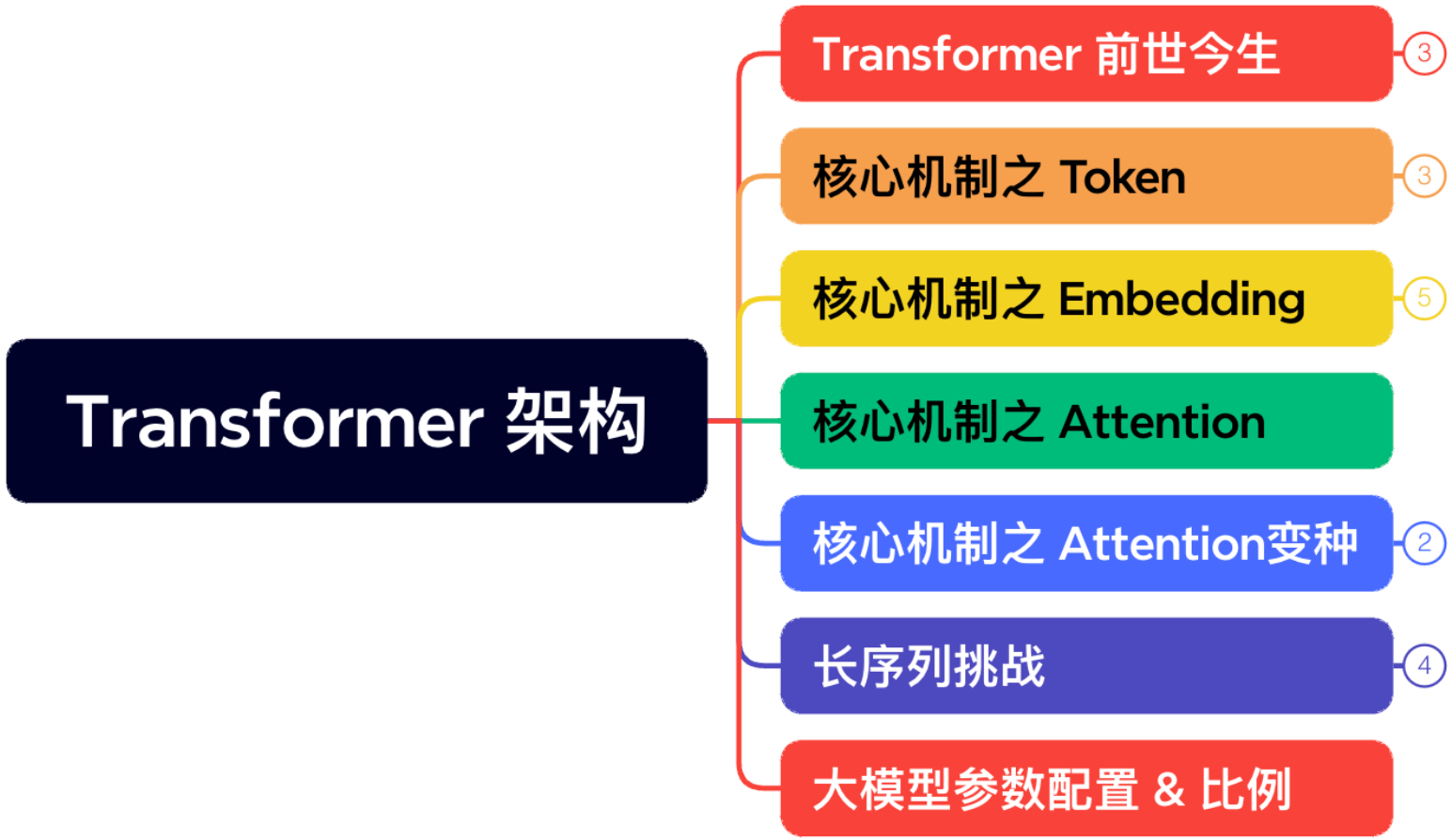



核心机制
Embedding


Contenet


Transformer 架构
Transformer 前世今生 3
核心机制之 Token 3
核心机制之 Embedding 5
核心机制之 Attention
核心机制之 Attention变种 2
长序列挑战 4
大模型参数配置 & 比例


好好想想

视频目录大纲

- 绝对位置编码（Absolute Positional Encoding）
- 相对位置编码（Relative Positional Encoding）
- 旋转位置编码（Rotary Positional Embedding, RoPE）

Embedding 核心任务

- 核心作用是将离散的输入ID转换为连续的向量表示，通过查表为每个符号赋予具有语义信息向量。

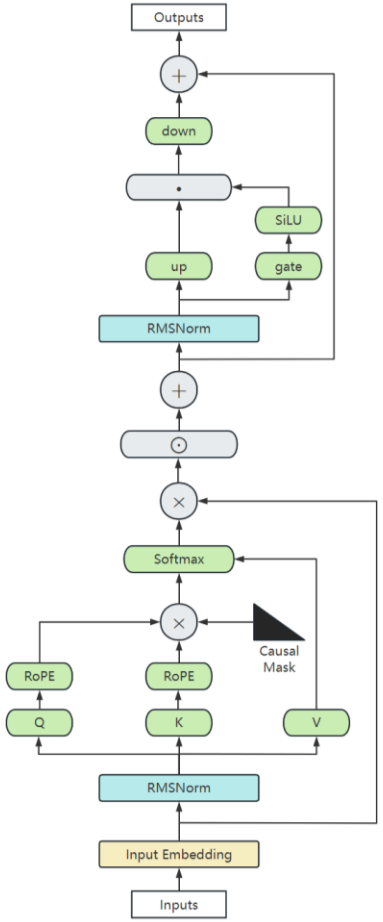
"This is a input text."

