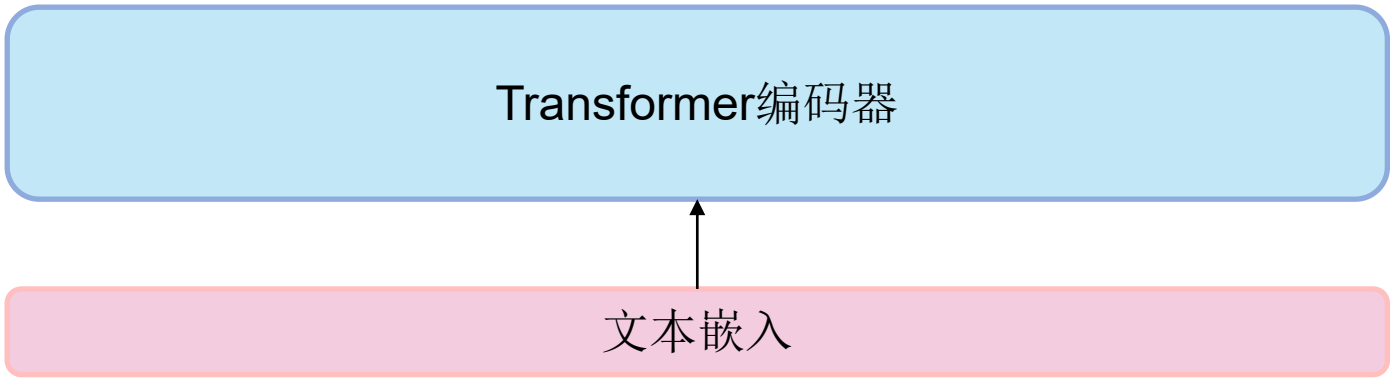
“下句预测”

