RoPE（旋转位置编码）讲义

从问题、对象、约束到公式、引理、定理与工程实践

摘要

RoPE（Rotary Position Embedding）不是“把位置向量加到 token 上”这么简单，它的核心是：把单个 attention head 的通道按二维一组拆开，在每一组上做与位置成正比的旋转，从而让 query 和 key 的内积在代数上自动变成“内容项乘上相对位移相位”的形式。这样一来，位置依赖不是以绝对位置直接相加进入模型，而是以相位差的方式进入注意力打分。本讲义沿着“问题 - 对象 - 约束 - 形式定义 - 引理 - 定理 - 例子 - 工程”这条路线展开，并尽量把关键公式拆到二维 block、band、head 以及 attention logit 的原子项层面。

目录

1 问题 **2**

1.1 RoPE 要解决什么问题 2

1.2 没有它之前卡在哪里 2

1.3 从绝对位置到相对位置的需求 3

2 对象 **3**

2.1 位置 p 3

2.2 注意力中的 q, k 向量 4

2.3 内积与 token-pair 比较关系 4

2.4 RoPE 的基本操作流程 4

3 核心问题 **5**

3.1 如何让位置进入 attention 5

3.2 如何让打分依赖相对位移 5

3.3 如何兼顾近距离分辨率与远距离稳定性 6

3.4 如何在低参数与低复杂度下实现 6

4 自由度 **7**

4.1 频率表 θ_i 的选择 7

4.2 head 维度与 band 的划分 7

4.3 位置尺度与上下文缩放 7

4.4 RoPE 作用于 q, k 还是其他对象 8

4.5 本质变化与表面变化 8

5 约束 **9**

5.1 保持向量长度与几何结构 9

5.2 相对位移必须进入比较关系 9

5.3 计算复杂度约束 9

5.4 与标准 attention 流水线的兼容性 10

5.5 被排除的几类方案 10

6 坐标系 **10**

6.1 绝对位置与相对位置 10

6.2 加法型与几何变换型 11

6.3 单尺度与多尺度 11

6.4 显式解耦与混合耦合 11

6.5 RoPE 在位置编码坐标系中的位置 12

7 例子 **12**

7.1 单 band 的二维旋转 12

7.2 单个 head 的多 band RoPE 14

7.3 长上下文下的 scaled RoPE 17

7.4 近距离与远距离的数值例子 17

8 反例 **18**

8.1 直接加 learned absolute position embedding 18

8.2 只用单一频率的旋转 19

8.3 整体高维稠密大旋转 19

8.4 为什么这些方法不满足 RoPE 的关键目标 19

9 核心图景 20

9.1 一句话直觉 20

9.2 小表盘与相位差图景 20

9.3 多尺度位置感知的直观解释 20

10 形式定义 21

10.1 二维旋转矩阵 21

10.2 RoPE 在单个二维 block 上的定义 21

10.3 RoPE 在整个 head 上的 block-diagonal 形式 21

10.4 相对位移进入 attention 的公式 22

10.5 定义中每个条件的作用 22

11 基本操作 23

11.1 构造频率表 23

11.2 位置到相位的映射 23

11.3 head 的二维分解 24

11.4 在 attention logit 中度量距离效应 24

11.5 频率缩放、插值与长上下文扩展 25

12 关键引理 25

12.1 旋转的内积相减性质 25

12.2 旋转保持长度 25

12.3 多 band 带来多尺度表示 26

12.4 二维旋转为何是基本块 27

12.5 这些引理分别解决了什么障碍 27

13 核心定理（结构性结论） 28

13.1 相对位移表示结论 28

13.2 多尺度表示结论 28

13.3 二维分解结论 29

13.4 正交不变量结论 29

13.5 数学等价与工程可用性的区别 29

14 评价标准 **29**

14.1 什么算好 29

14.2 什么算坏 30

14.3 什么算深刻 30

14.4 什么算无效 30

14.5 RoPE 的工程评价维度 31

15 流派争论 **31**

15.1 RoPE 与其他位置编码机制的比较 31

15.2 原始 RoPE 与 scaled RoPE 31

15.3 显式解耦与频率混合 32

15.4 表达力、外推性与工程可控性的平衡 32

16 应用迁移 **32**

16.1 语言模型中的应用 32

16.2 代码模型中的应用 32

16.3 视觉与多模态中的迁移 32

16.4 时间序列中的迁移 33

16.5 RoPE 失效或变差的条件 33

17 训练闭环 **33**

17.1 能否预测 33

17.2 能否举例 33

17.3 能否反驳 34

17.4 能否证明 34

17.5 能否应用 34

17.6 能否教给别人 34

18 总结 **34**

18.1 RoPE 的一句话本质 34

18.2 RoPE 的数学核心 35

18.3 RoPE 的工程核心 35
18.4 后续可展开主题 35

A 附录：单 band 公式的一次性原子项展开 36

B 附录：原始 RoPE 与线性缩放 RoPE 的对照 36

1 问题

1.1 RoPE 要解决什么问题

在没有任何位置编码时，自注意力对输入 token 的排列几乎是“置换等变”的。如果把同一批 token 的顺序打乱，而内容向量本身不变，那么注意力层只看到一堆向量的相互相似度，却看不到“第几个词在前，第几个词在后”。于是模型无法区分“猫追狗”和“狗追猫”，也无法区分“定义在前、使用在后”还是“使用在前、定义在后”。

设第 p 个 token 的输入表示为 $\mathbf{x}_p$，线性投影后得到 query 与 key：

$$\mathbf{q}_p = W_Q \mathbf{x}_p, \qquad \mathbf{k}_m = W_K \mathbf{x}_m.$$

若不引入位置，注意力 logit 只有

$$\ell(p,m) = \frac{\mathbf{q}_p^\top \mathbf{k}_m}{\sqrt{d_h}},$$

它只依赖内容，不显式依赖位置索引 p, m。RoPE 的目标不是“随便塞一点位置信号进去”，而是让

位置进入 attention 的方式既有几何结构，又主要通过$(m-p)$ 进入。

1.2 没有它之前卡在哪里

在 Transformer 的早期设计里，最常见的位置方案有两类：

第一类是把绝对位置向量直接加到输入上，即

$$\mathbf{x}'_p = \mathbf{x}_p + \mathbf{e}_p.$$

然后再计算

$$\mathbf{q}_p = W_Q \mathbf{x}'_p, \qquad \mathbf{k}_m = W_K \mathbf{x}'_m.$$

这样位置当然进来了，但进入方式是“先加法混合，再投影”，模型最终学到的是绝对位置与内容的缠绕表示。它能工作，但相对位移 $m-p$ 并不是公式中天然出现的主角。

第二类是显式相对位置偏置，比如在 logit 里再加一个依赖 $m-p$ 的项：

$$\ell(p,m) = \frac{\mathbf{q}_p^\top \mathbf{k}_m}{\sqrt{d_h}} + b_{m-p}.$$

这种方法把“相对距离”直接加到分数上，思路是对的，但它更像在打分层面外加一个修正项，并没有把位移几何地嵌入 query-key 的比较本身。

RoPE 卡位的位置恰好在这里：我们希望 relative position 不是一个后加罚分项，也不是输入层里被动混进去的绝对索引，而是变成“比较两个 token 时，它们的相位差就是位置差”的内在机制。