Tokenization

[CLS]	This	is	a	input	.	[SEP]
101	2023	2003	1037	7953	1012	102

Embeddings

0.0390,	-0.0558,	-0.0440,	0.0119,	0069,	0.0199,	-0.0788,
-0.0123,	0.0151,	-0.0236	-0.0037,	0.0057,	-0.0095,	0.0202
0.0208,	0.0031,	-0.0283,	-0.0402,	-0.0016,	-0.0099	-0.0352,
...	....	...	...	...	...	...



01

绝对位置编码

APE

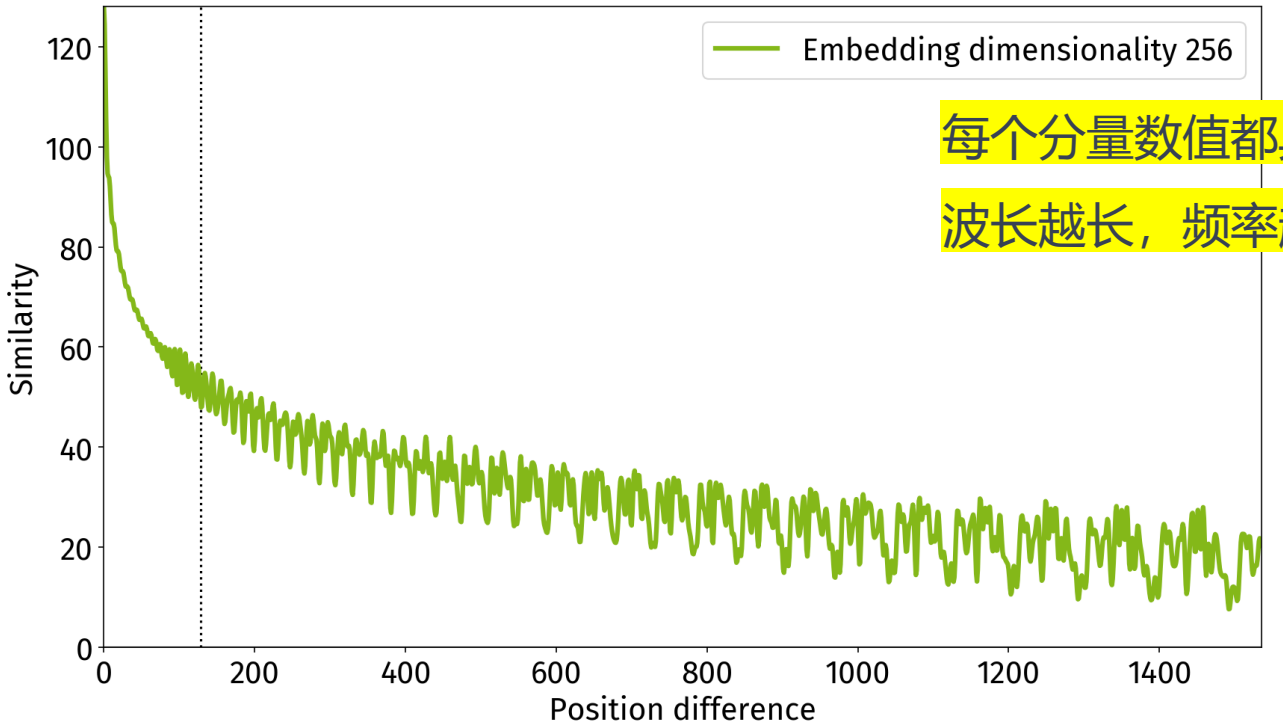
Absolute Positional Encoding

- **原理：** 为序列中每个位置分配唯一的固定或可学习向量，直接表征绝对位置索引。
  - Sinusoidal编码： Transformer 原论文采用的三角函数公式生成的固定编码，支持外推长序列 。
  - Learnable编码： 通过 Embedding 随机初始化并随训练优化的可学习参数。

- **适用任务：**
  - 短序列任务（如机器翻译、文本分类），需明确绝对位置信息 。
  - 对位置敏感的下游任务（如命名实体识别），需精准定位特定位置的上下文。

Learnable 可学习绝对位置编码

- **原理：**随机初始化位置编码矩阵作为可训练参数，模型只能感知每个词向量所处的绝对位置，无法感知词向量之间的相对位置。
- **适用：**早期预训练模型（BERT、GPT1、ALBERT），不具备长度外推性。