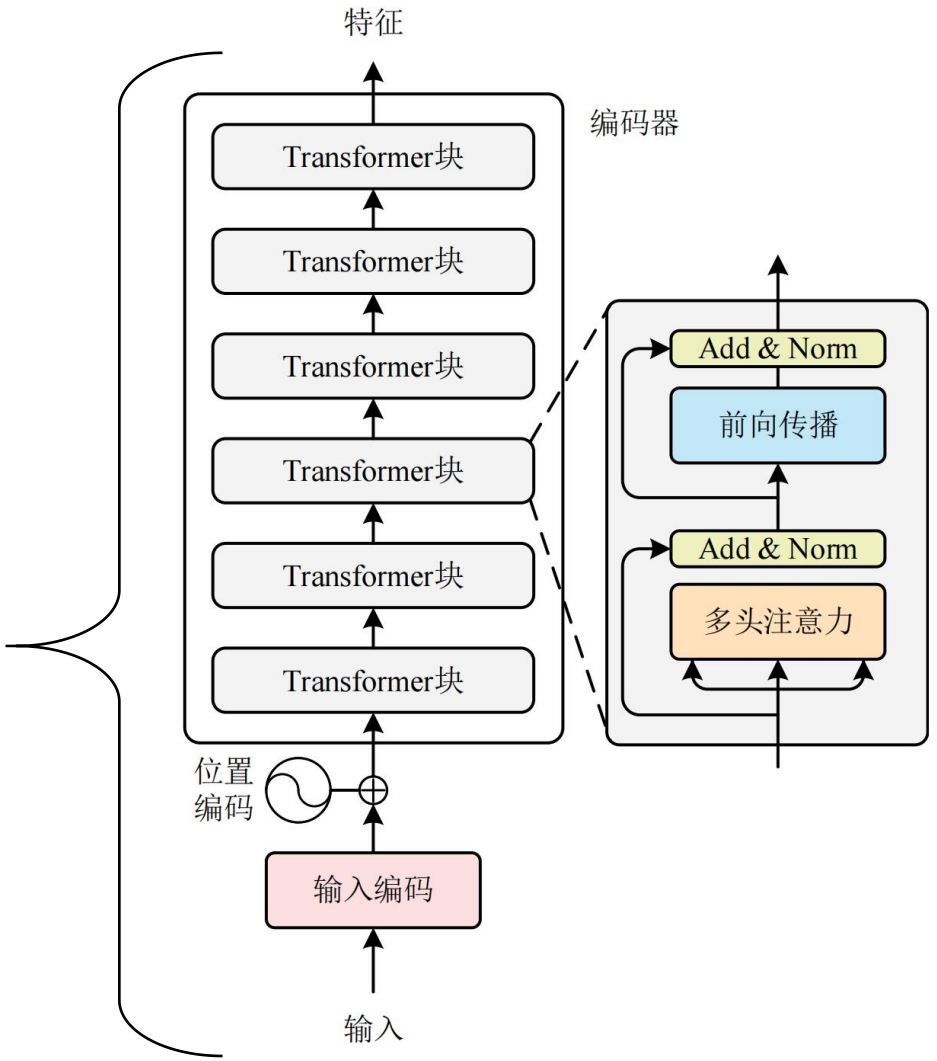


目录

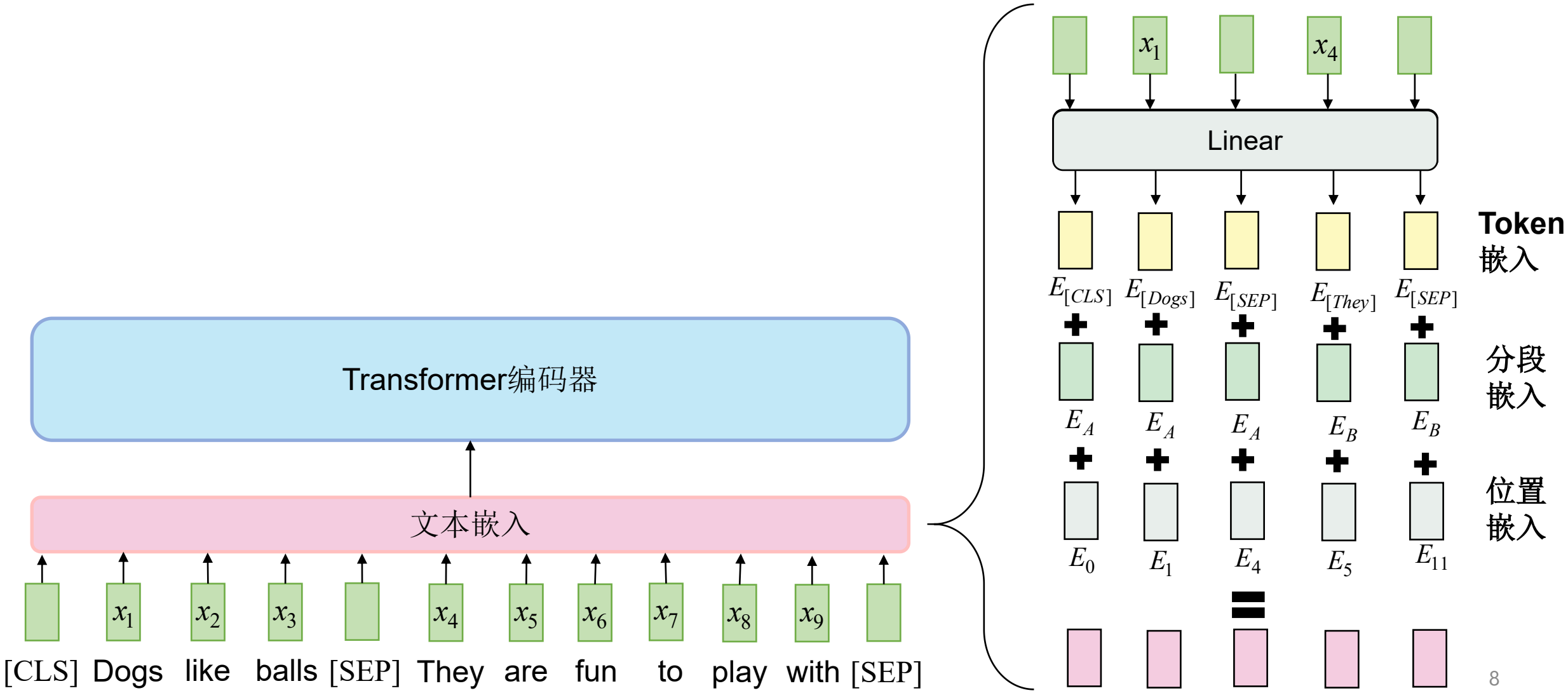
编码器结构-BERT家族

1.BERT结构



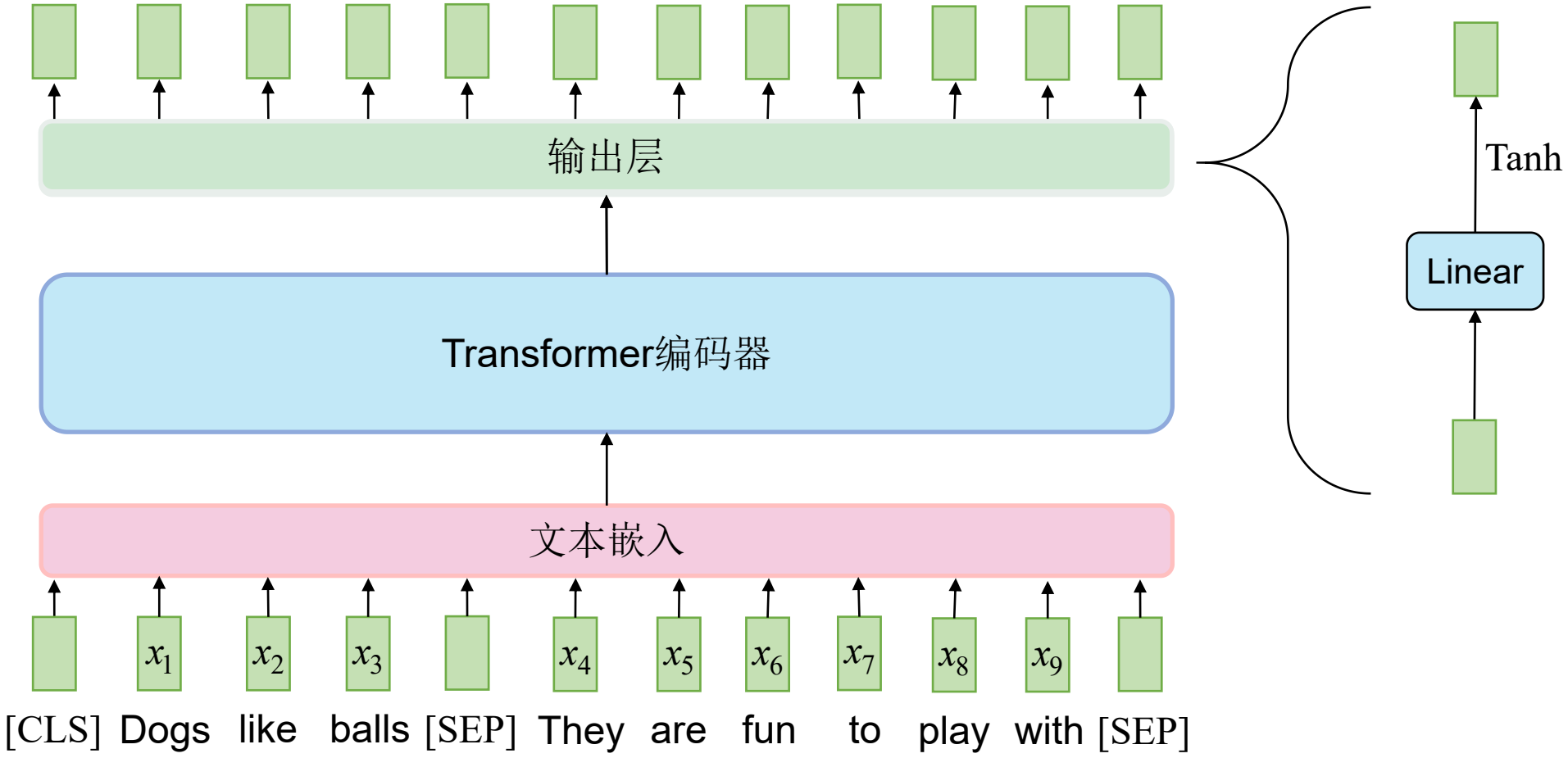


编码器结构-BERT家族