1.3 从绝对位置到相对位置的需求

自然语言、代码、数学表达、时间序列，都更关心相对次序关系。例如“当前词往前看 3 个 token”比“当前词在第 517 个位置”更具有可迁移性。因此，理想位置编码应满足：

(1) 模型知道顺序；

(2) 注意力打分更容易依赖相对位移而不是绝对编号；

(3) 长度扩大时仍然能以同一套规则工作；

(4) 不要显著增加参数量和计算复杂度。

RoPE 的关键创新是：先对每个位置 p 施加旋转 $\mathbf{R}(p\theta_i)$，再让 query 和 key 做内积。由于旋转矩阵满足

$$\mathbf{R}(a)^{\mathsf{T}}\mathbf{R}(b) = \mathbf{R}(b-a),$$

于是位置依赖会自动折叠为差值 $b-a$。也就是说，绝对位置被转成了相对相位差。这正是 RoPE 名字里“rotary”的本质含义。

2 对象

2.1 位置 p

本讲义中，位置 p 指序列中的离散索引，通常取

$$p \in \{0, 1, 2, \ldots, L-1\}.$$

如果另一个 token 的位置记为 m，那么二者的相对位移记为

$$\Delta = m - p.$$

有时也有人使用 $\Delta = p - m$。两种记法都可以，但正弦项的符号会随之改变。本文统一采用

$$\Delta = m - p.$$

从工程上看，位置 p 是整数；从几何上看，它会被映射成每个 band 上的相位

$$\phi_i(p) = p\theta_i.$$

因此 RoPE 并不是直接处理整数索引，而是处理“位置乘频率”得到的角度。

2.2 注意力中的 q, k 向量

在单个 attention head 中，设 head 维度为 d_h，并要求 d_h 为偶数。令

$$d_h = 2m.$$

于是一个 d_h-维向量可以拆成 m 个二维块。对任意 $\mathbf{x} \in \mathbb{R}^{d_h}$，记其第 i 个二维 block 为

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_{2i} \\ x_{2i+1} \end{bmatrix}, \qquad i = 0, 1, \ldots, m-1.$$

单头里的 query 与 key 向量分别写成

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ \cdots \\ q_{d_h-1} \end{bmatrix}, \qquad \mathbf{k} = \begin{bmatrix} k_0 \\ k_1 \\ \cdots \\ k_{d_h-1} \end{bmatrix}.$$

对应的第 i 个 block 为

$$\mathbf{q}^{(i)} = \begin{bmatrix} q_{2i} \\ q_{2i+1} \end{bmatrix}, \qquad \mathbf{k}^{(i)} = \begin{bmatrix} k_{2i} \\ k_{2i+1} \end{bmatrix}.$$

RoPE 处理的对象不是整条序列一起做一个高维大旋转，而是每一对通道单独做一个二维旋转。这使得代数形式可解、几何解释直观、实现也只需要预先缓存 cos 和 sin。

2.3 内积与 token-pair 比较关系

注意力分数是 token-pair 之间的比较器。给定 query 位置 p 和 key 位置 m，单头 logit 为

$$\ell(p, m) = \frac{\tilde{\mathbf{q}}_p^{\mathsf{T}} \tilde{\mathbf{k}}_m}{\sqrt{d_h}},$$

其中 $\tilde{\mathbf{q}}_p, \tilde{\mathbf{k}}_m$ 是加入位置后的 query 与 key。

因此，RoPE 的核心问题并不是“怎么编码位置向量”，而是“如何改造 $\tilde{\mathbf{q}}_p^{\mathsf{T}} \tilde{\mathbf{k}}_m$ 这件事”，使得这个比较器对位移 $\Delta = m - p$ 有可控的敏感性。所有后续公式都围绕这一点展开。

2.4 RoPE 的基本操作流程

RoPE 在单头上的流程可以概括成四步：

(1) 选择一组频率 $\theta_0, \theta_1, \ldots, \theta_{m-1}$；

(2) 把位置 p 映射成各 band 的相位 $\phi_i(p) = p\theta_i$；

(3) 对 $\mathbf{q}_p, \mathbf{k}_p$ 的每个二维 block 施加旋转；

(4) 用旋转后的 $\tilde{\mathbf{q}}_p, \tilde{\mathbf{k}}_m$ 做内积，得到位置相关注意力分数。

最重要的一点是：RoPE 通常只作用在 q 和 k 上，不作用在 v 上。因为注意力权重由 q-k 比较决定，而 v 负责被加权汇聚的内容本身。如果对 v 也旋转，往往会改变内容表示，但不直接改善“相对位移进入打分”的结构。

表 1: 本文常用符号表

符号	含义
p, m	两个 token 的位置索引
$\Delta = m - p$	相对位移
d_h	单个 head 的维度，要求为偶数
$m = d_h/2$	二维 block（或 band）个数
θ_i	第 i 个 band 的角频率
$\phi_i(p) = p\theta_i$	位置 p 在第 i 个 band 上的相位
$\mathbf{R}(\phi)$	二维旋转矩阵
$\mathbf{q}^{(i)}, \mathbf{k}^{(i)}$	第 i 个二维 block 的 query 与 key
$\tilde{\mathbf{q}}_p, \tilde{\mathbf{k}}_m$	施加 RoPE 后的位置相关 query 与 key
$\ell(p, m)$	query 在位置 p 与 key 在位置 m 的单头注意力 logit

3 核心问题

3.1 如何让位置进入 attention

最直接的想法是把位置加到输入上。但 RoPE 采用的是更“靠近比较器”的做法：不是改输入的加法结构，而是改 query 和 key 的几何朝向。如果把第 i 个 block 看成二维平面中的一个向量，则位置 p 只做一件事：

$$\mathbf{q}_p^{(i)} \mapsto \mathbf{R}(p\theta_i)\mathbf{q}_p^{(i)}, \qquad \mathbf{k}_m^{(i)} \mapsto \mathbf{R}(m\theta_i)\mathbf{k}_m^{(i)}.$$

于是位置不是作为额外坐标加进去，而是作为旋转角度乘进去。

这个想法的优点是，位置直接进入了注意力最核心的双线性比较关系里。你比较两个 token，不是比较“原向量”，而是比较“各自转到对应位置后的向量”。

3.2 如何让打分依赖相对位移

RoPE 的真正关键在下面这个代数恒等式：

$$(\mathbf{R}(p\theta_i)\mathbf{q}^{(i)})^{\mathsf{T}}(\mathbf{R}(m\theta_i)\mathbf{k}^{(i)}) = (\mathbf{q}^{(i)})^{\mathsf{T}}\mathbf{R}((m-p)\theta_i)\mathbf{k}^{(i)}.$$

注意，右边只剩下 $(m-p)\theta_i$，绝对位置 p, m 消失了。因此位置依赖不再分别附着在 query 和 key 上，而是通过相对位移 $\Delta = m - p$ 进入。

更精确地说，RoPE 让“位置项的依赖形式”天然地变成 relative，而不是说整个 logit 只依赖相对位置。因为内容项 $\mathbf{q}^{(i)}, \mathbf{k}^{(i)}$ 仍然来自不同 token。准确表述应为：

RoPE 使位置进入 logit 的方式主要经由相对位移Δ.

