Bert

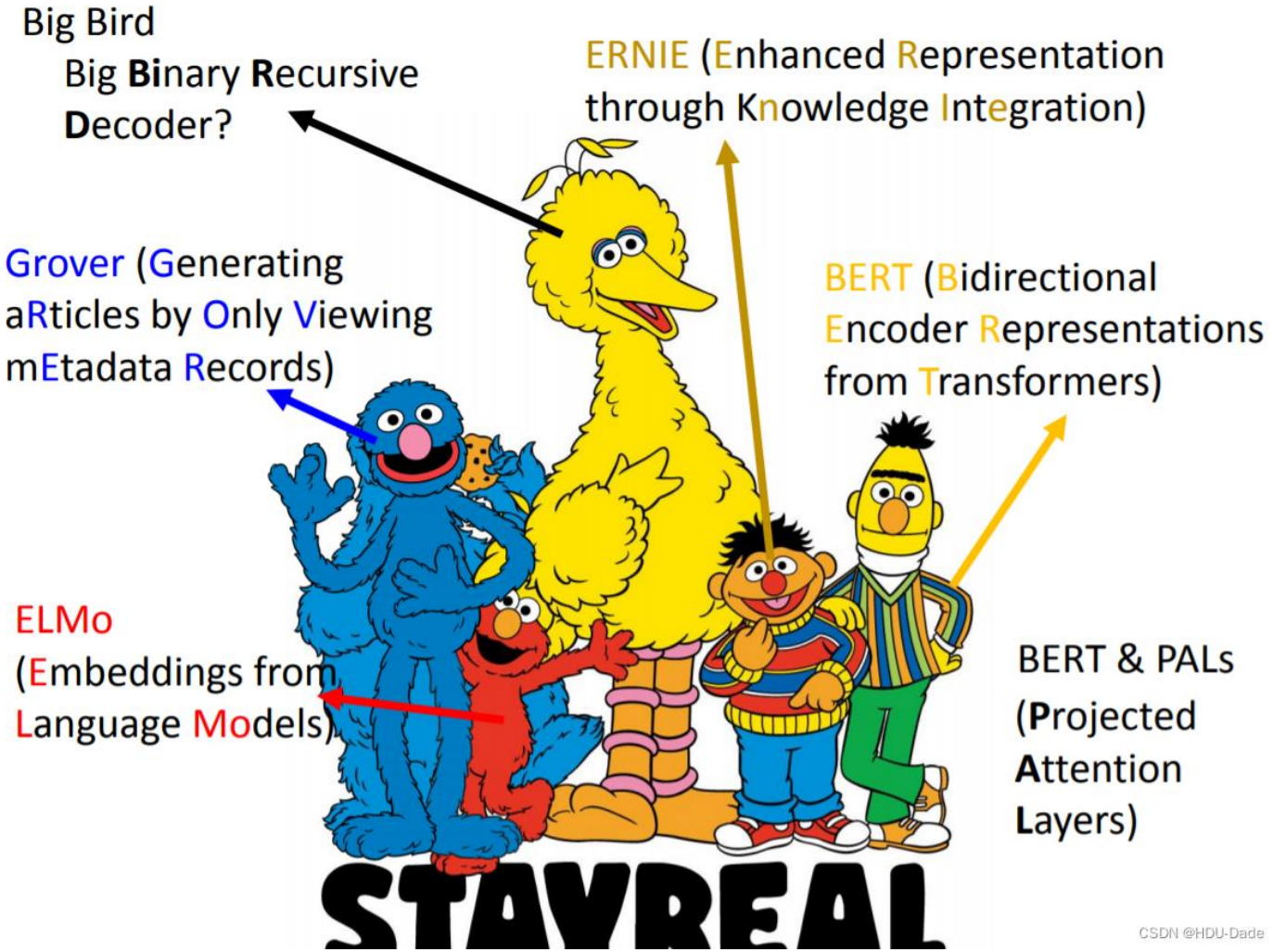
自然语言处理（NLP）

- Textual entailment(文本蕴涵)
- Question answering(问题回答)
- Semantic similarity assessment(语义相似度评估)
- Document classification(文本分类)
- Machine translation(机器翻译)