编码器结构-BERT家族

1.BERT结构



目录

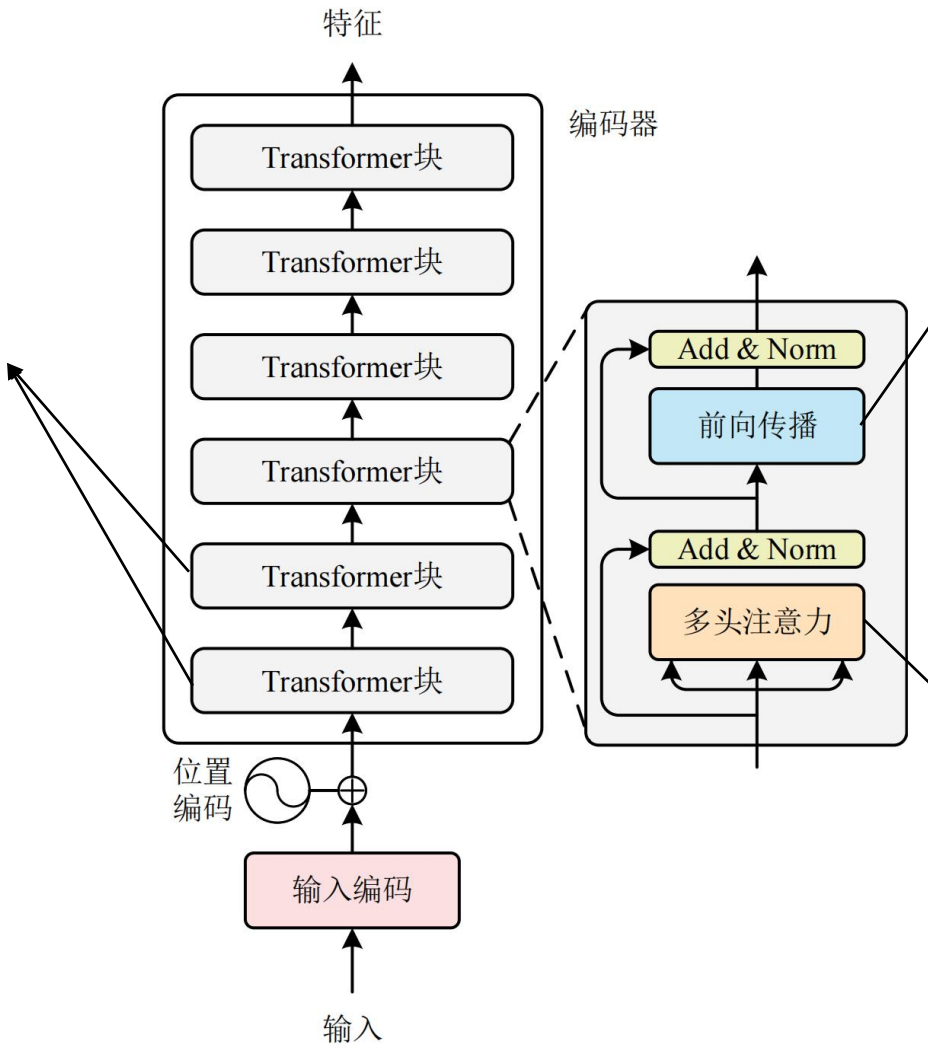
编码器结构-BERT家族

3.BERT的变体

BERT的参数量太大--ALBERT

“跨层参数共享”

全共享：
Transformer
块的参数共享



共享前馈网络层：