3.3 如何兼顾近距离分辨率与远距离稳定性

如果只用一个频率 θ，那么 logit 的位置项只是一个单尺度振荡函数：

$$a\cos(\Delta\theta) + b\sin(\Delta\theta).$$

这会遇到经典矛盾：

- θ 大时，小位移很容易被区分，但相位绕得快，长距离会周期性混叠；
- θ 小时，长距离更稳定，但近邻分辨率不够。

RoPE 的策略是同时放很多频率。设第 i 个 band 的贡献是

$$c_i(\Delta) = a_i\cos(\Delta\theta_i) + b_i\sin(\Delta\theta_i).$$

其对位移的局部敏感性满足

$$\frac{\partial c_i}{\partial \Delta} = \theta_i\big(-a_i\sin(\Delta\theta_i) + b_i\cos(\Delta\theta_i)\big),$$

从而

$$\left|\frac{\partial c_i}{\partial \Delta}\right| \le \theta_i\sqrt{a_i^2 + b_i^2}.$$

所以大频率 band 提供高局部敏感度，小频率 band 提供长程平稳性，多 band 叠加就能形成“近处精细、远处平滑”的多尺度位置感知。

3.4 如何在低参数与低复杂度下实现

原始 RoPE 几乎不增加可训练参数。频率表通常是固定的，旋转所需只是 $\cos\phi_i(p)$ 和 $\sin\phi_i(p)$ 两个表。对每个二维 block 的计算也只有

$$\begin{bmatrix} x'_{2i} \\ x'_{2i+1} \end{bmatrix} = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \begin{bmatrix} x_{2i} \\ x_{2i+1} \end{bmatrix},$$

即

$$x'_{2i} = x_{2i}\cos\phi_i - x_{2i+1}\sin\phi_i, \qquad x'_{2i+1} = x_{2i}\sin\phi_i + x_{2i+1}\cos\phi_i.$$

所以整体开销与维度线性同阶，和常规 attention 管线非常兼容，这也是 RoPE 在大模型中被广泛采用的重要原因。

4 自由度

4.1 频率表 θ_i 的选择

RoPE 不是只有一种频率表。最常见的标准选择是

$$\theta_i = B^{-2i/d_h}, \qquad B = 10000, \qquad i = 0, 1, \ldots, m-1.$$

这意味着相邻 band 的频率按几何级数衰减。如果 $d_h = 8$，则 $m = 4$，频率近似为

$$\theta_0 = 1, \quad \theta_1 = 0.1, \quad \theta_2 = 0.01, \quad \theta_3 = 0.001.$$

这套表既简单，又天然覆盖快慢不同的时间尺度。

但工程上仍有自由度：可以改 B，可以让前几维更快、后几维更慢，也可以用分段或插值方式重新分配。频率表是“多尺度结构”的直接来源，因此是最重要的自由度之一。

4.2 head 维度与 band 的划分

如果单头维度是 d_h，则可旋转的二维块数量是

$$m = \frac{d_h}{2}.$$

这意味着 head 越大，可容纳的 band 越多，多尺度能力越强。例如：

$$d_h = 64 \Rightarrow m = 32, \qquad d_h = 128 \Rightarrow m = 64.$$

但并不是所有通道都必须旋转。一些实现只对前 d_{rot} 维做 RoPE，其中 d_{rot} 为偶数，剩余维度保持不变。这相当于在“纯内容通道”和“位置敏感通道”之间做折中。

但 band 的划分含义很容易被误解。它不是“某个 token 用高频、另一个 token 用低频”，而是单个 token 的 $\mathbf{q}, \mathbf{k}$ 表示在一个 head 内部被拆成很多二维 block，每个 block 绑定一个频率 θ_i。于是任意一对位置 p, m 做 attention 比较时，所有 band 都会同时参与：

$$\ell(p, m) = \frac{1}{\sqrt{d_h}} \sum_{i=0}^{m-1} s_i(p, m), \qquad s_i(p, m) = (\tilde{\mathbf{q}}_p^{(i)})^{\mathsf{T}} \tilde{\mathbf{k}}_m^{(i)}.$$

因此单个 token 需要携带的不是一个单尺度位置码，而是一整套“位置比较工具箱”：之后不管它和哪个 token 做内积，模型都能在所有 band 上同时测量相对位移。

4.3 位置尺度与上下文缩放

RoPE 的相位来自

$$\phi_i(p) = p\theta_i.$$

如果上下文长度从训练时的 L_{train} 扩到测试时的 L_{test}，相位可能增长过快，导致高频 band 快速绕圈。于是常见做法是把 p 换成某个缩放后的 $g(p)$，例如最简单的线性缩放

$$g(p) = \frac{p}{\alpha}, \qquad \alpha > 1,$$

对应

$$\phi_i(p) = g(p)\theta_i = \frac{p\theta_i}{\alpha}.$$

更一般地，也可以用分段函数、平滑函数或按 band 自适应的函数。本质上，scaled RoPE 不是单一公式，而是一族“修改相位映射 g”的方法。

4.4 RoPE 作用于 q, k 还是其他对象

标准 RoPE 作用于 query 和 key：

$$\tilde{\mathbf{q}}_p = \mathcal{R}_p \mathbf{q}_p, \qquad \tilde{\mathbf{k}}_m = \mathcal{R}_m \mathbf{k}_m.$$

通常不作用于 value。原因很直接：attention 权重由 q-k 相似度决定，而 value 是被权重聚合的内容。若对 v 也旋转，则会改变输出内容的坐标系，但不一定改善“相对位移进入 logit”的结构。

当然，也存在一些变体会对更丰富的对象施加旋转或相位变换。但只要目标是“让注意力打分带上相对位移”，作用于 q 和 k 是最干净的最小方案。

4.5 本质变化与表面变化

把握 RoPE 时，要区分“本质变化”和“表面变化”。

本质变化是：

(1) 把高维 head 显式拆成二维 block；

(2) 每个 block 做位置相关旋转；

(3) 利用旋转的群性质，让位置以差值形式进入内积。

表面变化是：

(1) 频率表选 10000 还是别的底数；

(2) 旋转全部维度还是部分维度；

(3) 相位映射是 p 还是 $g(p)$；

(4) 具体实现里先缓存 cos/sin 还是在线计算。

理解这一层区分后，就不会把“某种具体缩放实现”误当成 RoPE 本体。

5 约束

5.1 保持向量长度与几何结构

RoPE 之所以优雅，一个根本原因是它是正交变换。二维旋转矩阵 $\mathbf{R}(\phi)$ 满足

$$\mathbf{R}(\phi)^{\mathsf{T}}\mathbf{R}(\phi) = I_2,$$

因此对任意二维向量 $\mathbf{x} \in \mathbb{R}^2$，都有

$$\|\mathbf{R}(\phi)\mathbf{x}\|_2 = \|\mathbf{x}\|_2.$$

放到整个 head 上，RoPE 对每个二维 block 单独正交，所以整个向量的二范数也保持不变：

$$\|\mathcal{R}_p\mathbf{x}\|_2 = \|\mathbf{x}\|_2.$$

这保证了位置注入不会粗暴地放大或缩小向量长度，而只是改变各 block 的方向。

5.2 相对位移必须进入比较关系

如果一种位置编码方法只是在输入层混入绝对索引，但在打分公式里看不出相对位移，那么它就没有击中 RoPE 的关键目标。RoPE 的约束是：

$$\ell(p, m) \text{ 中的位置项应尽量通过 } (m - p) \text{ 出现。}$$

这是因为模型在泛化到新长度、新偏移时，relative displacement 往往比 absolute index 更稳定。

更严格地说，我们追求的是

$$(\mathcal{R}_p\mathbf{q})^{\mathsf{T}}(\mathcal{R}_m\mathbf{k}) = \mathbf{q}^{\mathsf{T}} f(m - p)\mathbf{k}$$