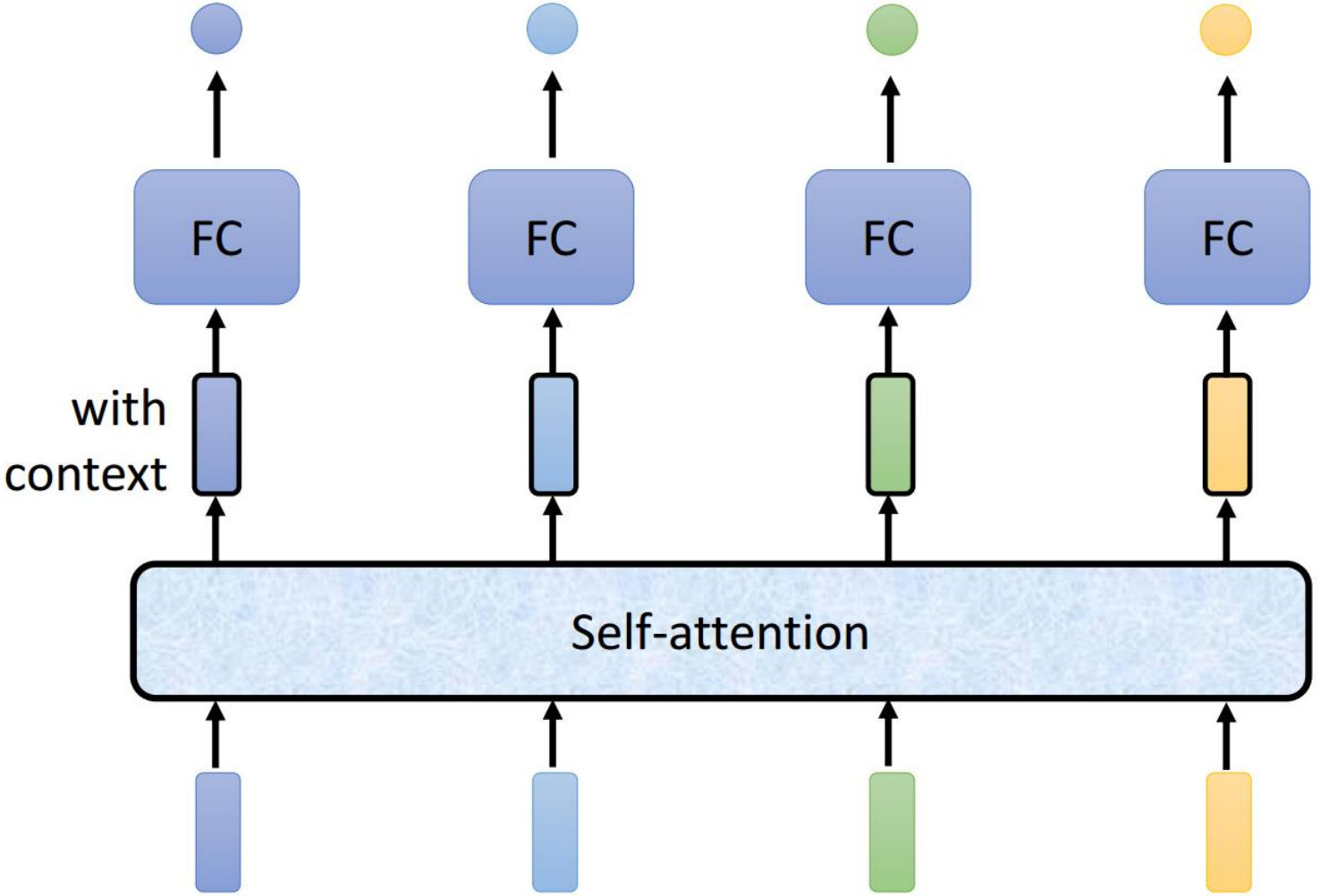
Transformer

Bert

芝麻街系列NLP模型

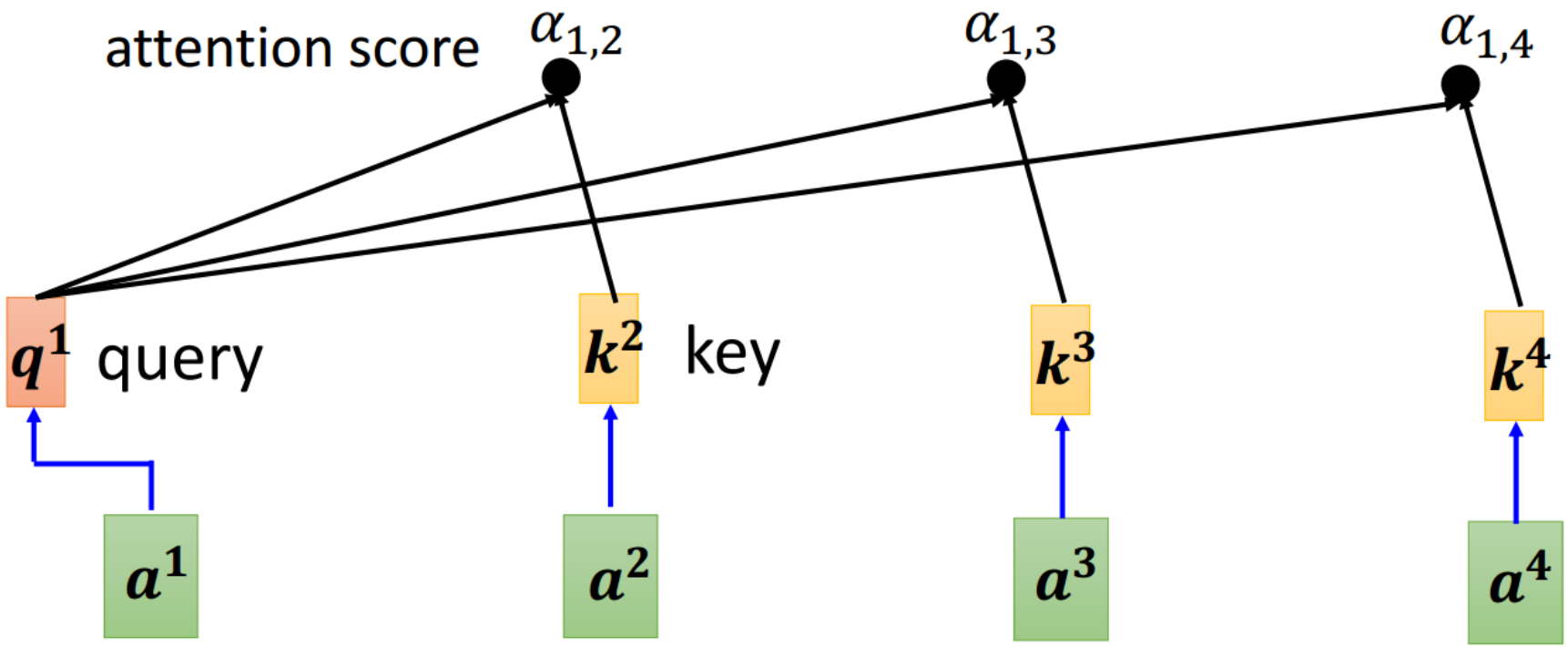

Big Bird
Big Binary Recursive Decoder?
ERNIE (Enhanced Representation through Knowledge Integration)
Grover (Generating aRticles by Only Viewing mEtadata Records)
BERT (Bidirectional Encoder Representations from Transformers)
ELMo (Embeddings from Language Models)
BERT & PALs (Projected Attention Layers)
STAYREAL
CSDN @HDU-Dade

注意力机制



注意力机制

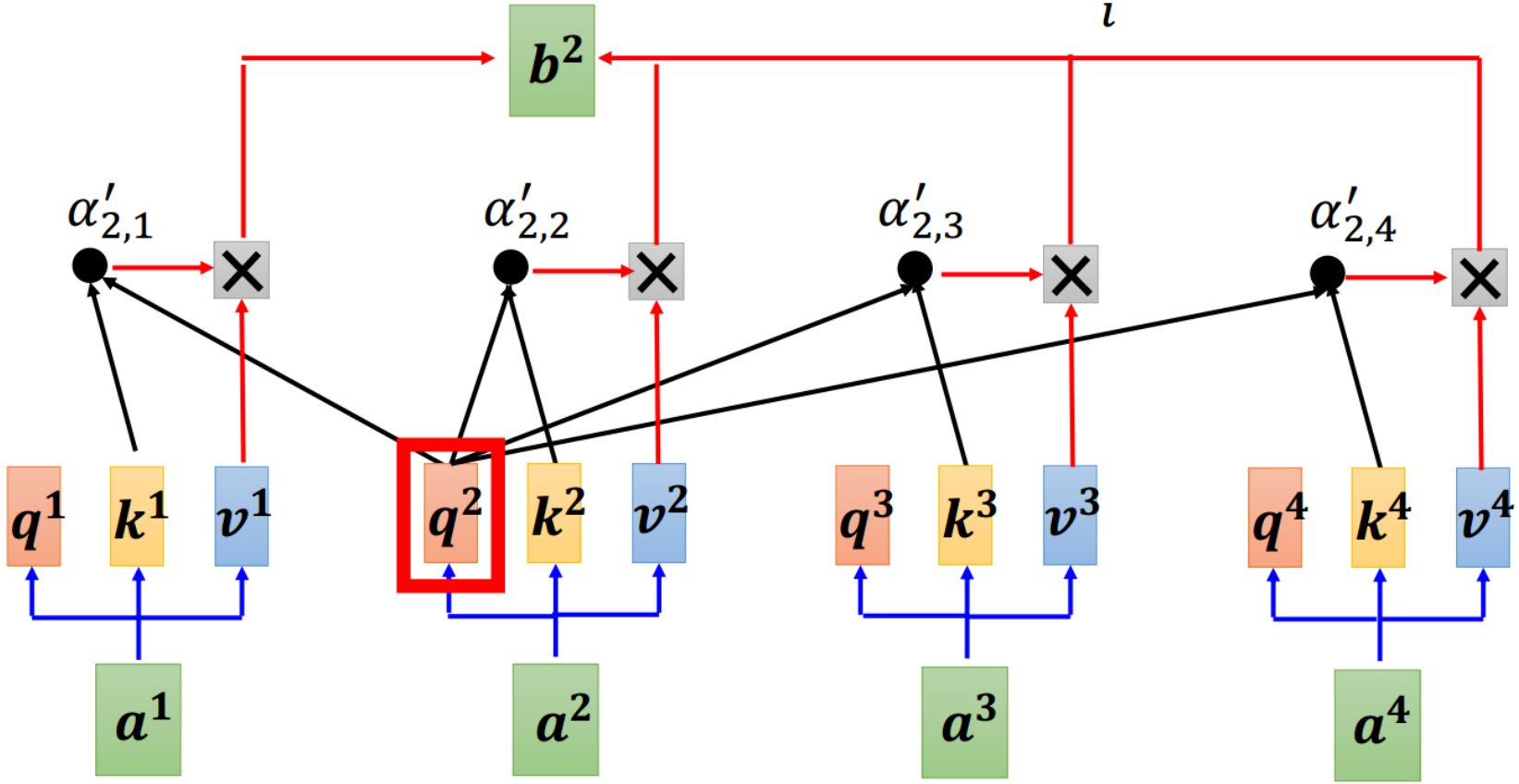
$$\alpha_{1,2} = \boldsymbol{q^1} \cdot \boldsymbol{k^2} \qquad \alpha_{1,3} = \boldsymbol{q^1} \cdot \boldsymbol{k^3} \qquad \alpha_{1,4} = \boldsymbol{q^1} \cdot \boldsymbol{k^4}$$



$$\boldsymbol{q^1} = W^q \boldsymbol{a^1} \qquad \boldsymbol{k^2} = W^k \boldsymbol{a^2} \qquad \boldsymbol{k^3} = W^k \boldsymbol{a^3} \qquad \boldsymbol{k^4} = W^k \boldsymbol{a^4}$$