每个分量数值都具有周期性，越靠后分量，波长越长，频率越低。

Sinusoidal 三角式绝对位置编码

- **原理:**
  - 通过不同频率的正弦/余弦函数生成位置编码，偶数维度用正弦，奇数维度用余弦。

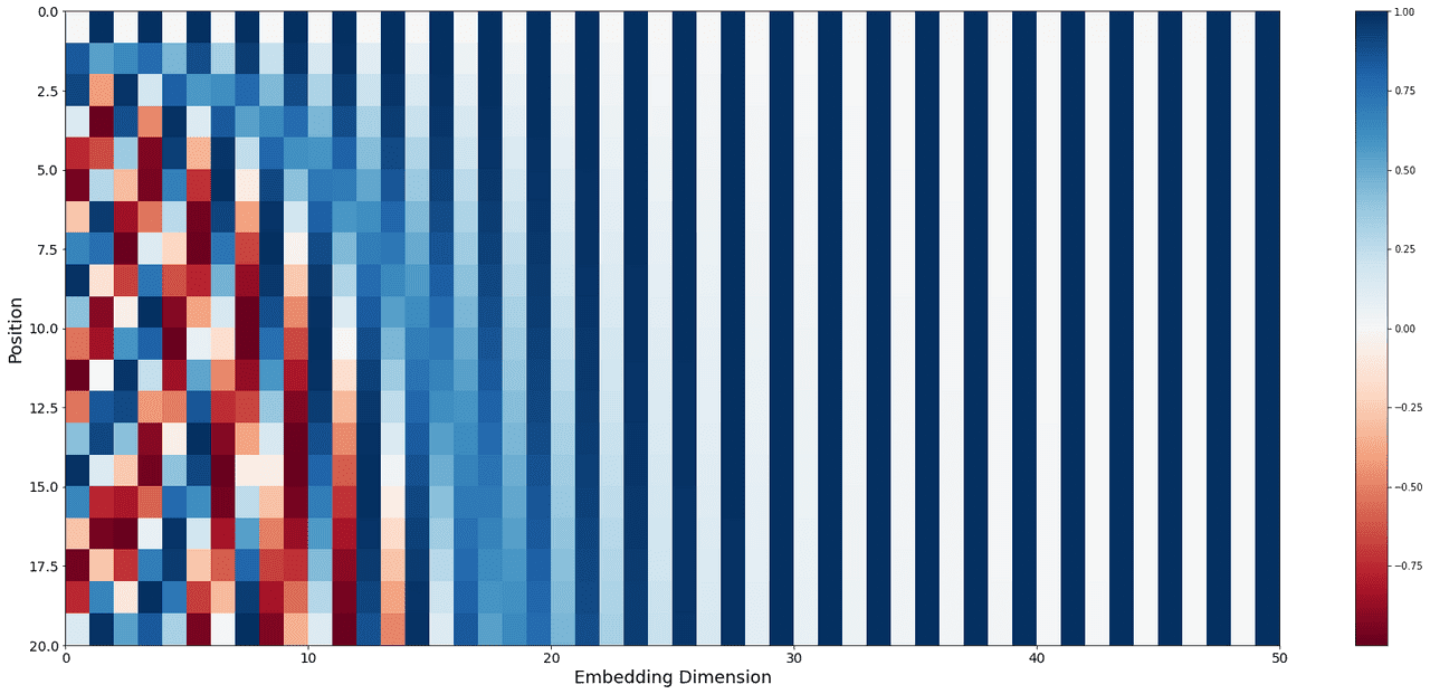
- **公式:**

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

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

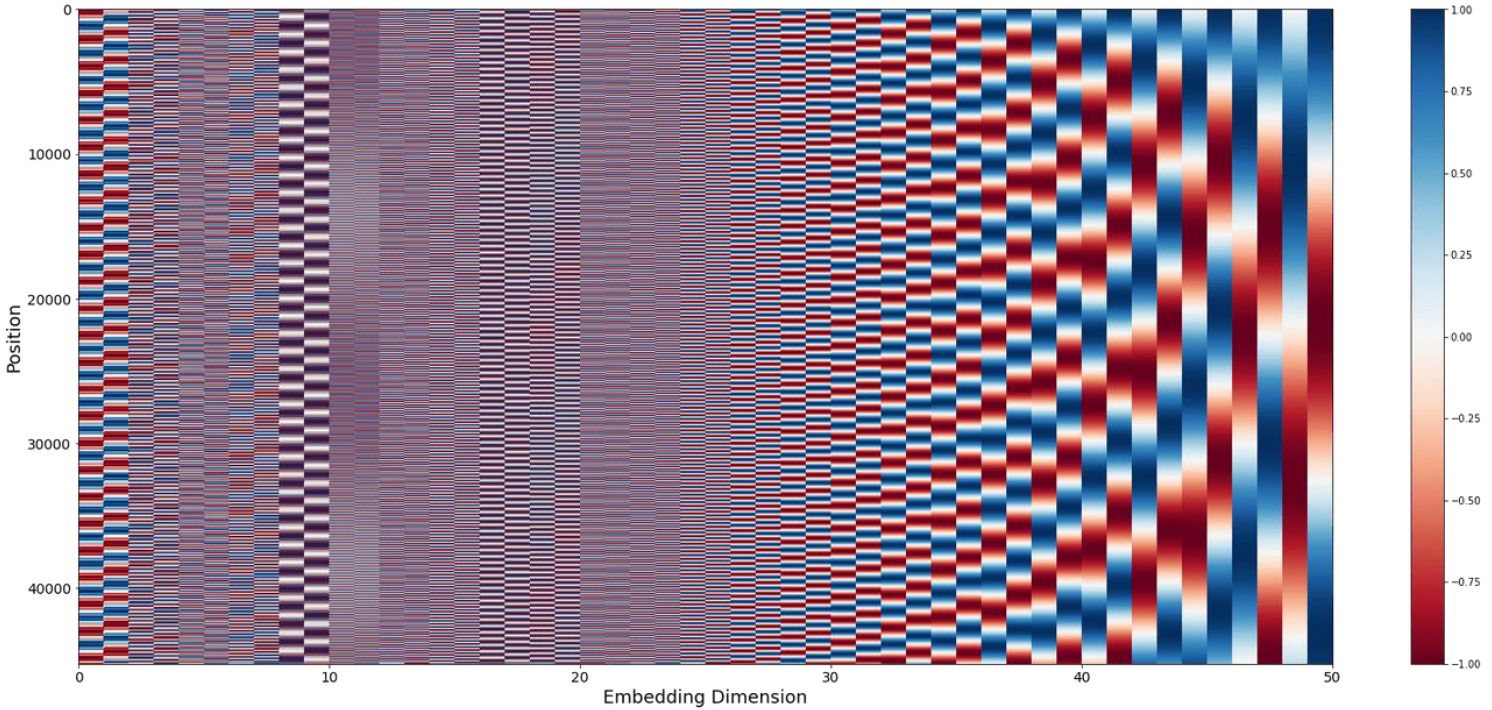
Sinusoidal 三角式绝对位置编码

- **特点：**无需训练，具有周期性外推能力，但无法直接表达相对位置关系。
- **适用：**短文本翻译（如原 Transformer）、需要固定位置感知的序列任务。
- **局限：**长序列（如超过 512）性能下降，需微调或扩展。



Sinusoidal 三角式绝对位置编码

- **特点：**无需训练，具有周期性外推能力，但无法直接表达相对位置关系。
- **适用：**短文本翻译（如原 Transformer）、需要固定位置感知的序列任务。
- **局限：**长序列（如超过 512）性能下降，需微调或扩展。