这样的结构。RoPE 选取 $f(\Delta) = \mathbf{R}(\Delta\theta_i)$ 达成了这一点。

5.3 计算复杂度约束

现代大模型最怕“看起来优雅、算起来昂贵”的方案。RoPE 的一个工程约束是：位置注入不能把注意力复杂度从 $O(L^2 d)$ 之外再抬高一个量级。RoPE 满足这一点，因为对每个 token、每个维度只做常数次乘加：

$$x'_{2i} = x_{2i}\cos\phi_i - x_{2i+1}\sin\phi_i, \qquad x'_{2i+1} = x_{2i}\sin\phi_i + x_{2i+1}\cos\phi_i.$$

这和线性层、归一化一样，都是逐 token 逐通道的局部操作。

5.4 与标准 attention 流水线的兼容性

标准 attention 流水线是

$$\mathbf{x}_p \xrightarrow{W_Q,W_K,W_V} \mathbf{q}_p, \mathbf{k}_p, \mathbf{v}_p \xrightarrow{\text{score}} \text{softmax} \xrightarrow{\text{weighted sum}} \text{output}.$$

RoPE 只是在 $\mathbf{q}_p, \mathbf{k}_p$ 与 score 之间加一个旋转层：

$$\mathbf{q}_p \mapsto \tilde{\mathbf{q}}_p, \qquad \mathbf{k}_p \mapsto \tilde{\mathbf{k}}_p.$$

因此它天然兼容 KV cache、张量并行、混合精度和已有推理框架。这比某些必须重写 attention kernel 的位置方案更容易落地。

5.5 被排除的几类方案

从约束反过来，可以看出哪些方案不适合做 RoPE 式位置编码：

(1) 会显著改变向量范数的非正交映射；

(2) 不能把位置差折叠进比较关系的纯绝对加法方案；

(3) 需要随序列长度线性增长的大量可训练位置参数；

(4) 需要对整个高维向量做复杂稠密矩阵乘法、难以缓存或并行的方案；

(5) 频率过于单一、无法覆盖多尺度需求的方案。

被排除并不等于“完全不能用”，而是说它们不满足 RoPE 试图同时达成的那组目标。

6 坐标系

6.1 绝对位置与相对位置

位置编码方法首先可以按“它更强调绝对还是相对”来划分。

绝对位置方法关心“这个 token 在第几位”，例如

$$\mathbf{x}'_p = \mathbf{x}_p + \mathbf{e}_p.$$

相对位置方法关心“两个 token 相差几步”，例如

$$\ell(p, m) = \frac{\mathbf{q}_p^\top \mathbf{k}_m}{\sqrt{d_h}} + b_{m-p}.$$

RoPE 介于二者之间：它在构造时使用绝对索引 p, m，但在比较时通过

$$\mathbf{R}(p\theta)^\top \mathbf{R}(m\theta) = \mathbf{R}((m-p)\theta)$$

把位置依赖转成相对位移。

6.2 加法型与几何变换型

第二个坐标轴是“加法型”还是“几何变换型”。

加法型方案把位置当作额外向量加到输入里：

$$\mathbf{x}'_p = \mathbf{x}_p + \mathbf{e}_p.$$

几何变换型方案则不增加坐标维度，而是改变原向量的方向、尺度或子空间位置。RoPE 属于典型几何变换型，因为它做的是旋转：

$$\mathbf{x}^{(i)} \mapsto \mathbf{R}(\phi_i(p))\mathbf{x}^{(i)}.$$

几何变换型的优势是：可以直接设计“比较器在变换前后如何变化”。

6.3 单尺度与多尺度

若所有位置信号只对应一个尺度，那么模型只能在一种“快慢节奏”上分辨位移。RoPE 用

$$\theta_0, \theta_1, \ldots, \theta_{m-1}$$

构成多尺度。快 band 负责近邻敏感，慢 band 负责长程稳定。这和只用一个 θ 的单尺度旋转形成鲜明对比。

一个很直观的比喻是“短尺子”和“长尺子”。高频 band 像短尺子：对 $\Delta = 1, 2, 3$ 这类小距离变化很快，因而分辨率高；低频 band 像长尺子：在大距离上变化更平缓，因而更适合作为长程稳定坐标。如果只有一个整体频率，模型就只有一把尺子，不可能同时把近处的小差别分得很细，又让远处坐标保持平滑。

需要强调的是，多尺度不是把不同 token 分配给不同尺子，而是同一对 token 的相对距离

$$\Delta = m - p$$

会被所有 band 同时测量，最后再在同一个 head 的 logit 里耦合成总响应。因此不同 head 的差别，不是“使用哪一个 band”，而是“更依赖哪些 band 的组合”。

6.4 显式解耦与混合耦合

有些位置方案会把所有通道混在一起学一个复杂映射，很难看出每个维度在做什么。RoPE 则是显式解耦：每两个维度一组，每组一个明确频率。因此整体结构可写成 block-diagonal：

$$\mathcal{R}_p = \mathrm{blkdiag}\big(\mathbf{R}(p\theta_0), \mathbf{R}(p\theta_1), \ldots, \mathbf{R}(p\theta_{m-1})\big).$$

显式解耦带来可解释性和可控性。工程上你能直接问：哪几个 band 太快、哪几个 band 太慢、要不要缩放 $g(p)$。

6.5 RoPE 在位置编码坐标系中的位置

把以上几条坐标轴合起来，RoPE 的位置可以概括为：

表 2: RoPE 在位置编码坐标系中的定位

维度	RoPE 的位置
绝对/相对	构造上用绝对位置，比较上呈现相对位移
加法/几何	几何变换型（二维旋转）
单尺度/多尺度	多尺度（多 band 频率表）
显式/混合	显式 block 解耦
参数量	原始版本几乎零额外可训练参数
复杂度	与标准 attention 高度兼容，逐维常数开销

因此，RoPE 不是简单的 sinusoidal embedding 变体，而是把“位置”从输入层符号变成了“比较器里的相位差”。这就是它在位置编码坐标系中的独特之处。