自注意力机制

$$b^2 = \sum_i \alpha'_{2,i} v^i$$



自注意力机制

$$q^i = W^q a^i \qquad q^1 q^2 q^3 q^4 = W^q \; a^1 a^2 a^3 a^4$$

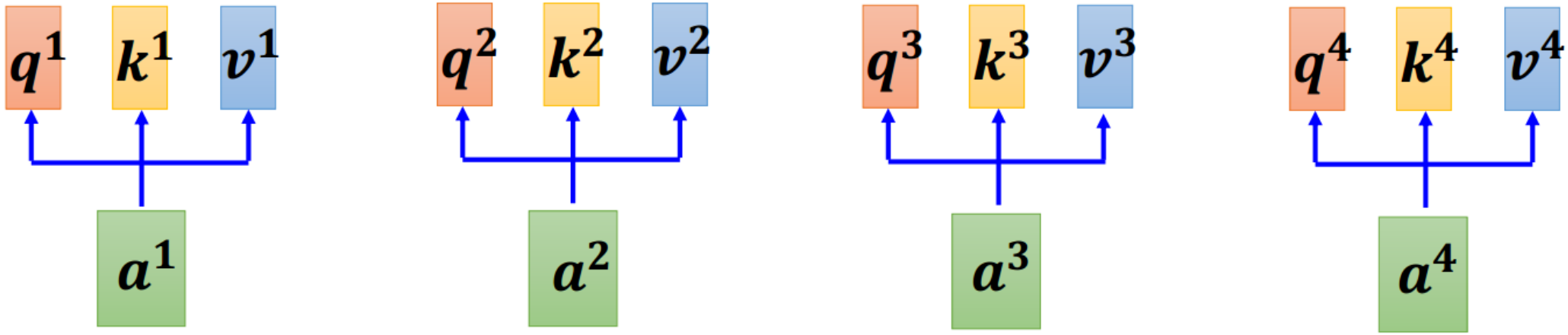
Q $\qquad$ I

$$k^i = W^k a^i \qquad k^1 k^2 k^3 k^4 = W^k \; a^1 a^2 a^3 a^4$$

K $\qquad$ I

$$v^i = W^v a^i \qquad v^1 v^2 v^3 v^4 = W^v \; a^1 a^2 a^3 a^4$$

V $\qquad$ I

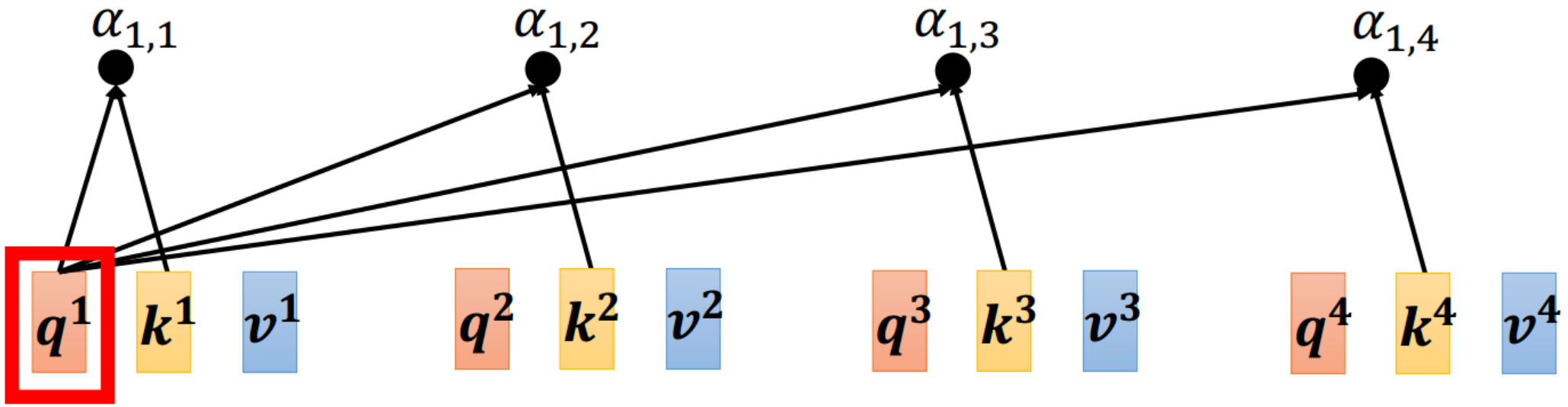


自注意力机制

$$\alpha_{1,1} = k^1 q^1 \quad \alpha_{1,2} = k^2 q^1$$

$$\alpha_{1,3} = k^3 q^1 \quad \alpha_{1,4} = k^4 q^1$$

$$\begin{matrix}\alpha_{1,1}\\ \alpha_{1,2}\\ \alpha_{1,3}\\ \alpha_{1,4}\end{matrix} = \begin{matrix}k^1\\ k^2\\ k^3\\ k^4\end{matrix} q^1$$