每个Transformer
块的前向传播层参
数共享

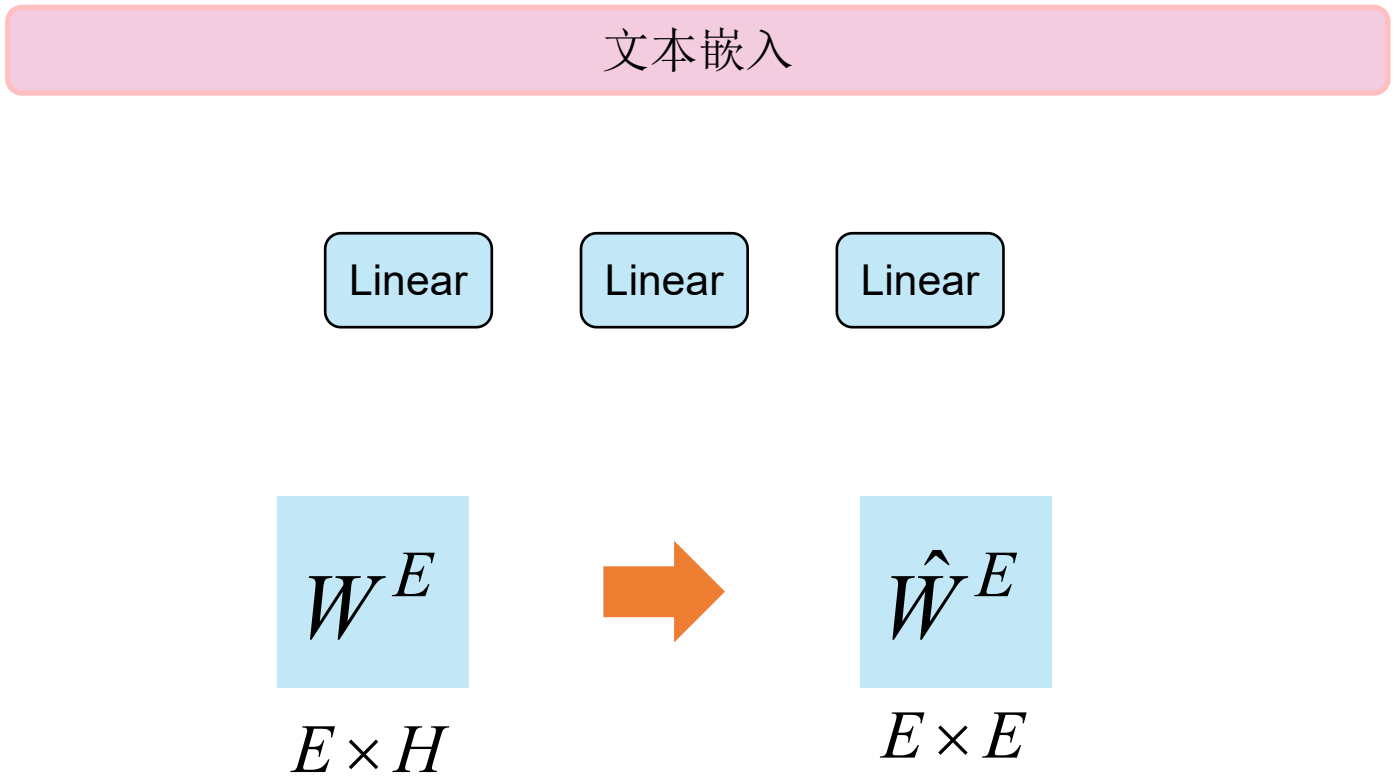
共享注意力层：
每个Transformer
块的多头注意力
层参数共享

编码器结构-BERT家族

3.BERT的变体

BERT的参数量太大--ALBERT

“嵌入层参数因子分解”