02

相对位置编码

RPE

相对位置编码

- **绝对位置编码缺点：**
  - 难以反应序列字符之间的相对位置关系。
  - 没有外推性，即表示不了比预训练文本长度更长的位置向量。

- **相对位置编码原理：**
  - 建模序列中任意两位置间的相对距离而非绝对索引，增强模型对局部结构的感知能力。

Self-Attention with RPE

- 相对位置信息在 self-attention 计算时候丢失，最直接在计算 self-attention 时候再加回来。

Self-Attention with Relative Position Representations

Peter Shaw
Google
petershaw@google.com

Jakob Uszkoreit
Google Brain
usz@google.com

Ashish Vaswani
Google Brain
avaswani@google.com

https://arxiv.org/pdf/1803.02155.pdf

Self-Attention with RPE

- 相对位置信息在 self-attention 计算时候丢失，最直接在计算 self-attention 时候再加回来。
- 计算 attention score 和 weighted value 时各加入可训练表示相对位置的参数，multi head之间可以共享。

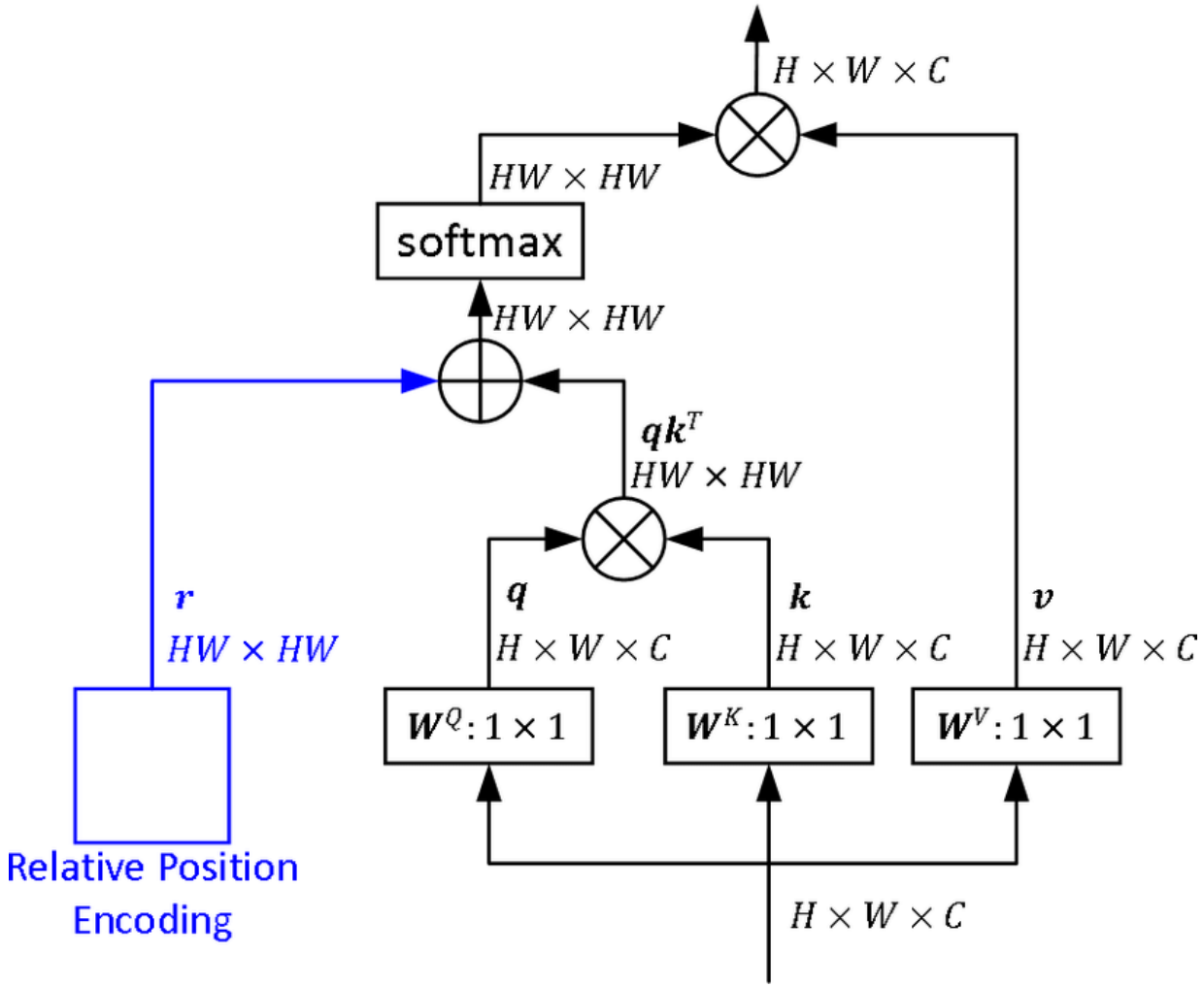
$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V + a_{ij}^V)$$