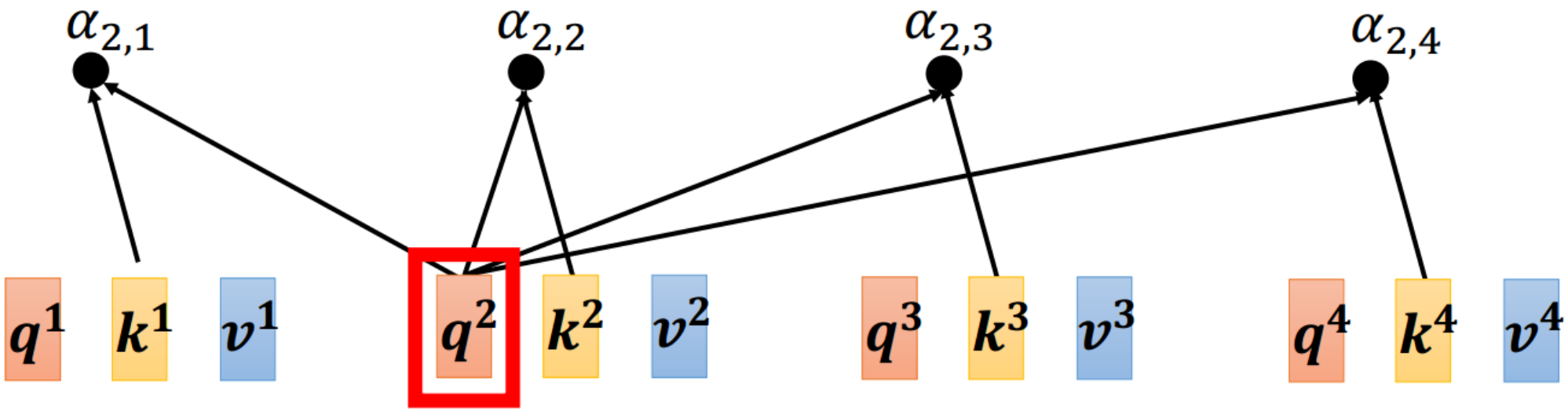


自注意力机制

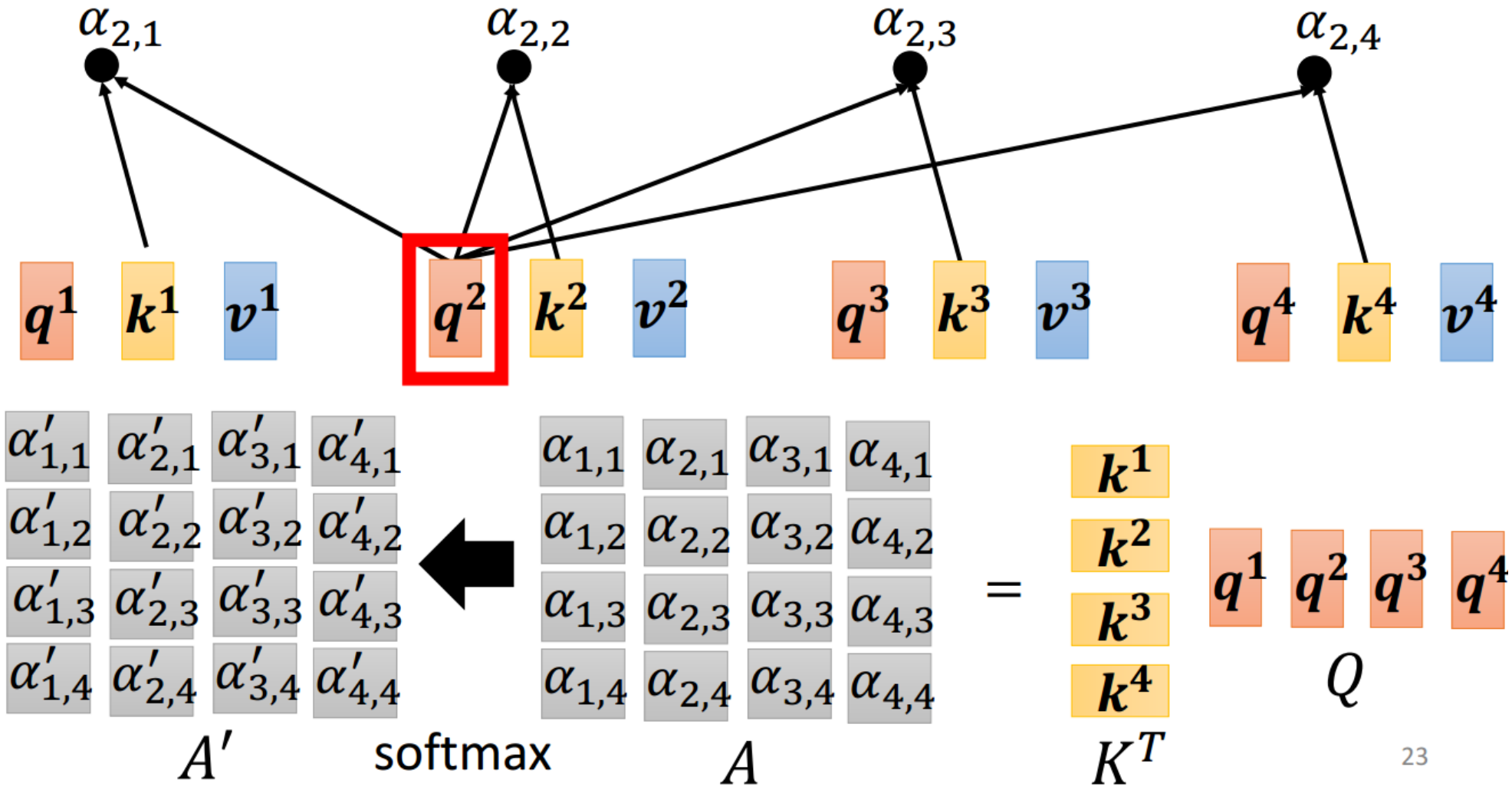
$$\alpha_{1,1} = k^1 q^1 \quad \alpha_{1,2} = k^2 q^1$$

$$\alpha_{1,3} = k^3 q^1 \quad \alpha_{1,4} = k^4 q^1$$

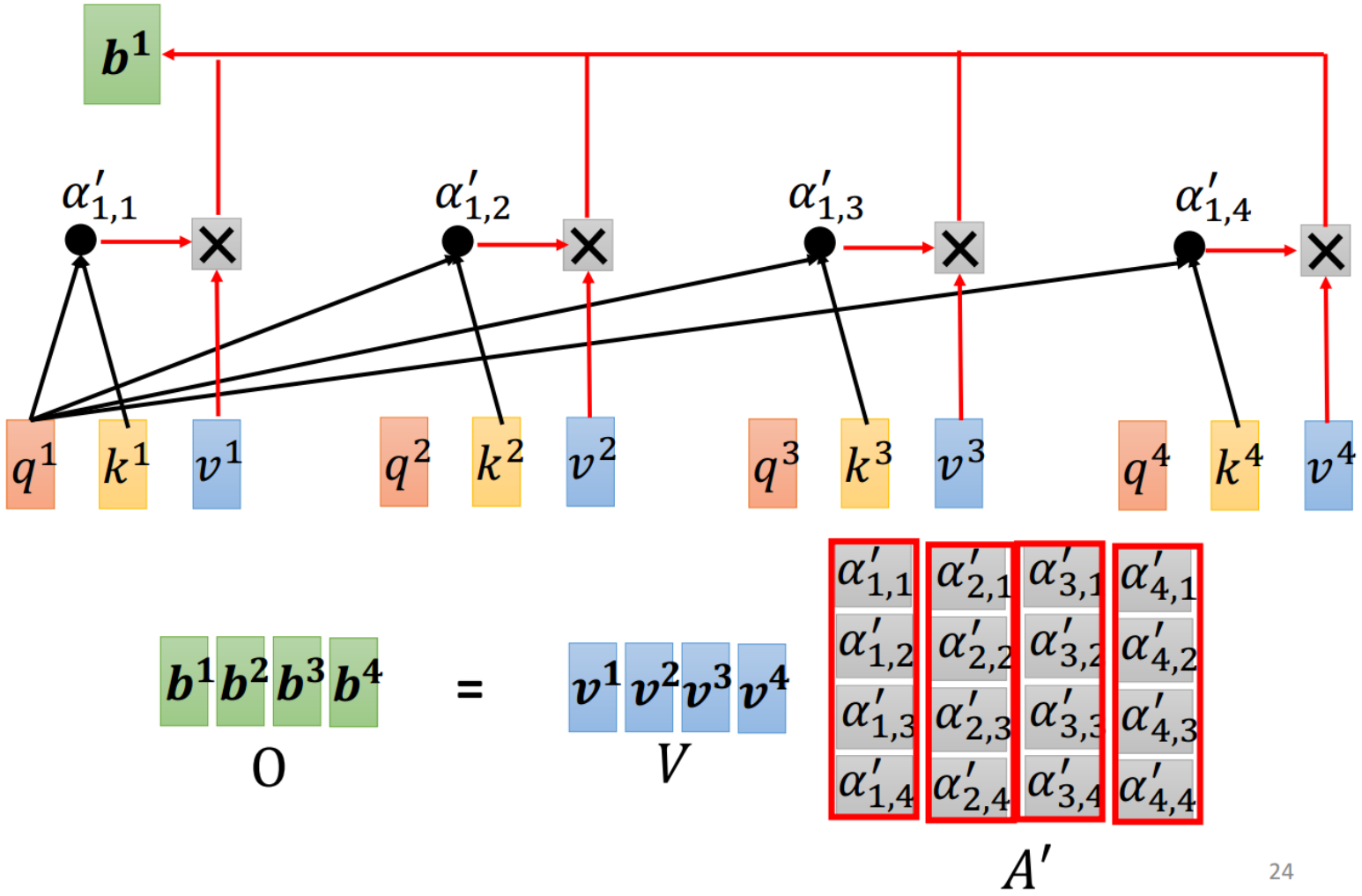
$$\begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{bmatrix} = \begin{bmatrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{bmatrix} q^1$$



自注意力机制


$\alpha_{2,1}$ $\alpha_{2,2}$ $\alpha_{2,3}$ $\alpha_{2,4}$
q^1 k^1 v^1 q^2 k^2 v^2 q^3 k^3 v^3 q^4 k^4 v^4
A'
softmax
A
K^T
Q
23