编码器结构-BERT家族

模型	参数量	层数 N	隐藏神经元数量	嵌入层
BERT-base	1.1 亿	12	768	768
BERT-large	3.4 亿	24	1024	1024
ALBERT-base	1200 万	12	768	128
ALBERT-large	1800 万	24	1024	128
ALBERT-xlarge	6000 万	24	2048	128
ALBERT-xxlarge	2.35 亿	12	4096	128

编码器结构-BERT家族

RoBERTa：改进BERT预训练

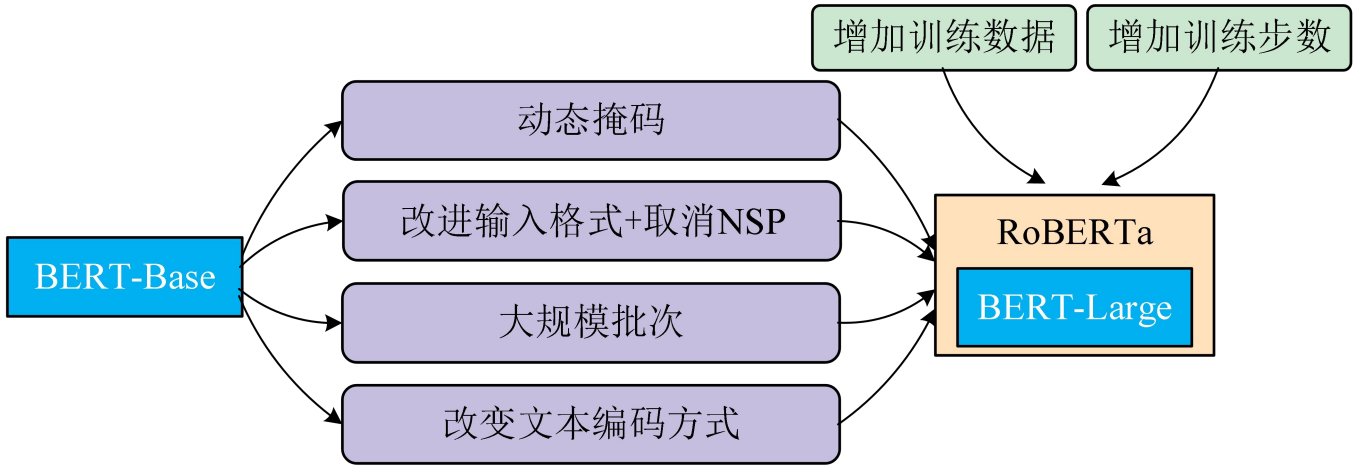
动态掩码

移除下句预测任务

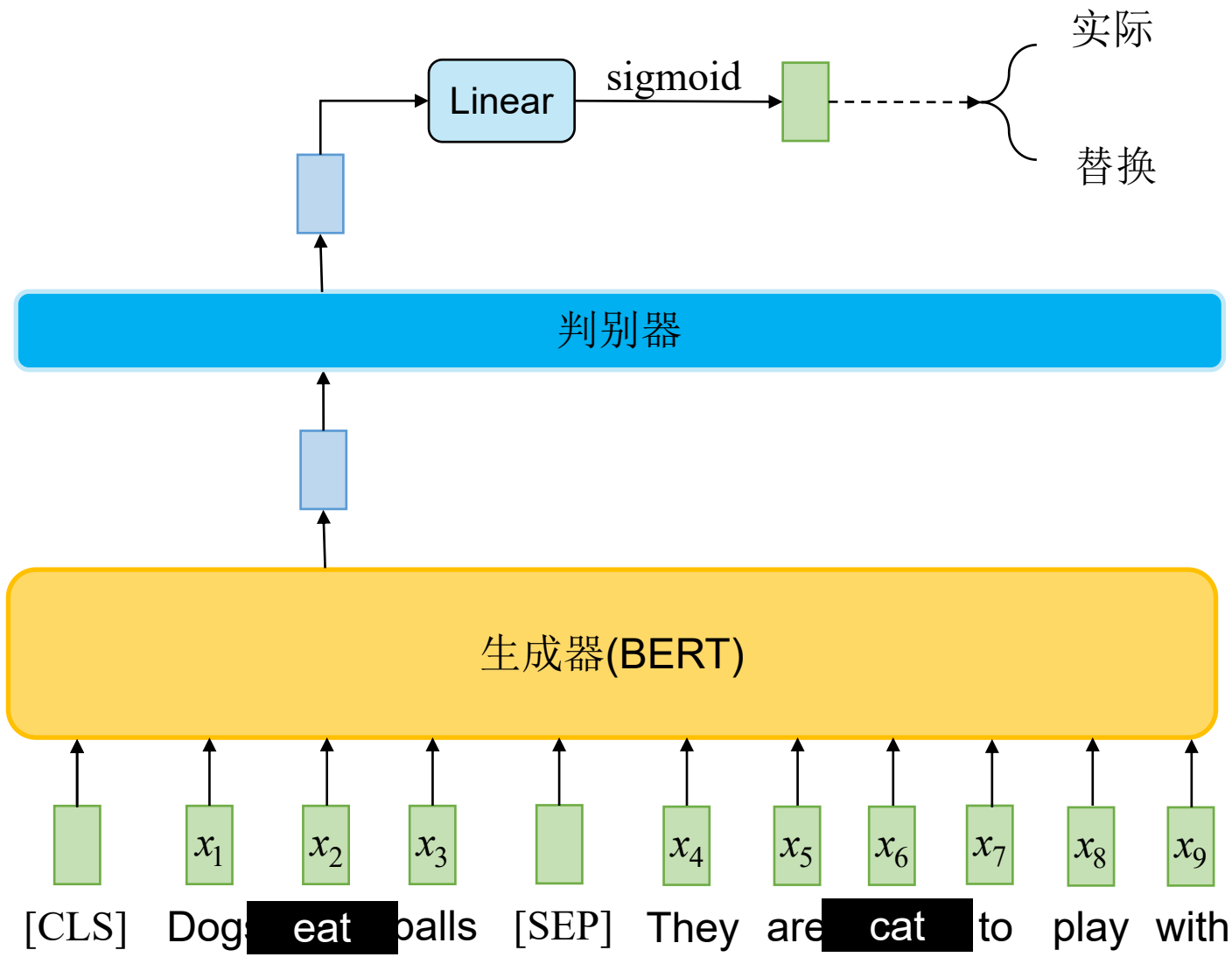
增加数据和训练步数

增大batch size

使用字节级字节对编码



编码器结构-BERT家族



= 从词汇表中随机选择一个token

目录

解码器结构-GPT家族

2018年6月 OpenAI发布GPT模型(0.1B)

2019年2月 OpenAI发布GPT-2模型(1.5B)

2020年5月 OpenAI发布GPT-3模型(175B)

2022年3月 OpenAI发布InstructionGPT,使用Instruction-tuning和RLHF

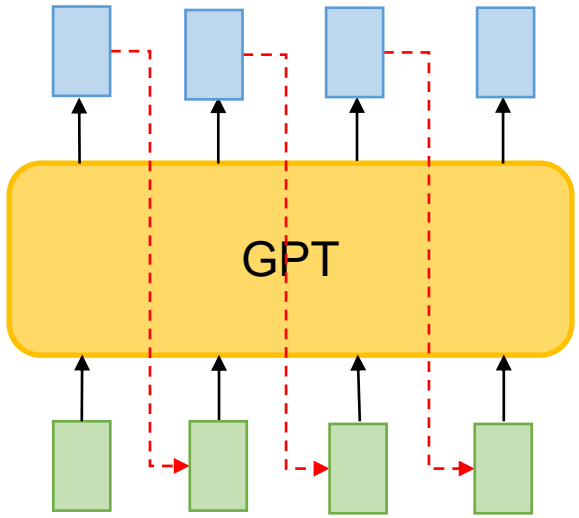
2022年12月 OpenAI发布ChatGPT

2023年3月 OpenAI发布ChatGPT-4

第一阶段:
提出新颖的自监督训练目标
设计模型架构
针对某部分领域数据单独Fine-tuning

第二阶段:
逐步扩大模型参数和训练语料规模
模型架构偏向生成式任务
Prompt技术在Fine-tuning阶段得以展现

第三阶段:
进一步增大模型参数量和训练语料规模
完全采用单向因果模型
注重人类反馈信息、对话交互
关注于模型可靠性、事实性、有偏性、安全性
更多丰富的Prompt技术用于Fine-tuning或Inference