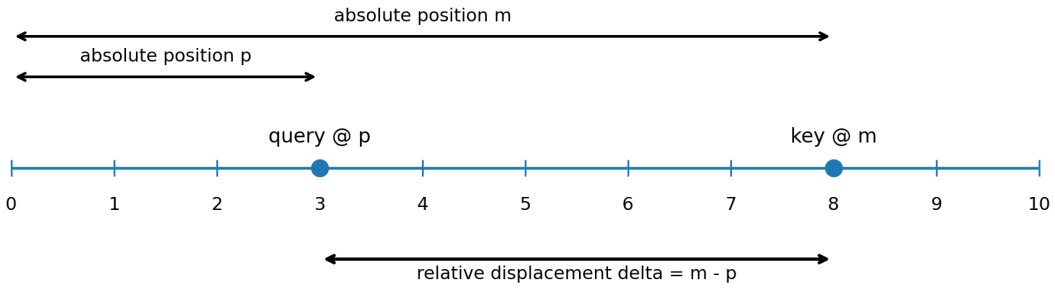

图 1: 绝对位置与相对位移示意。RoPE 在构造时知道 p 和 m，但在打分结构里更强调 $\Delta = m - p$。

7 例子

7.1 单 band 的二维旋转

先看最小非平凡例子：一个二维 block。设

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \end{bmatrix}, \qquad \mathbf{k} = \begin{bmatrix} k_0 \\ k_1 \end{bmatrix}, \qquad \mathbf{R}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

若 query 在位置 p，key 在位置 m，频率为 θ，则

$$\tilde{\mathbf{q}}_p = \mathbf{R}(p\theta)\mathbf{q}, \qquad \tilde{\mathbf{k}}_m = \mathbf{R}(m\theta)\mathbf{k}.$$

把原子项写开：

$$\tilde{\mathbf{q}}_p = \begin{bmatrix} q_0\cos(p\theta) - q_1\sin(p\theta) \\ q_0\sin(p\theta) + q_1\cos(p\theta) \end{bmatrix},$$

$$\tilde{\mathbf{k}}_m = \begin{bmatrix} k_0\cos(m\theta) - k_1\sin(m\theta) \\ k_0\sin(m\theta) + k_1\cos(m\theta) \end{bmatrix}.$$

于是单 band 分数

$$s(p,m) = \tilde{\mathbf{q}}_p^{\mathsf{T}}\tilde{\mathbf{k}}_m$$

可以完全展开为

$$\begin{aligned} s(p,m) = {} & \big(q_0\cos(p\theta) - q_1\sin(p\theta)\big)\big(k_0\cos(m\theta) - k_1\sin(m\theta)\big) \\ & + \big(q_0\sin(p\theta) + q_1\cos(p\theta)\big)\big(k_0\sin(m\theta) + k_1\cos(m\theta)\big). \end{aligned}$$

继续逐项整理：

$$\begin{aligned} s(p,m) = {} & q_0k_0\cos(p\theta)\cos(m\theta) - q_0k_1\cos(p\theta)\sin(m\theta) \\ & - q_1k_0\sin(p\theta)\cos(m\theta) + q_1k_1\sin(p\theta)\sin(m\theta) \\ & + q_0k_0\sin(p\theta)\sin(m\theta) + q_0k_1\sin(p\theta)\cos(m\theta) \\ & + q_1k_0\cos(p\theta)\sin(m\theta) + q_1k_1\cos(p\theta)\cos(m\theta). \end{aligned}$$

把同类项合并，利用

$$\cos a\cos b + \sin a\sin b = \cos(b-a),$$

$$\sin a\cos b - \cos a\sin b = \sin(a-b) = -\sin(b-a),$$

得到

$$s(p,m) = (q_0k_0 + q_1k_1)\cos((m-p)\theta) + (q_1k_0 - q_0k_1)\sin((m-p)\theta).$$

这已经把“原子项”压缩为两部分：

(1) 同向内积项 $q_0k_0 + q_1k_1$；

(2) 旋转耦合项 $q_1k_0 - q_0k_1$。

位置只剩 $(m-p)\theta$。

下面给一个数值例子。取

$$\mathbf{q} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad \mathbf{k} = \begin{bmatrix} 1.5 \\ -0.5 \end{bmatrix}, \quad \theta = 0.2, \quad p = 3, \quad m = 8.$$

则

$$p\theta = 0.6, \qquad m\theta = 1.6, \qquad (m-p)\theta = 1.0.$$

先转 query：

$$\tilde{\mathbf{q}}_p = \begin{bmatrix} 2\cos 0.6 - 1\sin 0.6 \\ 2\sin 0.6 + 1\cos 0.6 \end{bmatrix} \approx \begin{bmatrix} 2\times 0.8253 - 1\times 0.5646 \\ 2\times 0.5646 + 1\times 0.8253 \end{bmatrix} = \begin{bmatrix} 1.0860 \\ 1.9546 \end{bmatrix}.$$

再转 key：

$$\tilde{\mathbf{k}}_m = \begin{bmatrix} 1.5\cos 1.6 - (-0.5)\sin 1.6 \\ 1.5\sin 1.6 + (-0.5)\cos 1.6 \end{bmatrix} \approx \begin{bmatrix} 1.5\times(-0.0292) + 0.5\times 0.9996 \\ 1.5\times 0.9996 - 0.5\times(-0.0292) \end{bmatrix} = \begin{bmatrix} 0.4560 \\ 1.5140 \end{bmatrix}.$$

于是

$$s(p,m) = 1.0860\times 0.4560 + 1.9546\times 1.5140 \approx 3.4544.$$

若直接用相对位移公式，则

$$q_0k_0 + q_1k_1 = 2\times 1.5 + 1\times(-0.5) = 2.5,$$

$$q_1k_0 - q_0k_1 = 1\times 1.5 - 2\times(-0.5) = 2.5,$$

因此

$$s(p,m) = 2.5\cos 1.0 + 2.5\sin 1.0 \approx 2.5\times 0.5403 + 2.5\times 0.8415 \approx 3.4544,$$

与直接旋转后点积完全一致。

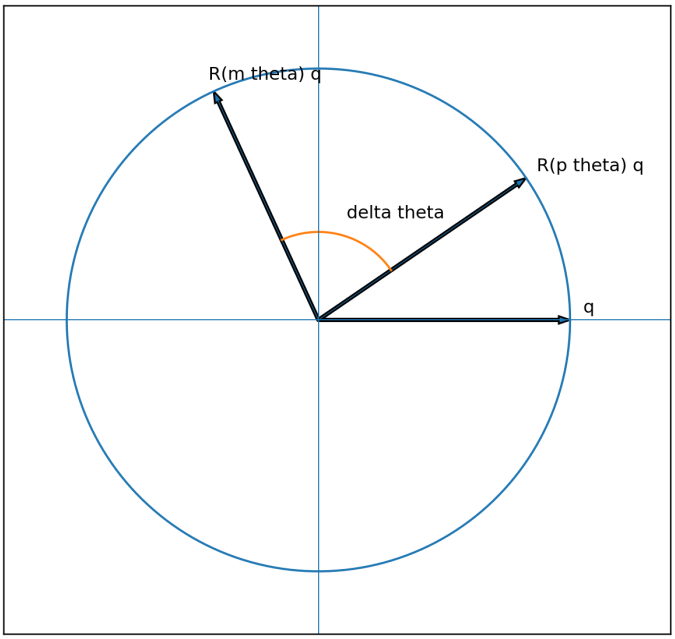

图 2: 单 band 的二维旋转图景。位置 p 和 m 分别对应两个旋转角，而比较只看相位差。

7.2 单个 head 的多 band RoPE

令单头维度 $d_h = 8$，则有 $m = 4$ 个二维 band。取标准频率表

$$\theta_0 = 1, \quad \theta_1 = 0.1, \quad \theta_2 = 0.01, \quad \theta_3 = 0.001.$$