自注意力机制

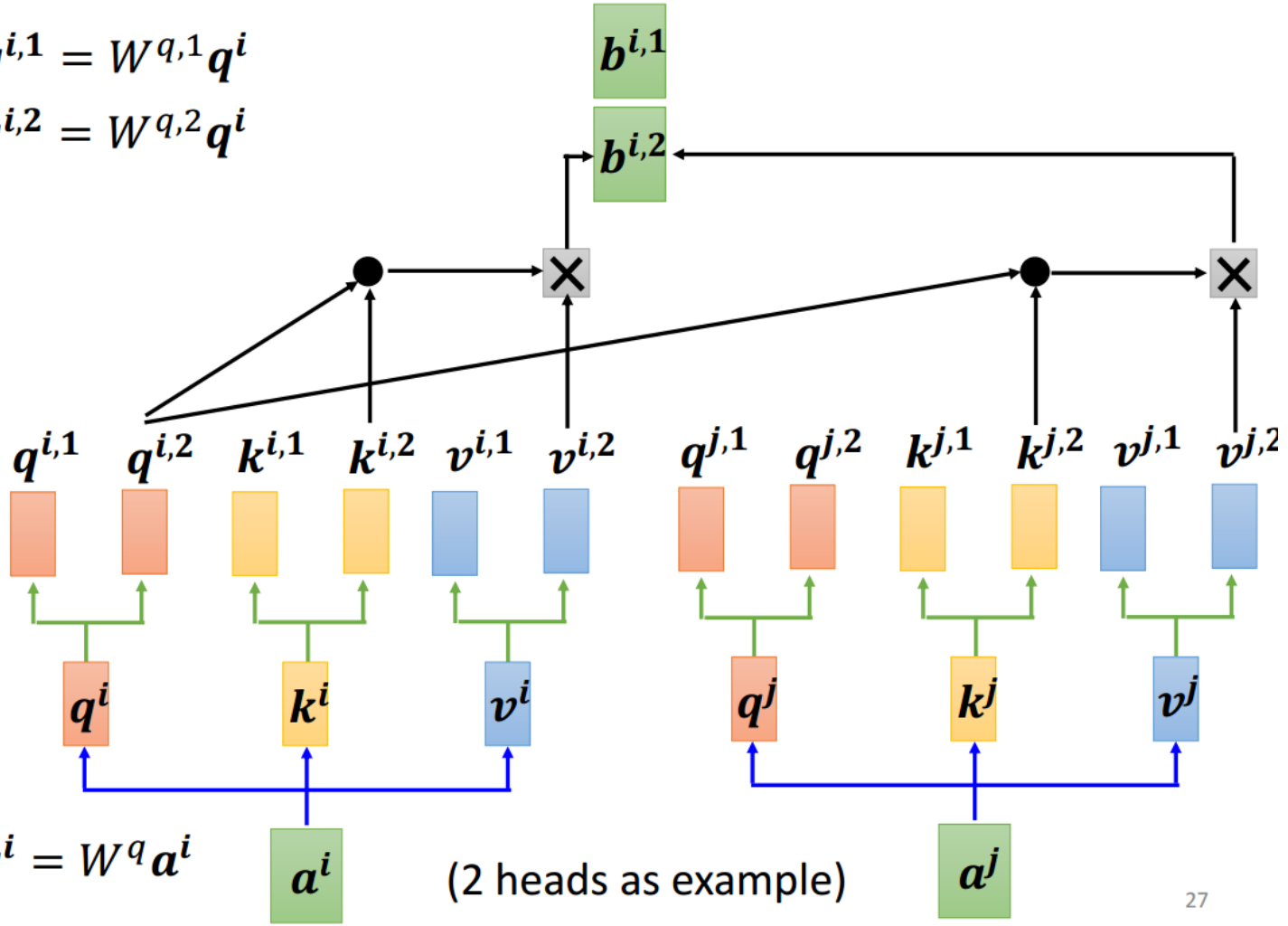

b^1
$\alpha'_{1,1}$
$\alpha'_{1,2}$
$\alpha'_{1,3}$
$\alpha'_{1,4}$
q^1 k^1 v^1
q^2 k^2 v^2
q^3 k^3 v^3
q^4 k^4 v^4
$b^1 b^2 b^3 b^4$ = $v^1 v^2 v^3 v^4$
O
V
$\alpha'_{1,1}$ $\alpha'_{2,1}$ $\alpha'_{3,1}$ $\alpha'_{4,1}$
$\alpha'_{1,2}$ $\alpha'_{2,2}$ $\alpha'_{3,2}$ $\alpha'_{4,2}$
$\alpha'_{1,3}$ $\alpha'_{2,3}$ $\alpha'_{3,3}$ $\alpha'_{4,3}$
$\alpha'_{1,4}$ $\alpha'_{2,4}$ $\alpha'_{3,4}$ $\alpha'_{4,4}$
A'

24

多头自注意力机制

$$b^i = W^O \begin{bmatrix} b^{i,1} \\ b^{i,2} \end{bmatrix}$$

$$q^{i,1} = W^{q,1} q^i$$
$$q^{i,2} = W^{q,2} q^i$$



$$q^i = W^q a^i$$

(2 heads as example)

27

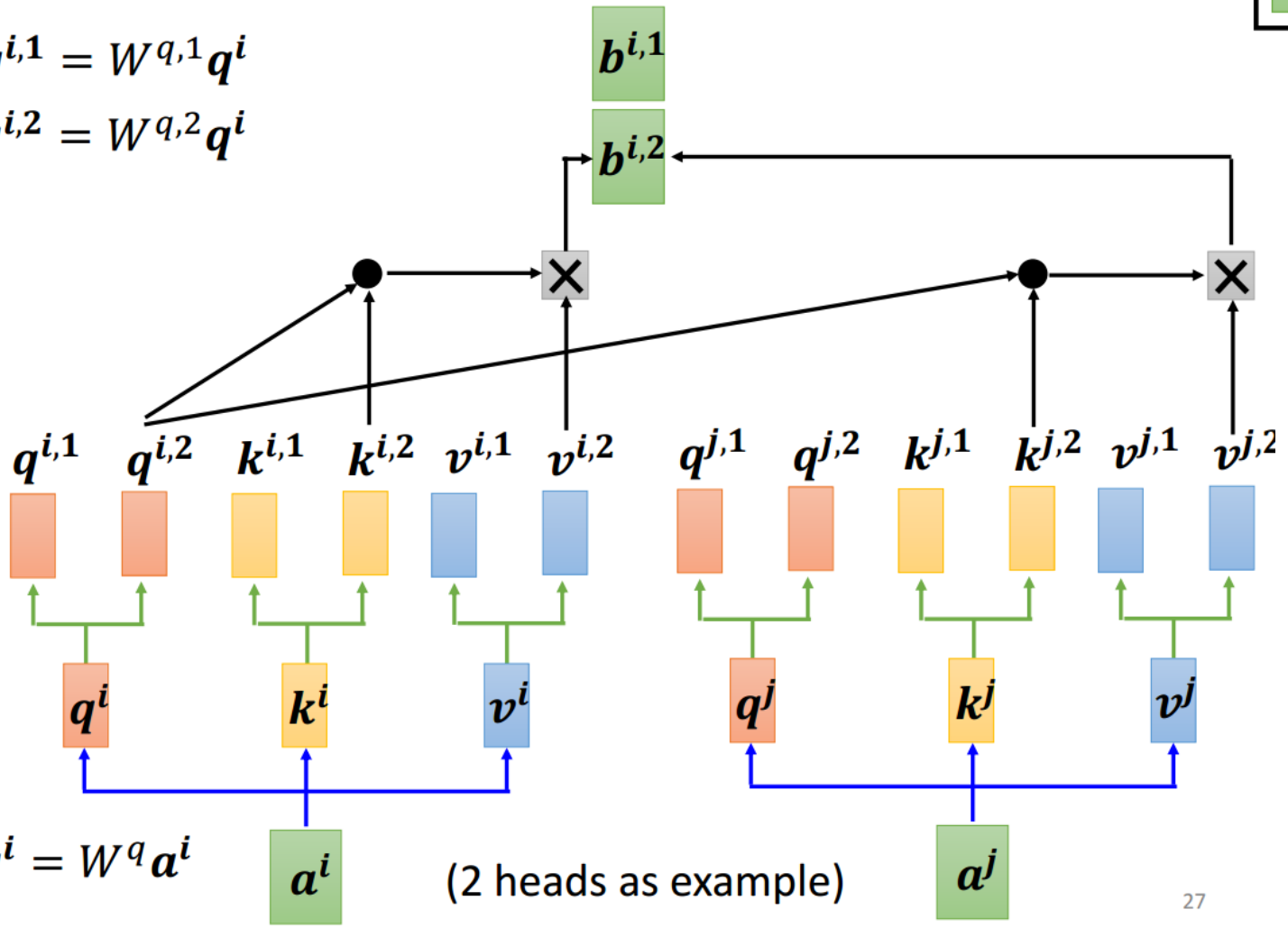
多头自注意力机制

$$b^i = W^O \begin{bmatrix} b^{i,1} \\ b^{i,2} \end{bmatrix}$$

$$\begin{bmatrix} Q \\ K \\ V \end{bmatrix} = \begin{bmatrix} W^q \\ W^k \\ W^v \end{bmatrix} I$$