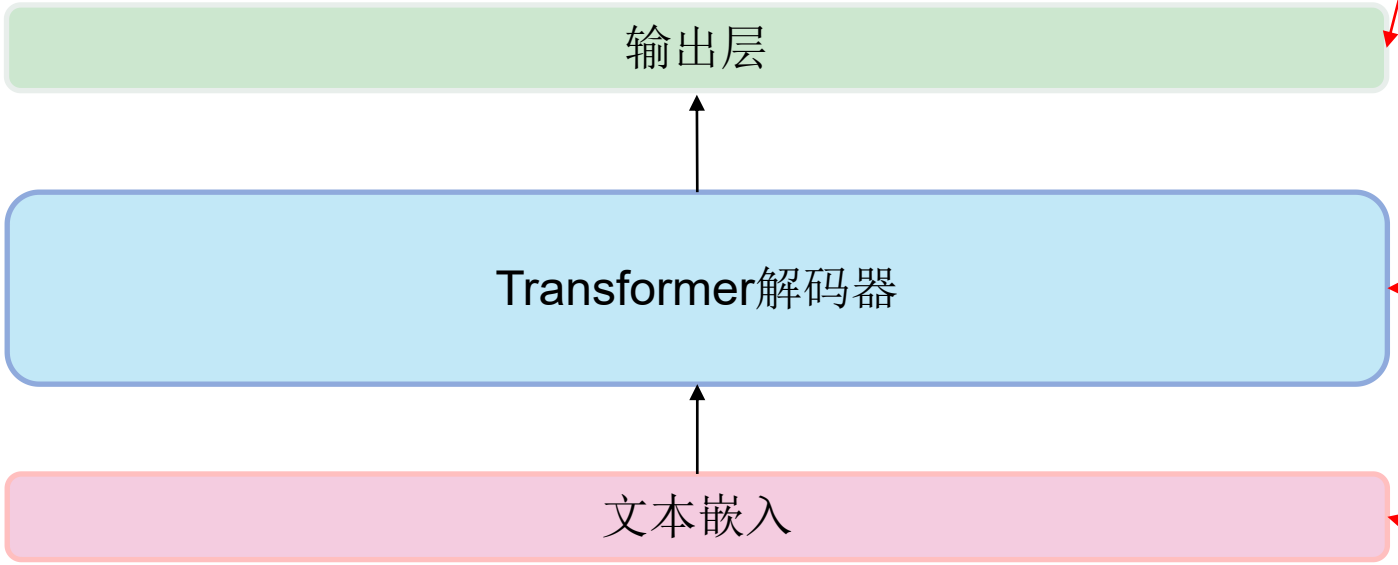


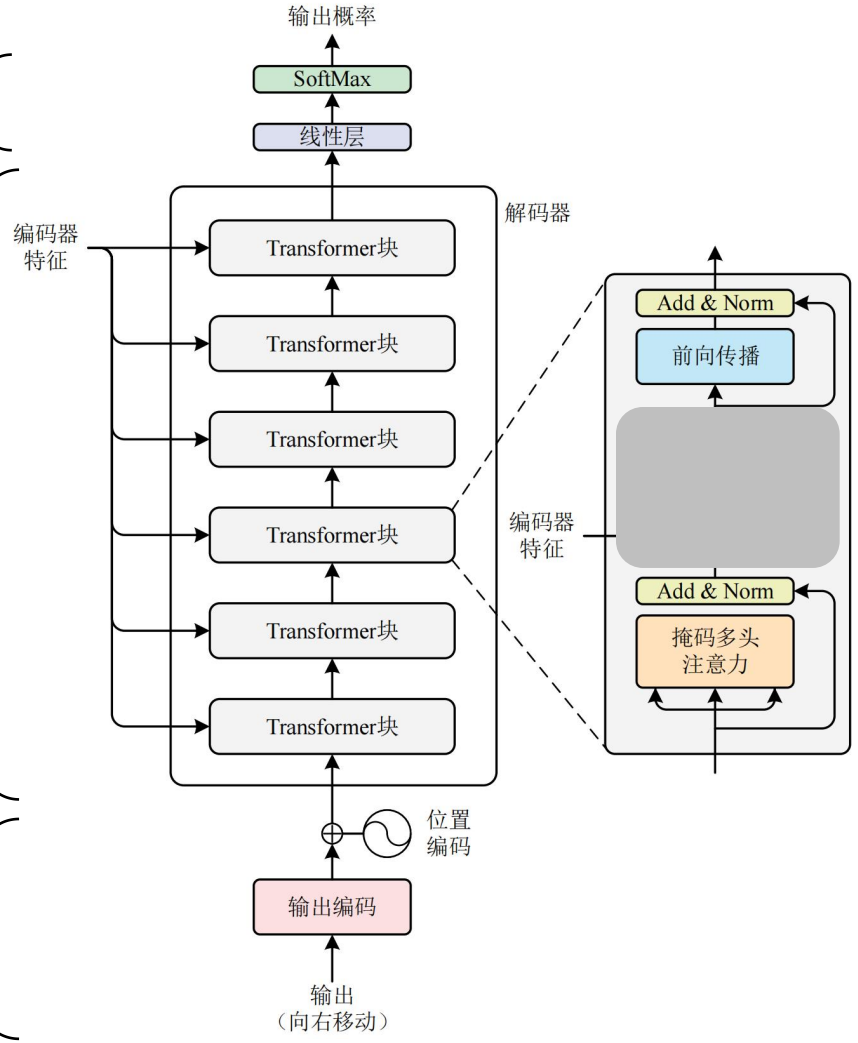
Transformer解码器样式

OpenAI

解码器结构-GPT家族

1.GPT结构



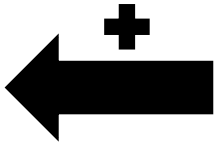


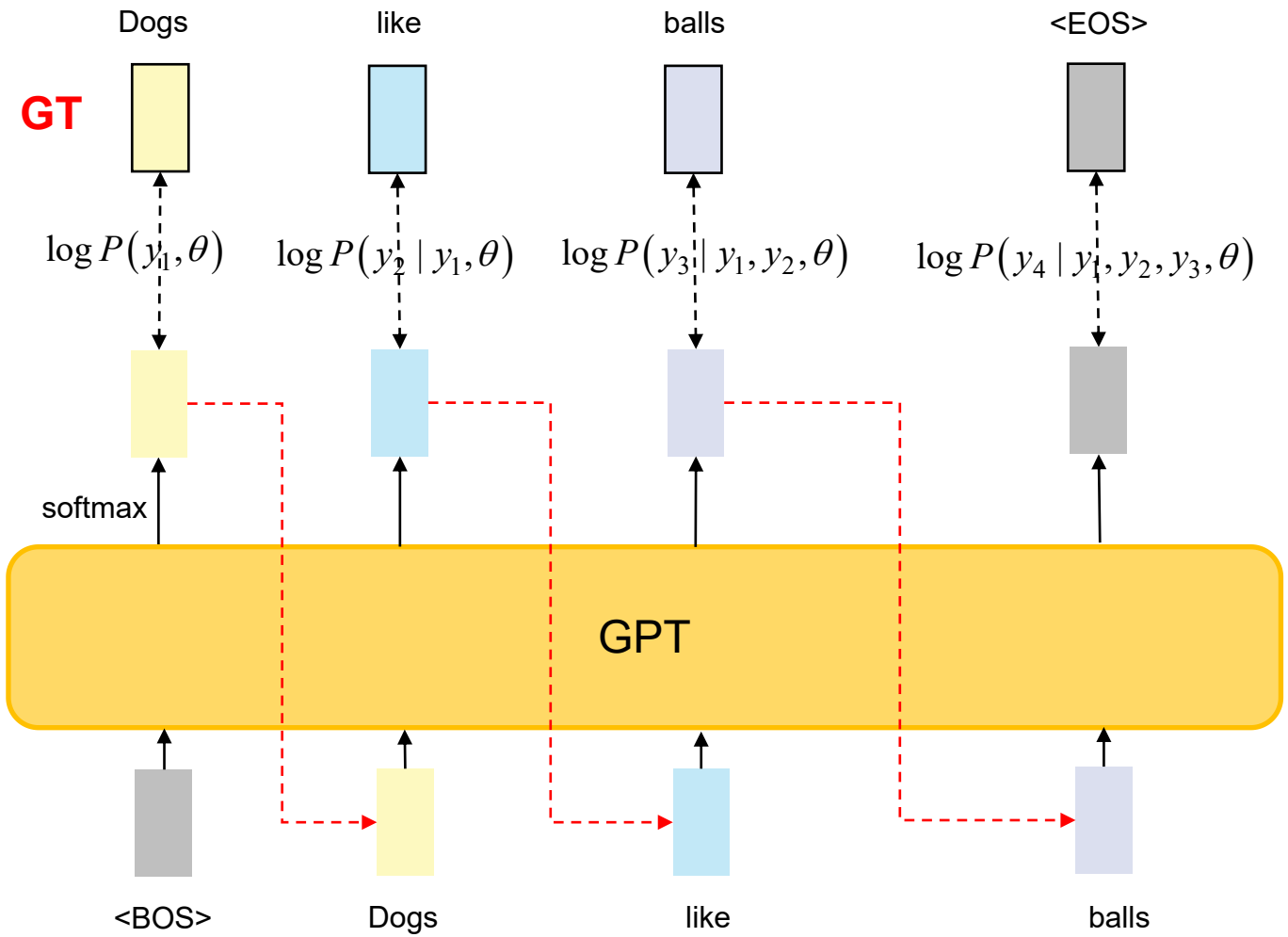
解码器结构-GPT家族

2.自回归预训练

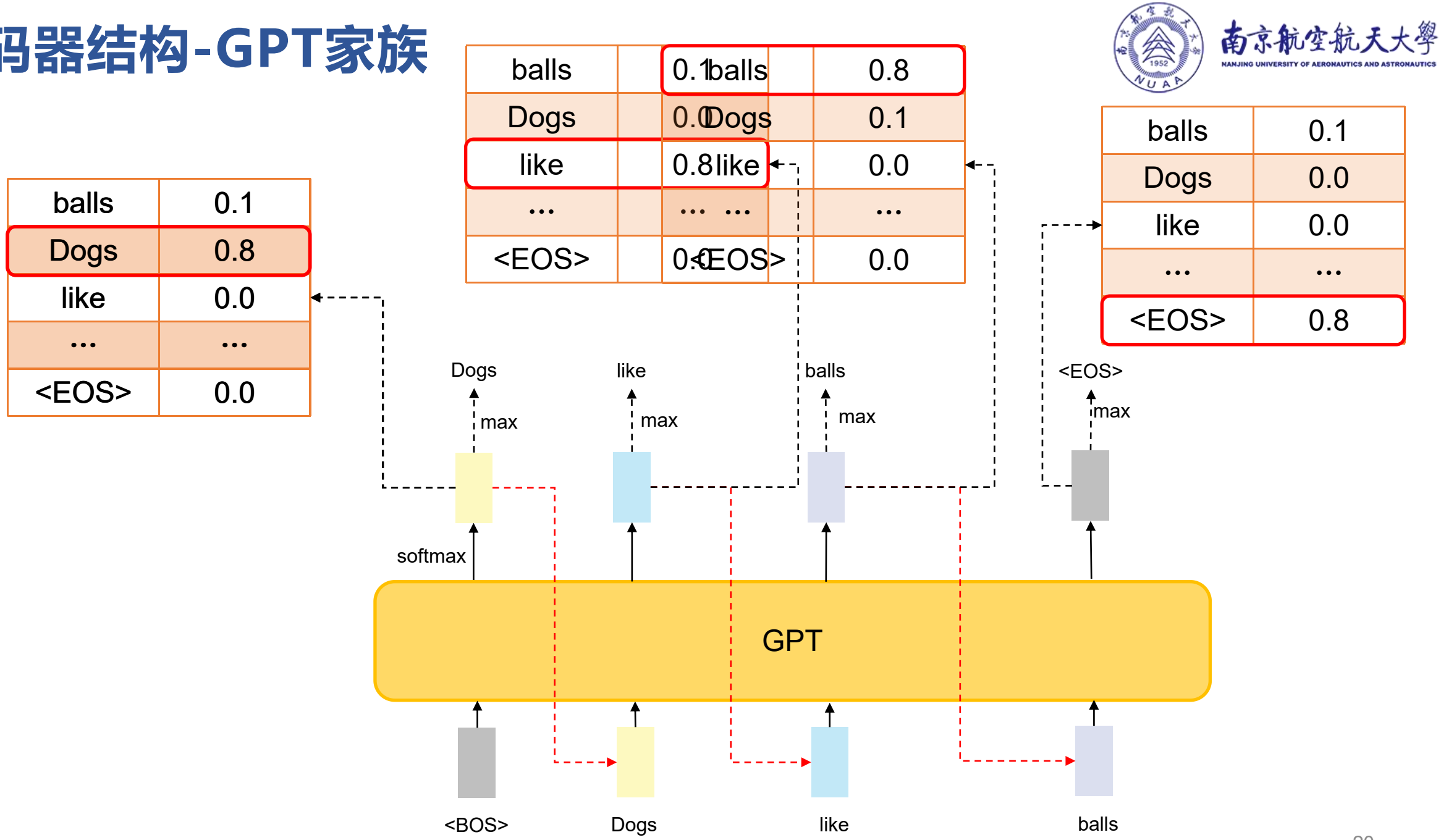
自监督学习

$$-\sum_{t=1}^{n}\log P(y_t \mid y_{<t},\theta)$$





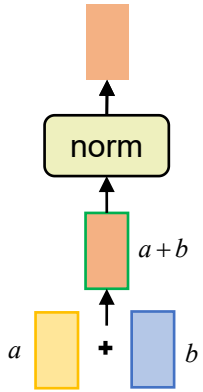
解码器结构-GPT家族