取

$$\mathbf{q}=\begin{bmatrix}1\\2\\0\\1\\2\\0\\1\\-1\end{bmatrix},\qquad \mathbf{k}=\begin{bmatrix}2\\1\\1\\0\\0\\1\\-1\\2\end{bmatrix},\qquad p=2,\ m=5.$$

则相对位移为

$$\Delta=m-p=3.$$

第 0 个 band：

$$\mathbf{q}^{(0)}=\begin{bmatrix}1\\2\end{bmatrix},\qquad \mathbf{k}^{(0)}=\begin{bmatrix}2\\1\end{bmatrix}.$$

内容系数

$$a_0=q_0k_0+q_1k_1=1\times2+2\times1=4,$$

$$b_0=q_1k_0-q_0k_1=2\times2-1\times1=3.$$

位置角差

$$\Delta\theta_0=3\times1=3.$$

所以

$$s_0=4\cos3+3\sin3\approx-3.5366.$$

第 1 个 band：

$$\mathbf{q}^{(1)}=\begin{bmatrix}0\\1\end{bmatrix},\qquad \mathbf{k}^{(1)}=\begin{bmatrix}1\\0\end{bmatrix}.$$

于是

$$a_1=0\times1+1\times0=0,\qquad b_1=1\times1-0\times0=1,\qquad \Delta\theta_1=3\times0.1=0.3,$$

$$s_1=0\cdot\cos0.3+1\cdot\sin0.3\approx0.2955.$$

第 2 个 band：

$$\mathbf{q}^{(2)}=\begin{bmatrix}2\\0\end{bmatrix},\qquad \mathbf{k}^{(2)}=\begin{bmatrix}0\\1\end{bmatrix}.$$

于是

$$a_2=2\times0+0\times1=0,\qquad b_2=0\times0-2\times1=-2,\qquad \Delta\theta_2=3\times0.01=0.03,$$

$$s_2=0\cdot\cos0.03+(-2)\sin0.03\approx-0.0600.$$

第 3 个 band：

$$\mathbf{q}^{(3)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad \mathbf{k}^{(3)} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}.$$

于是

$$a_3 = 1 \times (-1) + (-1) \times 2 = -3,$$

$$b_3 = (-1) \times (-1) - 1 \times 2 = -1,$$

$$\Delta\theta_3 = 3 \times 0.001 = 0.003,$$

$$s_3 = -3\cos 0.003 - \sin 0.003 \approx -3.0030.$$

整个 head 的未归一化分数为

$$s_{\text{head}} = s_0 + s_1 + s_2 + s_3 \approx -6.3041.$$

除以 $\sqrt{8}$ 后得到标准单头 logit。这个例子很直观地显示出：

- 高频 band（如 $\theta_0 = 1$）随位移变化很快；
- 低频 band（如 $\theta_3 = 0.001$）几乎只发生缓慢变化；
- 多 band 叠加得到多尺度相对位置表示。

还要特别说明一个常见误解：不是 band 0 服务第一个 token、band 1 服务第二个 token，而是同一对位置 (p, m) 会在四个 band 上同时比较，然后加总成一个单头 logit：

$$\ell_{\text{head}}(\Delta) = \frac{1}{\sqrt{d_h}} \sum_{i=0}^{m-1} s_i(\Delta).$$

因此“高频更局部、低频更稳定”的意思不是它们各自独立输出不同结论，而是它们共同塑造这个 head 的距离响应函数。如果某个 head 学到的有效权重更偏向高频 band，则 $\ell_{\text{head}}(\Delta)$ 在小 Δ 区域变化更快；如果更偏向低频 band，则同一个总分在大距离上会更平缓、更稳定。

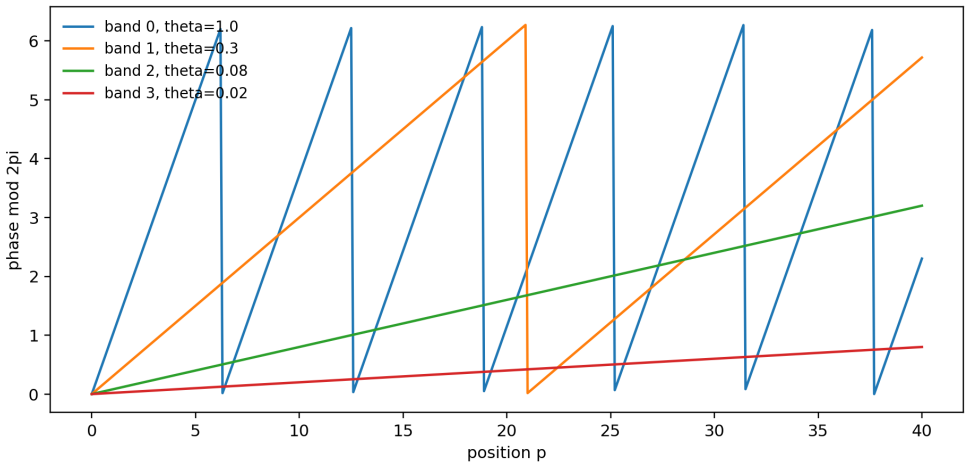

图 3: 多 band 相位随位置变化示意。高频 band 很快绕圈，低频 band 缓慢变化。

7.3 长上下文下的 scaled RoPE

长上下文时，原始相位 $\phi_i(p) = p\theta_i$ 可能过大。例如仍取 $d_h = 8$、$\theta_i = \{1, 0.1, 0.01, 0.001\}$，若相对位移是

$$\Delta p = 4096,$$

则各 band 相位差为

$$\Delta p\,\theta_0 = 4096, \qquad \Delta p\,\theta_1 = 409.6, \qquad \Delta p\,\theta_2 = 40.96, \qquad \Delta p\,\theta_3 = 4.096.$$

前两个 band 已经绕很多圈，容易出现强周期混叠。

若用最简单的线性缩放 $g(p) = p/8$，则相位差变成

$$\frac{\Delta p}{8}\theta_0 = 512, \qquad \frac{\Delta p}{8}\theta_1 = 51.2, \qquad \frac{\Delta p}{8}\theta_2 = 5.12, \qquad \frac{\Delta p}{8}\theta_3 = 0.512.$$