$$e_{ij} = \frac{x_i W^Q (x_j W^K + a_{ij}^K)^T}{\sqrt{d_z}}$$

Self-Attention with RPE

- 相对位置信息在 self-attention 计算时候丢失，最直接在计算 self-attention 时候再加回来。

- 计算 attention score 和 weighted value 时各加入可训练表示相对位置的参数，multi head之间可以共享。



03

旋转位置编码

RoPE

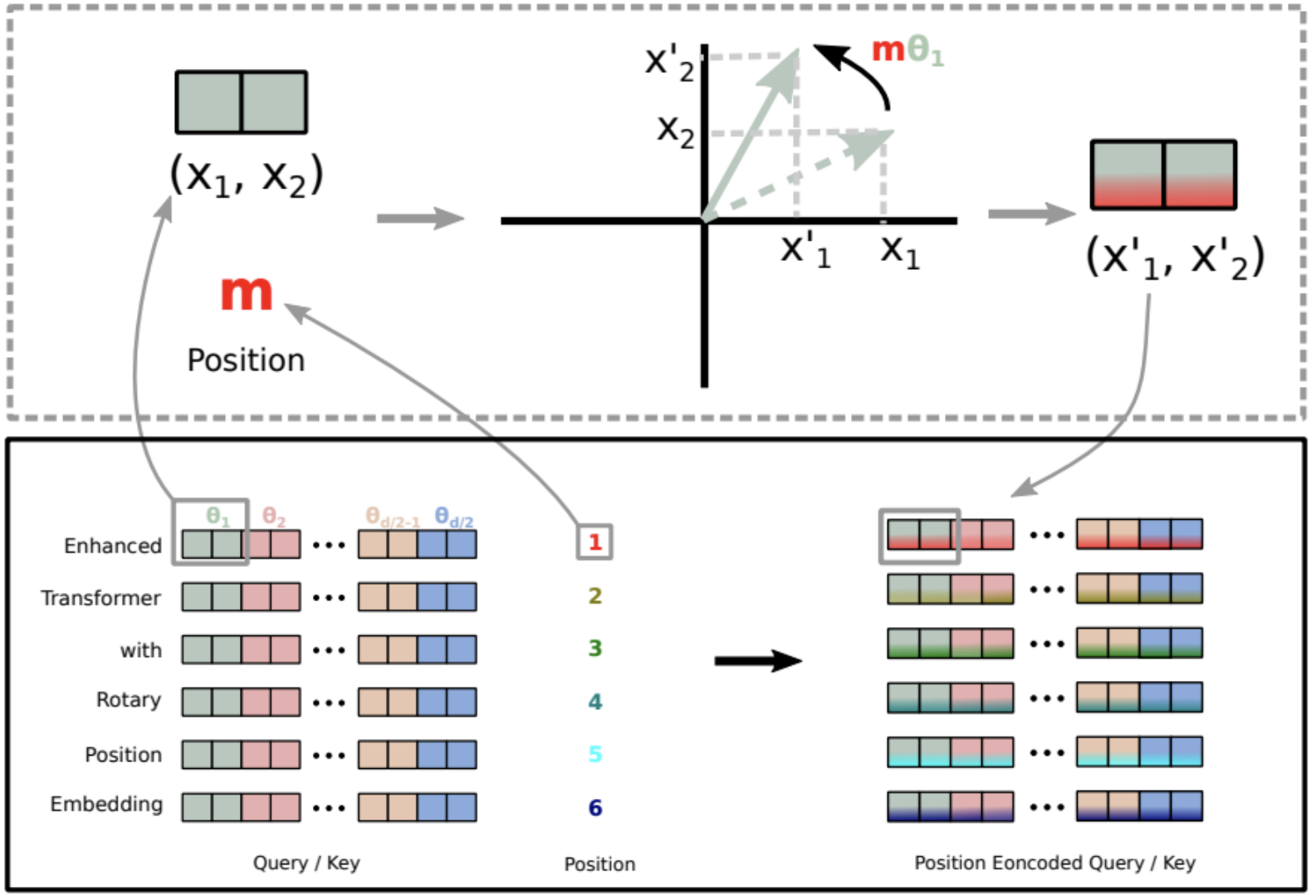
旋转位置编码

Roformer: Enhanced Transformer With Rotray Position Embedding

- **原理：**
  - 将位置信息编码为旋转矩阵，作用于查询（Query）和键（Key）的注意力计算中，隐式融合绝对位置与相对位置的双重特性。RoPE 位置编码通过将一个向量旋转某个角度，为其赋予位置信息。
- **优势：**
  - 外推性强：支持超长序列推理（如LLaMA-2支持4k→16k扩展）
  - 远程衰减：通过波长设计自动衰减远距离依赖
  - 数学等价：通过旋转操作实现相对位置偏移的数学等价性。
- **适用：**
  - 长文本生成（如Qwen系列模型），需处理超长上下文（如32K token以上）。
  - 多模态任务（如图文对齐），需灵活适配不同模态的序列长度。

旋转位置编码

Roformer: Enhanced Transformer With Rotray Position Embedding



旋转位置编码 RoPE

- 通过绝对位置编码的方式实现相对位置编码。回顾我们此前定义的位置编码函数，该函数表示对词向量 q 添加绝对位置信息 m，得到 qm：

$$q_m = f(q, m)$$

- RoPE 希望 qm 与 kn 之间的点积，即 $f(q,m) \cdot f(k,n)$ 中能够带有相对位置信息 $m - n$。

旋转位置编码 RoPE

- 那么 $f(q,m)\cdot f(k,n)$ 如何才算带有相对位置信息呢?
- 需要能够将 $f(q,m)\cdot f(k,n)$ 表示成一个关于 q 、 k 、 $m-n$ 的函数 $g(q,k,m-n)$

- 其中 $m-n$ 便表示着两个向量之间的相对位置信息:

$$f(q,m)\cdot f(k,n)=g(q,k,m-n)$$

二维位置编码

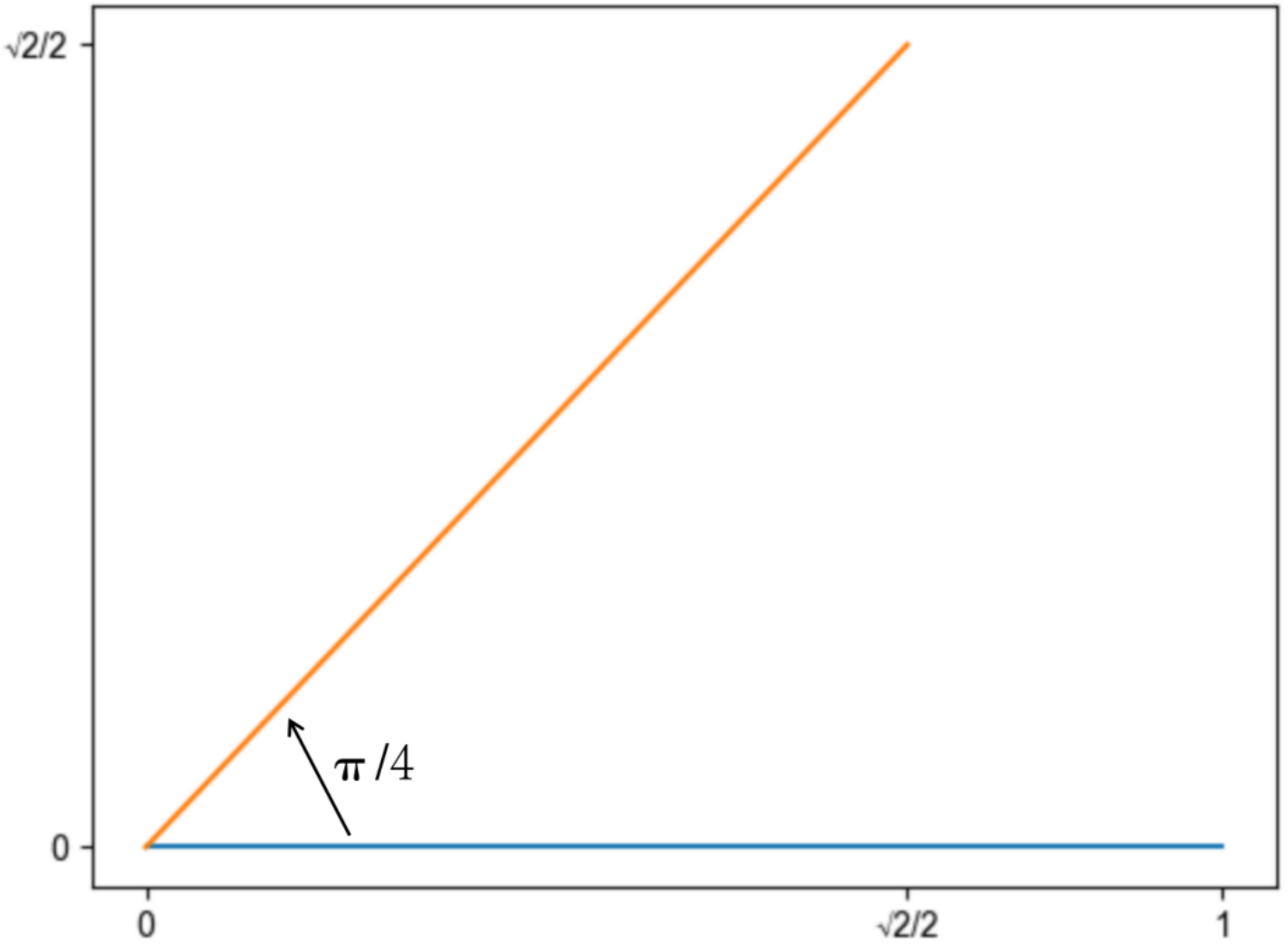
- 假设词向量是二维，作者得到如下位置编码函数，其中 m 为位置下标， θ 为常数：

$$f(q,m) = R_m q = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \end{pmatrix}$$

- 以二维向量 $(1,0)$ 为例，将其逆时针旋转45度，弧度为 $\pi/4$ ，将得到新的二维向量，向量的模长未发生改变，仍然是 1。计算过程如下：

$$\begin{pmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} \\ \sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\frac{\pi}{4} \\ \sin\frac{\pi}{4} \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix}$$

二维位置编码



二维位置编码

- 得到的是一个向量旋转的函数，左侧的 Rm 是一个旋转矩阵， $f(q,m)$ 表示在保持向量 q 的模长的同时，将其逆时针旋转 $m\theta$ 。

$$f(q,m) = R_m q = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \end{pmatrix}$$