$$q^{i,1} = W^{q,1} q^i$$

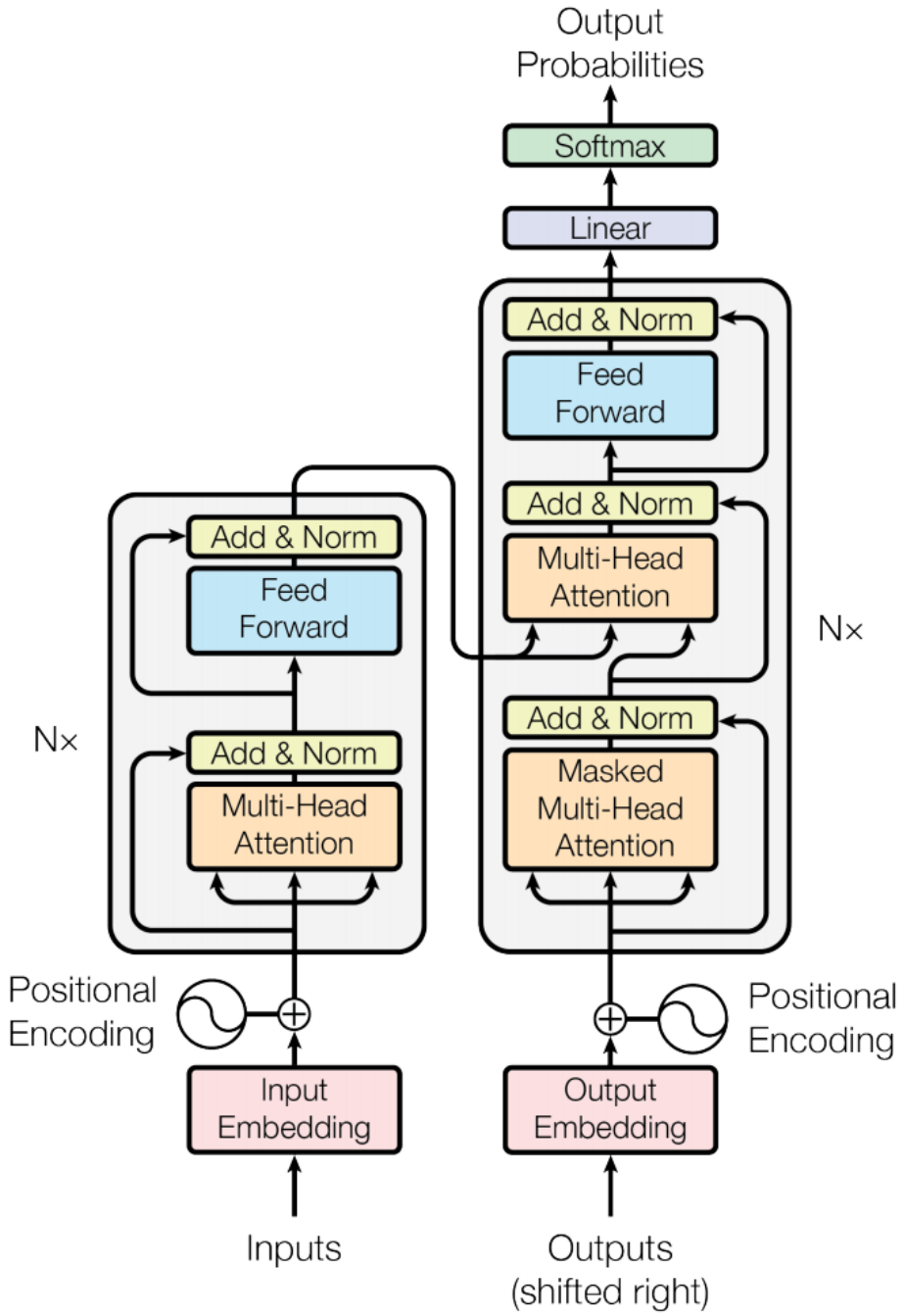
$$q^{i,2} = W^{q,2} q^i$$



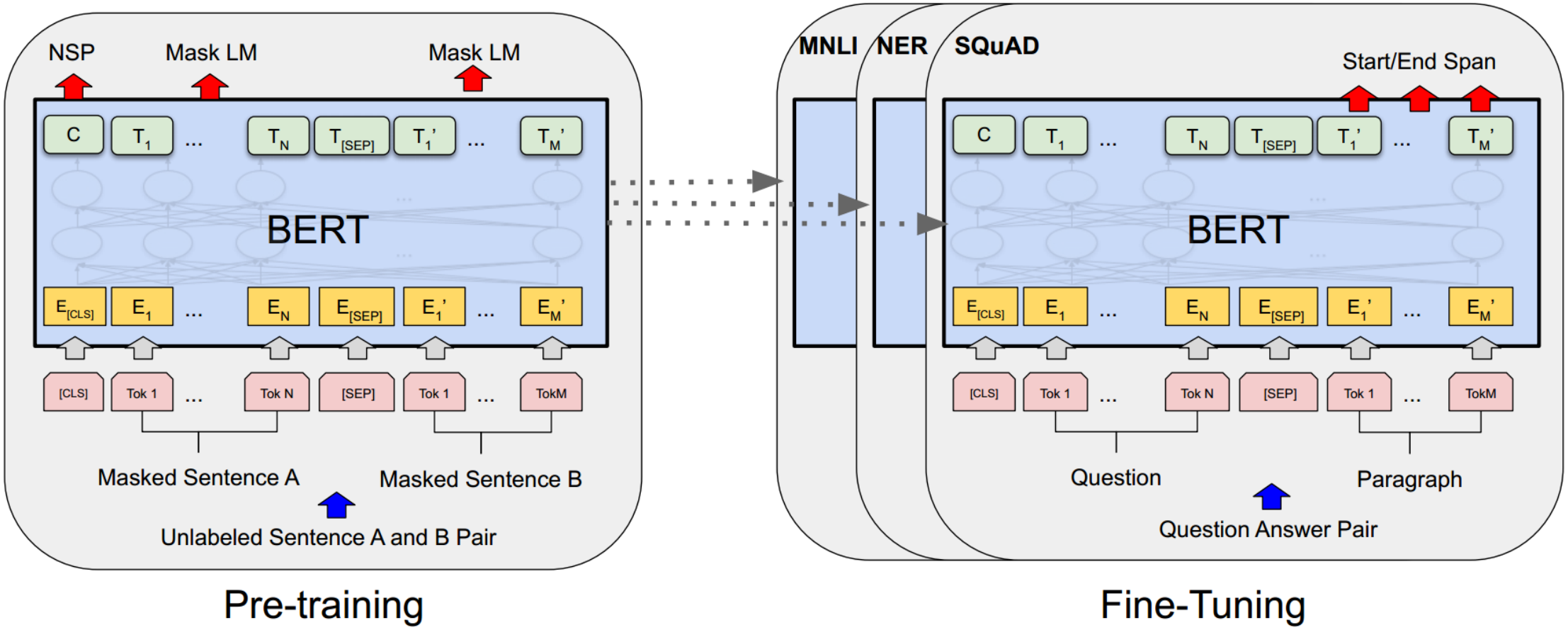
$$q^i = W^q a^i$$

27

Transformer



Bert


NSP
Mask LM
Mask LM
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
TokM
Masked Sentence A
Masked Sentence B
Unlabeled Sentence A and B Pair
Pre-training
MNLI
NER
SQuAD
Start/End Span
Question
Paragraph
Question Answer Pair
Fine-Tuning

预训练初始化（Pre-training）

- 通常情况下，一个已经在大规模数据上训练过的模型可以提供一个好的参数初始值，一个好的初始值会使得网络收敛到一个泛化能力高的局部最优解.
- 预训练初始化（Pretrained Initialization）通常指的是在深度学习模型训练之前，使用预训练模型的权重作为初始权重。这种方法在自然语言处理（NLP）和计算机视觉等领域中非常常见
- **更快的收敛、 更好的性能、 减少过拟合**

微调（Fine-Tuning）

- 对一个已经在相关任务上预训练好的模型（预训练模型）进行额外的训练，以适应一个新的、更具体的任务（目标任务）。
- 通过利用预训练模型在大量数据上学习到的通用特征，来提高模型在新任务上的性能。