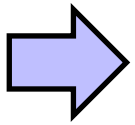

balls 0.1
Dogs 0.8
like 0.0
... ...
<EOS> 0.0
balls 0.1 balls 0.8
Dogs 0.0 Dogs 0.1
like 0.8 like 0.0
... ...
<EOS> 0.0 <EOS> 0.0
balls 0.1
Dogs 0.0
like 0.0
... ...
<EOS> 0.8
Dogs
like
balls
<EOS>
max
max
max
max
softmax
GPT
<BOS>
Dogs
like
balls

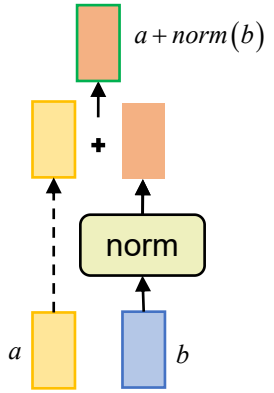
解码器结构-GPT家族

3.后续改进

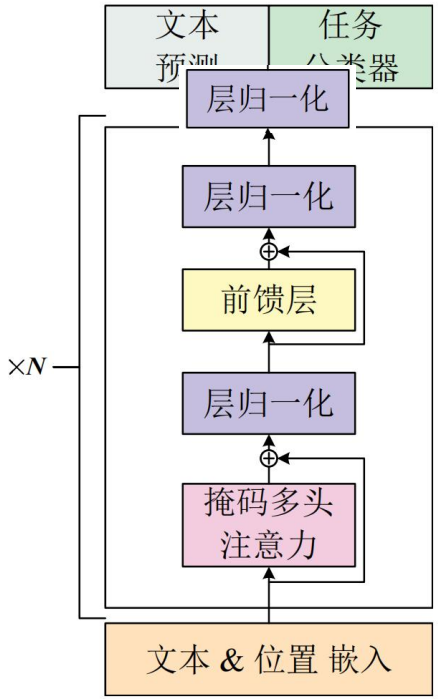
GPT-2







1)