可以看到，缩放并没有消除周期性，但明显减慢了相位增长速度，尤其对慢 band 更有利。

这也是 scaled RoPE 的核心思想：

$$\phi_i(p) = g(p)\theta_i$$

中的 g 不再等于原始 p，而是某种压缩后的坐标。不同变体的差别，基本都体现在 g 的设计上。

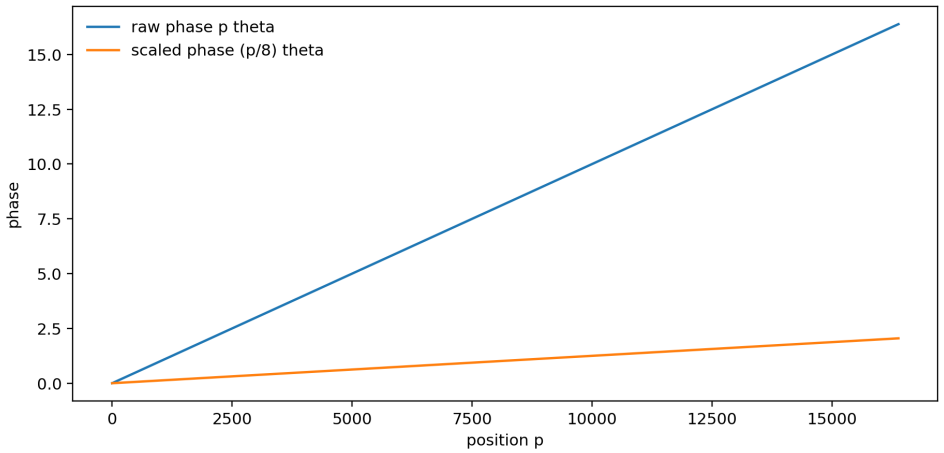

图 4: 长上下文下相位压缩示意。例子中用线性缩放 $p \mapsto p/8$ 说明 scaled RoPE 的基本想法。

7.4 近距离与远距离的数值例子

为了说明“近距离分辨率”和“远距离稳定性”的矛盾，先看单 band 函数

$$f(\Delta) = \cos(\Delta\theta).$$

若 $\theta = 1$（高频 band），则

$$f(0) = \cos 0 = 1, \qquad f(1) = \cos 1 \approx 0.5403, \qquad f(2) = \cos 2 \approx -0.4161.$$

只差一个 token，数值就明显改变，说明近邻分辨率高。但该 band 的周期长度是

$$T = \frac{2\pi}{\theta} = 2\pi \approx 6.283.$$

所以 $\Delta = 0$ 和 $\Delta \approx 6$ 的相位已经非常接近，混叠来得很快。

若 $\theta = 0.01$（低频 band），则

$$f(0) = 1, \qquad f(1) = \cos 0.01 \approx 0.99995, \qquad f(2) = \cos 0.02 \approx 0.99980.$$

近邻几乎分不开，但周期长度变成

$$T = \frac{2\pi}{0.01} \approx 628.3,$$

长距离更稳定。例如

$$f(628) = \cos(6.28) \approx 0.999995.$$

所以单 band 无法同时满足两边需求，多 band 恰好在这里提供折中。

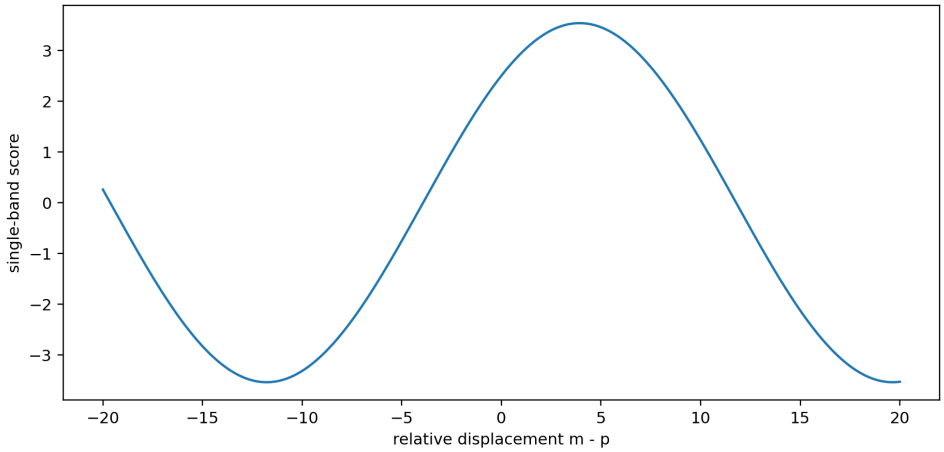

图 5: 单 band 分数随相对位移变化的示意。RoPE 的多 band 版本就是把很多这样的不同频率曲线叠加起来。

8 反例

8.1 直接加 learned absolute position embedding

设 learned absolute position embedding 为 $\mathbf{e}_p$，输入改写成

$$\mathbf{x}'_p = \mathbf{x}_p + \mathbf{e}_p.$$

则

$$\mathbf{q}_p = W_Q(\mathbf{x}_p + \mathbf{e}_p) = W_Q\mathbf{x}_p + W_Q\mathbf{e}_p,$$

$$\mathbf{k}_m = W_K(\mathbf{x}_m + \mathbf{e}_m) = W_K\mathbf{x}_m + W_K\mathbf{e}_m.$$

注意力 logit 展开为

$$\begin{aligned}\ell(p,m) &= \frac{1}{\sqrt{d_h}}\big(W_Q\mathbf{x}_p + W_Q\mathbf{e}_p\big)^{\mathsf{T}}\big(W_K\mathbf{x}_m + W_K\mathbf{e}_m\big) \\ &= \frac{1}{\sqrt{d_h}}\Big[(W_Q\mathbf{x}_p)^{\mathsf{T}}(W_K\mathbf{x}_m) + (W_Q\mathbf{x}_p)^{\mathsf{T}}(W_K\mathbf{e}_m) \\ &\qquad + (W_Q\mathbf{e}_p)^{\mathsf{T}}(W_K\mathbf{x}_m) + (W_Q\mathbf{e}_p)^{\mathsf{T}}(W_K\mathbf{e}_m)\Big].\end{aligned}$$

这里位置进入了，但并没有自然收缩成 $m-p$。模型当然可以通过训练“近似学会”某些相对关系，但公式本身不是 relative-first 的。

8.2 只用单一频率的旋转

若整个 head 只有一个频率 θ，则打分位置项只能写成

$$\ell(\Delta) = a\cos(\Delta\theta) + b\sin(\Delta\theta).$$

这种表示太单薄：它只有一个周期长度

$$T = \frac{2\pi}{\theta},$$

所以要么在近处分辨率高、远处很快绕圈，要么在远处稳定、近处太平。多尺度问题被压扁成单尺度问题，因此不能达到 RoPE 的关键目标。

8.3 整体高维稠密大旋转

设某人想在整个 d_h-维空间里用一个稠密矩阵 $U_p \in O(d_h)$ 做位置变换：

$$\tilde{\mathbf{q}}_p = U_p\mathbf{q}, \qquad \tilde{\mathbf{k}}_m = U_m\mathbf{k}.$$

则

$$\tilde{\mathbf{q}}_p^{\mathsf{T}}\tilde{\mathbf{k}}_m = \mathbf{q}^{\mathsf{T}}U_p^{\mathsf{T}}U_m\mathbf{k}.$$

若想让位置只通过相对位移进入，就必须要求

$$U_p^{\mathsf{T}}U_m = U_{m-p}$$

或某种等价的群结构。对任意稠密 learned matrix 而言，这个约束很强，训练和实现都复杂。更糟糕的是，稠密混合打破了每个二维 band 的显式可解释性，也不再方便预计算 $\cos/\sin$。

8.4 为什么这些方法不满足 RoPE 的关键目标

总结起来，上面三类反例的问题分别是：

(1) learned absolute embedding 不能在公式层面把位置化为相对位移；

(2) 单频旋转没有多尺度；

(3) 稠密大旋转虽然可能很强，但破坏了 RoPE 的简洁、可解和低复杂度。

RoPE 之所以恰到好处，在于它既不是“太弱”，也不是“太重”。它用二维旋转这块最小几何积木，以极低代价把 relative phase 写进了 attention 比较器里。