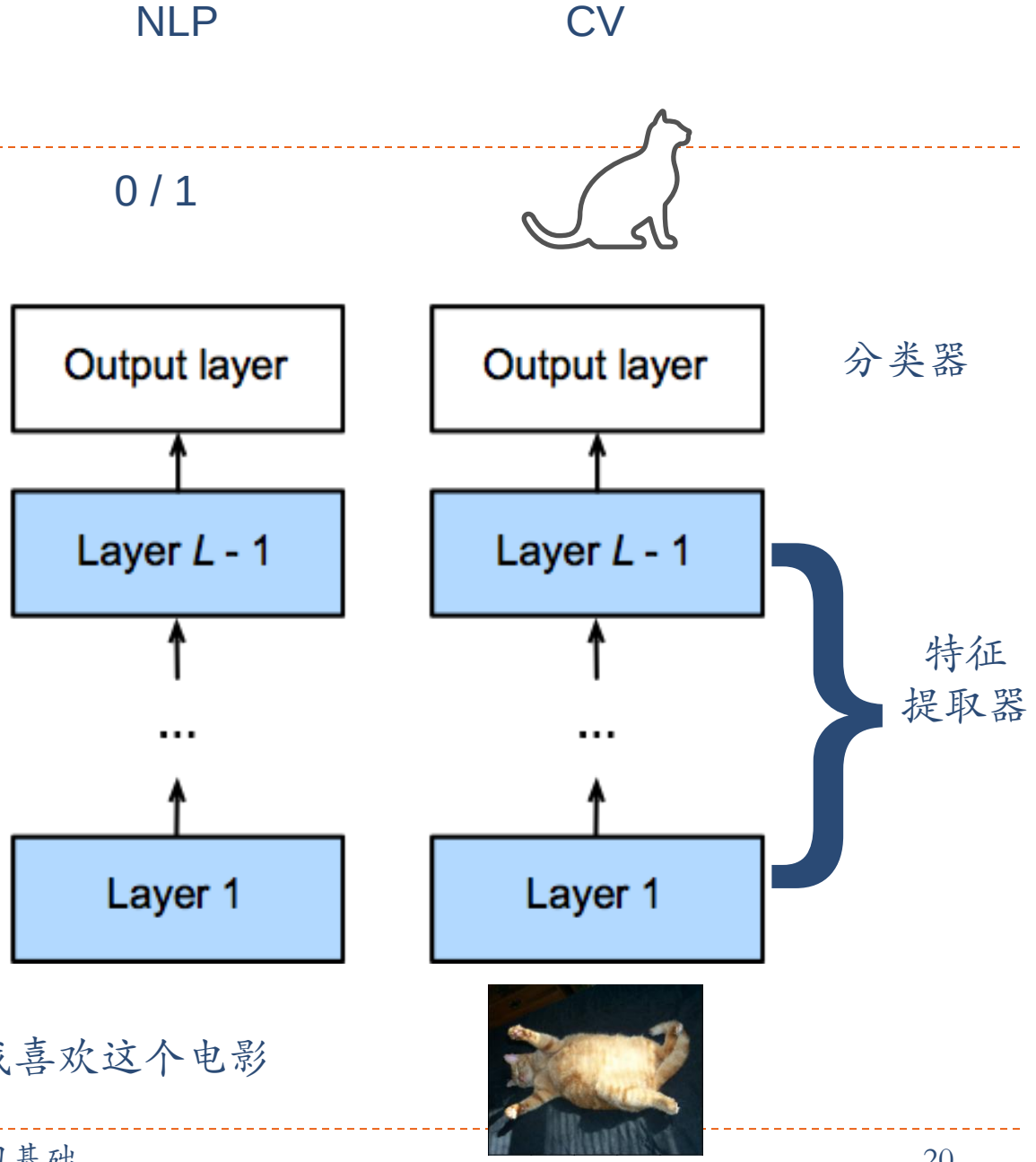
迁移学习（Transfer Learning）

- 迁移学习：**预训练模型-> 微调**

- 使用预先训练的模型为新任务提取单词/句子功能
- 需要构建一个新模型来捕获新任务所需的信息
- 通常不更新预先训练的模型的权重

BERT 的动机

- 基于微调的 NLP 方法
- 预训练模型捕获足够多的数据信息
- 只需要为新任务添加简单的输出层

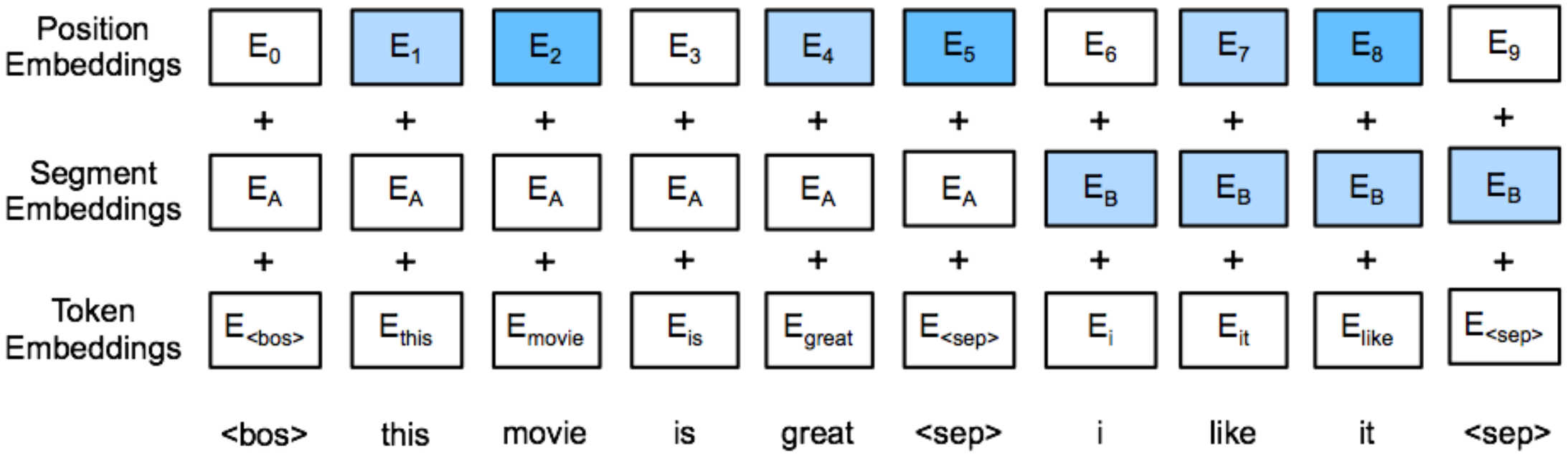


BERT 架构

- 一个（巨大的）变换器模型编码器（没有解码器）
- 两种模型大小：
  - 基础版：#blocks = 12，隐含大小= 768，#heads = 12，#参数 = 110M
  - 增强版：#blocks = 24，隐含大小= 1024，#heads = 16，#参数= 340M
- 使用超过30亿单词的大型语料库（书籍和维基百科）训练

输入

- 每个样本都是一对句子
- 添加其他细分嵌入



预训练任务1：掩码语言模型

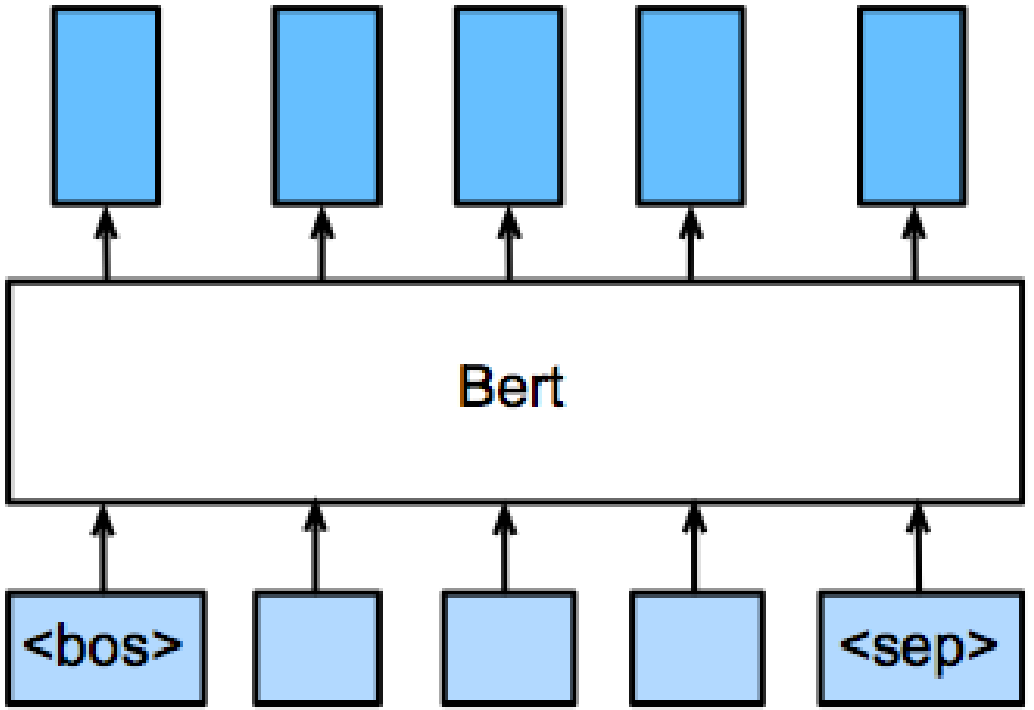
- 在每个句子中随机掩盖（例如15%）标记，预测这些掩码标记（<mask>）
  - 变换器模型是双向的，它打破了标准语言模型的单向限制
- 微调任务中没有掩码标记（<mask>）
  - 80% 的时间，用 <mask> 替换选定的标记
  - 10% 的时间，用随机挑选的picked tokens替换
  - 10% 的时间，保留原始标记