缩放参数

增大输入序列长度

解码器结构-GPT家族

GPT-2移除了微调模型，完全只做无监督预训练　　zero-shot 零样本学习

2)

$$-\sum_{t=1}^{n}\log P(y_t|y_{<t},\theta) \quad \Rightarrow \quad P(x)=\prod_{i=1}^{n}P(x_n|x_1,x_2,...,x_{n-1})$$

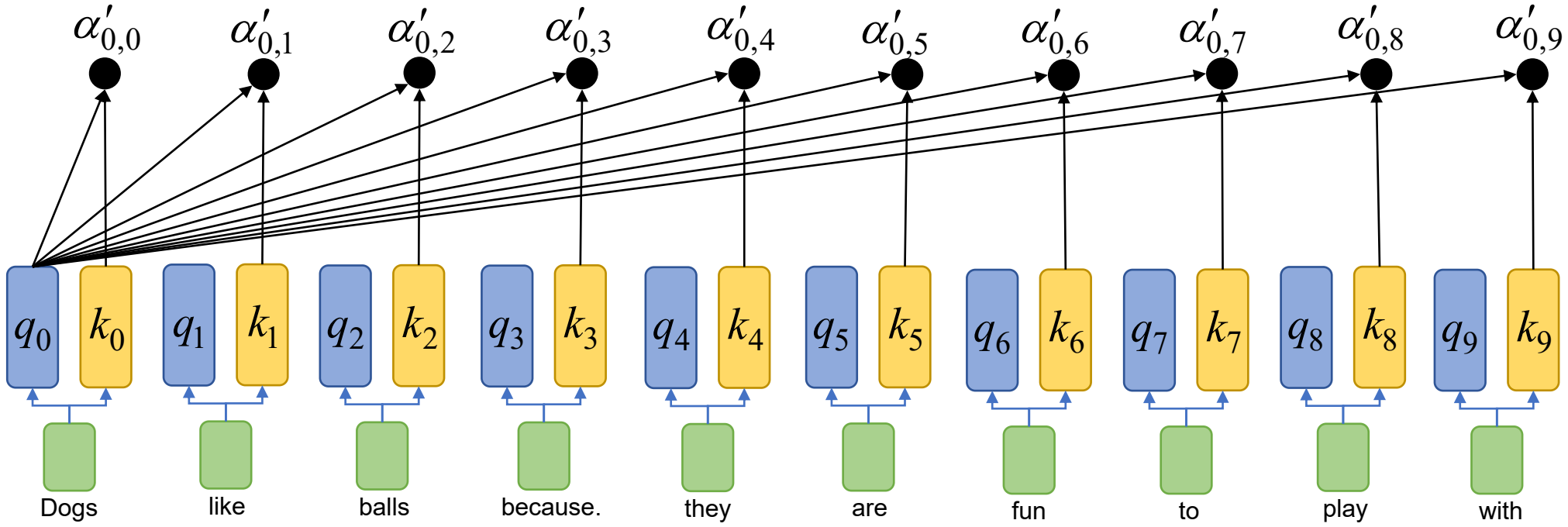
3)　　更大规模的预训练数据集

解码器结构-GPT家族

GPT-3

1)

k=2



解码器结构-GPT家族

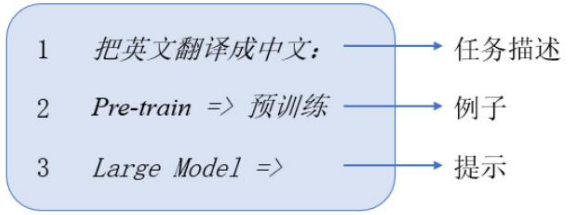
2)

是否进行微调和使用多少数据进行微调

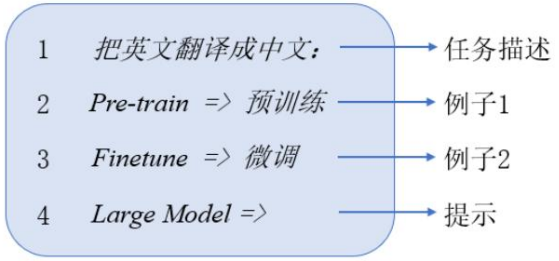
In-Context Learning的三种设置

零样本：该模型仅根据任务的自然语言描述来预测答案，不进行梯度更新。

一样本：除了任务描述外，模型还能看到任务的单个示例，不进行梯度更新。

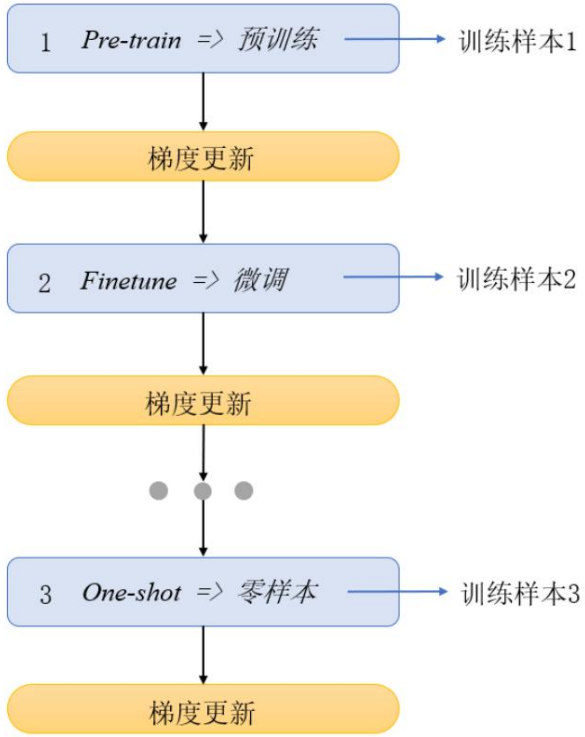


少样本：除了任务描述外，模型还能看到一些任务示例（两个及以上），不进行梯度更新。



传统方法中的微调设置（不在GTP-3里使用）

微调：该模型通过使用大量的示例任务库进行反复梯度更新来训练。



谢谢!
Thanks!