- 这意味着只需要将向量旋转某个角度，即可实现对该向量添加绝对位置信息，这就是旋转位置编码的由来。

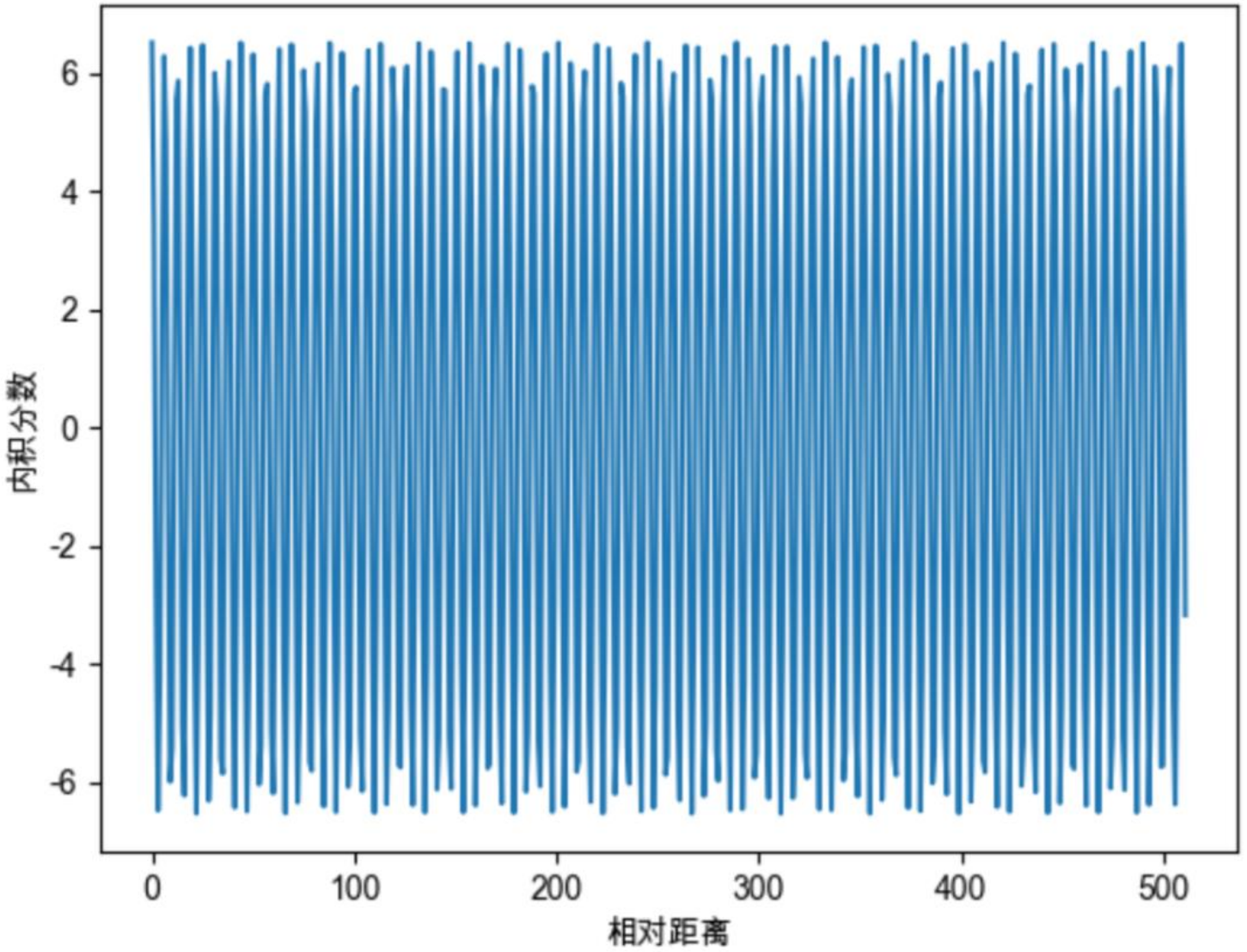
推广到多维

- 把高维向量，两两一组，分别旋转。最终高维向量的旋转可表示成如下公式，可以认为左侧便是高维向量的旋转矩阵：

$$\begin{pmatrix} \cos m\theta & -\sin m\theta & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta & \cos m\theta & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta & -\sin m\theta & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta & \cos m\theta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta & -\sin m\theta \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix}$$

缺乏远程衰减性

- 随机初始化两个向量 q 和 k，将 q 固定在位置 0 上，k 的位置从 0 开始逐步变大，依次计算 q 和 k 之间的内积。
- 随着 q 和 k 相对距离增加，之间内积分数呈现出一定震荡特性，缺乏重要远程衰减性，这并不是我们希望的。



缺乏远程衰减性

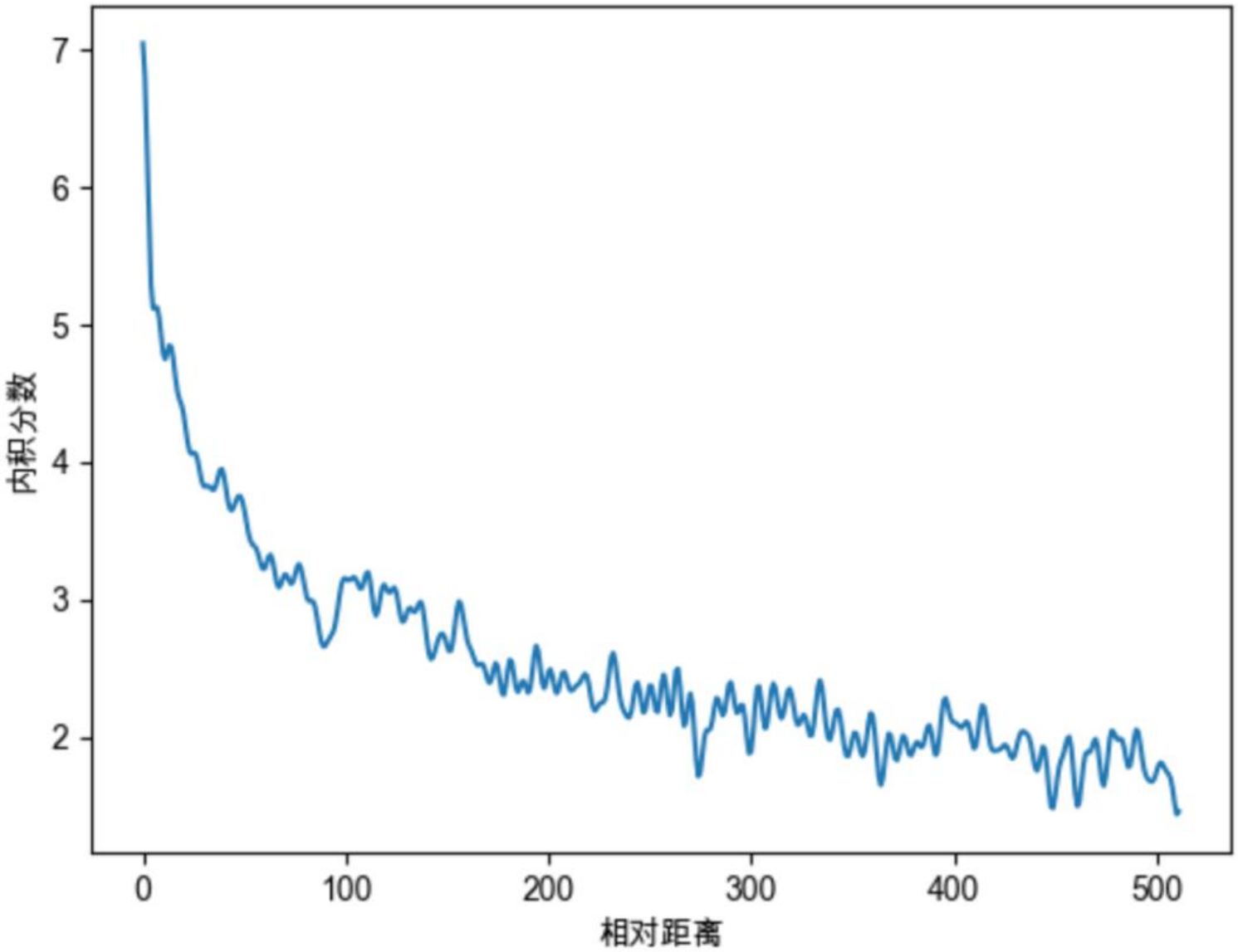
- 借鉴Sinusoidal位置编码，将每个分组的 θ 设为不同的常量，从而引入远程衰减的性质。沿用 Sinusoidal 位置编码设置， $\theta i = 10000 - 2i/d$ 。则可以将高维向量的旋转矩阵更新为如下：