预训练任务2：下一句话预测

- 50% 的时间，选择一个连续的句子对
  - <bos> 这部电影很棒 <sep> 我喜欢它 <sep>
- 50% 的时间，选择一个随机的句子对
  - <bos> 这部电影很棒 <sep> hello world <sep>
- 将变换器模型的输出 <bos> 输入到稠密层以预测它是否是顺序对（ sequential pair）

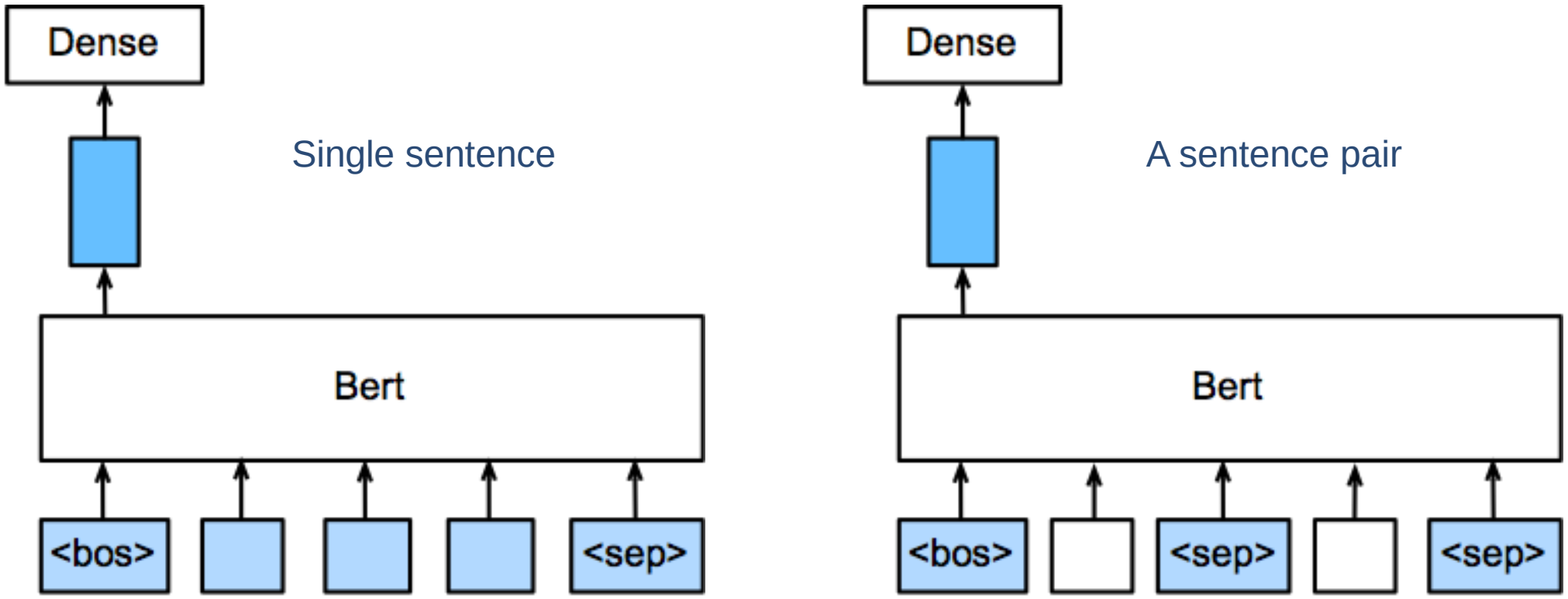
用 Bert 微调

- Bert 为捕获上下文信息的每个标记返回一个特征向量
- 不同的微调任务使用不同的向量集



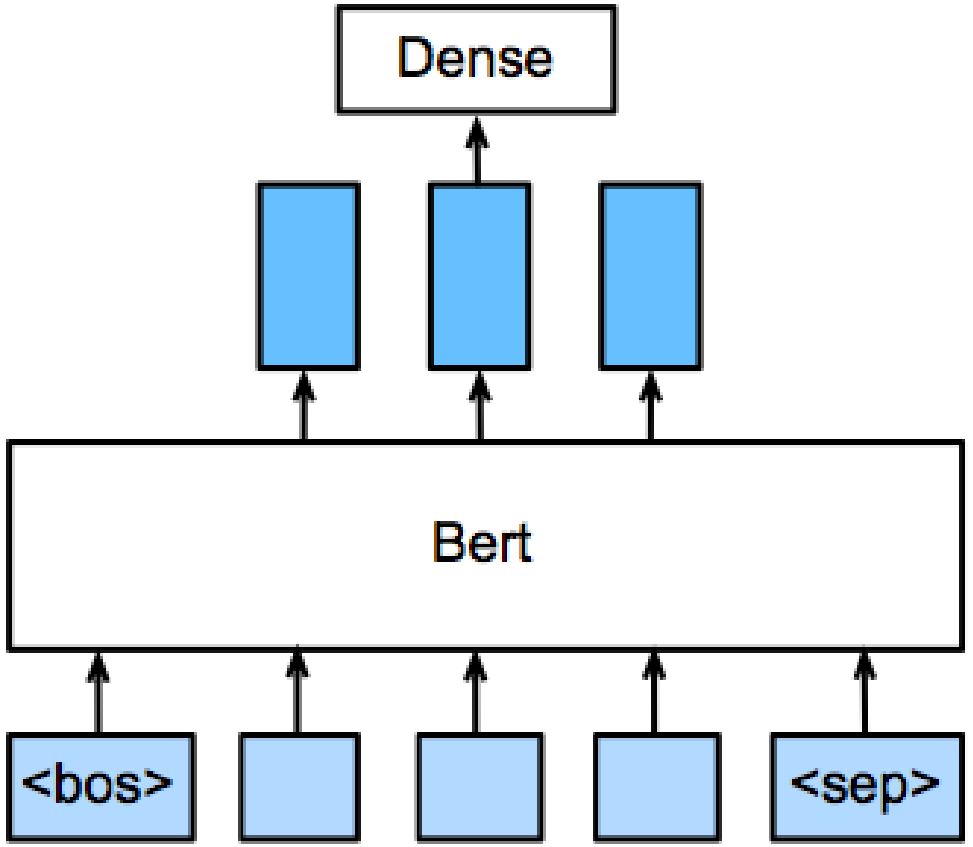
语句分类

- 将 <bos> 标记向量输入稠密输出层



命名实体识别

- 确定标记是否是命名实体，例如人员，组织和位置等等
- 将每个非特殊标记向量馈送到稠密输出层



自动问答

- 给定问题和描述文本，找到答案，这是描述中的文本段
- 给定 p_i，描述中的第 i 个标记，学习 $\mathbf{s}$ 中的 p_i，第 i 个标记是这段开始的概率：

$$p_1, \ldots, p_T = \text{softmax}(\langle \mathbf{s}, \mathbf{v}_1 \rangle, \ldots, \langle \mathbf{s}, \mathbf{v}_T \rangle)$$

- 同样可以学习第 i 个标记是这段结局的概率