$$\begin{pmatrix} \cos m\theta_0 & -\sin m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_0 & \cos m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1} \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix}$$

缺乏远程衰减性

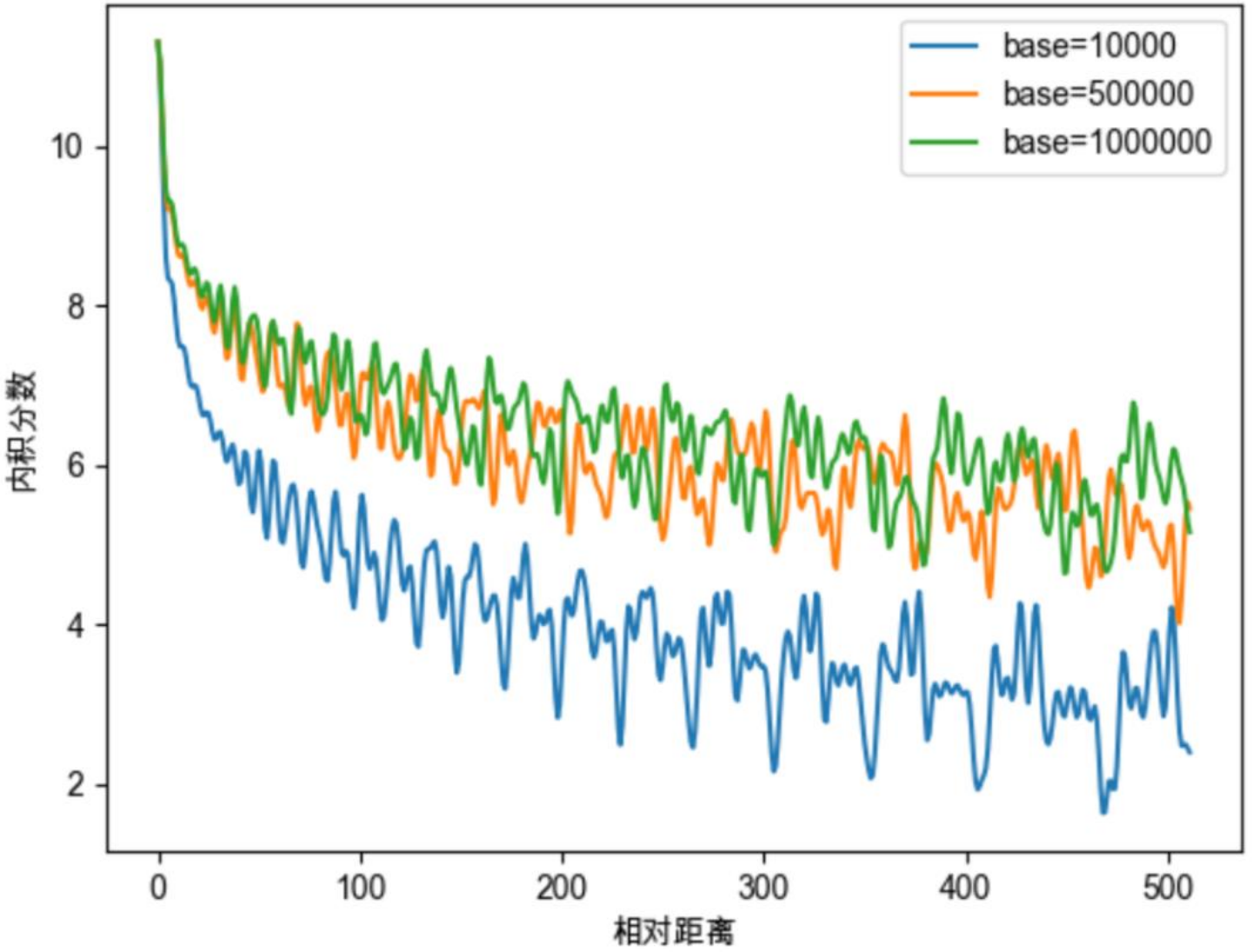
- 继续随机初始化两个向量 q 和 k，将 q 固定在位置 0 上，k 的位置从 0 开始逐步变大，依次计算 q 和 k 之间的内积。

- 发现随着 q 和 k 的相对距离的增加，它们之间的内积分数呈现出远程衰减的性质，



旋转位置编码 RoPE

- 目前大多长度外推工作都是通过放大base以提升模型的输入长度。
  - 例如 Code LLaMA 将 base 设为 1000000，LLaMA2 Long 设为 500000。
- 但更大的 base 也将会使得注意力远程衰减的性质变弱，改变模型的注意力分布，导致模型的输出质量下降。



总结与思考

- 外推性需求驱动创新：RoPE 因支持长度外推成为大模型主流选择。
- 任务导向设计：短文本任务仍可用绝对编码，长文本需相对或 RoPE。
- 多模态位置编码，有什么区别？怎么实现？

Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

GitHub https://github.com/chenzomi12/AIInfra

引用与参考

- https://erdem.pl/2021/05/understanding-positional-encoding-in-transformers

- PPT 开源在：https://github.com/chenzomi12/AIInfra