9 核心图景

9.1 一句话直觉

一句话直觉是：

RoPE = 给每一对通道挂上一块转速不同的小表盘，位置就是转过的角度。

两个 token 做注意力比较时，真正起作用的不是它们各自表盘的绝对指针位置，而是两个指针之间的夹角，也就是相位差。

9.2 小表盘与相位差图景

把第 i 个二维 block 想成一个小圆盘。位置 p 把这个圆盘转到角度 $p\theta_i$，位置 m 把另一个同频圆盘转到角度 $m\theta_i$。二者比较时，只看二者差角

$$(m-p)\theta_i.$$

这就是“绝对位置转成相对位移”的几何图景。

在这个图景里，二维是关键。因为二维空间恰好能容纳一个连续角度参数，而且旋转不改变长度。所以 RoPE 不是任意魔法，而是最朴素的平面旋转几何。

9.3 多尺度位置感知的直观解释

若所有小表盘都以同一速度转动，模型只能看到一种尺度的位移。RoPE 的高明之处是给不同二维 block 不同转速：

$$\theta_0 > \theta_1 > \cdots > \theta_{m-1}.$$

快表盘对小位移更敏感；慢表盘在长距离上更稳定。模型最终看到的是很多快慢不同的表盘一起构成的相位指纹。不是“哪个 token 戴高频表盘、哪个 token 戴低频表盘”，而是每个 token 都带着一整排快慢不同的小表盘；真正做 attention 时，两两 token 会在所有表盘上同时对表。

这也是为什么 RoPE 在直觉上非常像 “多尺度时钟系统”：同一位置 p 在不同 band 上对应不同相位，不同位置差 Δ 则在这组时钟上留下不同的组合痕迹。

10 形式定义

10.1 二维旋转矩阵

定义 10.1 (二维旋转矩阵). 对任意实数角度 ϕ，定义

$$\mathbf{R}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

它满足以下基本性质：

$$\mathbf{R}(\phi)^{\mathsf{T}} = \mathbf{R}(-\phi), \qquad \mathbf{R}(\phi_1)\mathbf{R}(\phi_2) = \mathbf{R}(\phi_1+\phi_2), \qquad \mathbf{R}(\phi)^{\mathsf{T}}\mathbf{R}(\phi) = I_2.$$

10.2 RoPE 在单个二维 block 上的定义

定义 10.2 (单 block RoPE). 设第 i 个二维 *block* 的频率为 θ_i，位置为 p。对任意二维向量

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_{2i} \\ x_{2i+1} \end{bmatrix},$$

定义其位置相关表示为

$$\tilde{\mathbf{x}}_p^{(i)} = \mathbf{R}(p\theta_i)\mathbf{x}^{(i)}.$$

写成原子项：

$$\tilde{\mathbf{x}}_p^{(i)} = \begin{bmatrix} x_{2i}\cos(p\theta_i) - x_{2i+1}\sin(p\theta_i) \\ x_{2i}\sin(p\theta_i) + x_{2i+1}\cos(p\theta_i) \end{bmatrix}.$$

10.3 RoPE 在整个 head 上的 block-diagonal 形式

定义 10.3 (整头 RoPE). 设 $d_h = 2m$。定义位置 p 对应的整头旋转算子为

$$\mathcal{R}_p = \mathrm{blkdiag}\big(\mathbf{R}(p\theta_0), \mathbf{R}(p\theta_1), \ldots, \mathbf{R}(p\theta_{m-1})\big) \in \mathbb{R}^{d_h \times d_h}.$$

于是对任意 $\mathbf{x} \in \mathbb{R}^{d_h}$，定义

$$\tilde{\mathbf{x}}_p = \mathcal{R}_p\mathbf{x}.$$

对 query 和 key，写作

$$\tilde{\mathbf{q}}_p = \mathcal{R}_p\mathbf{q}_p, \qquad \tilde{\mathbf{k}}_m = \mathcal{R}_m\mathbf{k}_m.$$

10.4 相对位移进入 attention 的公式

单头 attention logit 定义为

$$\ell(p,m)=\frac{\tilde{\mathbf{q}}_p^{\mathsf{T}}\tilde{\mathbf{k}}_m}{\sqrt{d_h}}.$$

利用 block 结构展开：

$$\ell(p,m)=\frac{1}{\sqrt{d_h}}\sum_{i=0}^{m-1}(\tilde{\mathbf{q}}_p^{(i)})^{\mathsf{T}}\tilde{\mathbf{k}}_m^{(i)}.$$

对每个 block，再利用旋转群性质：

$$(\tilde{\mathbf{q}}_p^{(i)})^{\mathsf{T}}\tilde{\mathbf{k}}_m^{(i)}=(\mathbf{q}^{(i)})^{\mathsf{T}}\mathbf{R}((m-p)\theta_i)\mathbf{k}^{(i)}.$$

若记

$$\mathbf{q}^{(i)}=\begin{bmatrix}q_{2i}\\ q_{2i+1}\end{bmatrix},\qquad \mathbf{k}^{(i)}=\begin{bmatrix}k_{2i}\\ k_{2i+1}\end{bmatrix},$$

则第 i 个 band 的分数可写成

$$s_i(p,m)=(q_{2i}k_{2i}+q_{2i+1}k_{2i+1})\cos((m-p)\theta_i)+(q_{2i+1}k_{2i}-q_{2i}k_{2i+1})\sin((m-p)\theta_i).$$

因此整个单头 logit 为

$$\boxed{\ell(p,m)=\frac{1}{\sqrt{d_h}}\sum_{i=0}^{m-1}\Big[(q_{2i}k_{2i}+q_{2i+1}k_{2i+1})\cos((m-p)\theta_i)+(q_{2i+1}k_{2i}-q_{2i}k_{2i+1})\sin((m-p)\theta_i)\Big].}$$

这就是 RoPE 最重要的原子项公式。

10.5 定义中每个条件的作用

这个定义里的每个条件都不是装饰，而是有功能的：

(1) d_h 为偶数：保证能把 head 拆成二维 block；

(2) 每个 block 一个频率 θ_i：提供多尺度；

(3) 使用二维旋转矩阵 $\mathbf{R}(\phi)$：保证长度保持和群性质；

(4) 对 query 与 key 使用同类位置算子 $\mathcal{R}_p,\mathcal{R}_m$：保证内积中能出现 $\mathbf{R}((m-p)\theta_i)$；

(5) 不作用于 value：保持“位置主要控制比较关系，而非直接改写汇聚内容”。

注 10.4 (复数写法只是压缩记号). 若定义

$$z_i=q_{2i}+\mathrm{i}q_{2i+1},\qquad w_i=k_{2i}+\mathrm{i}k_{2i+1},$$

则单头 logit 也可写成

$$\ell(p,m)=\frac{1}{\sqrt{d_h}}\sum_{i=0}^{m-1}\Re\Big(z_i\overline{w_i}\,e^{\mathrm{i}(m-p)\theta_i}\Big).$$

但本讲义更强调实数域二维 block 的原子项展开，因为这更容易看清每个通道在做什么。

11 基本操作

11.1 构造频率表

最常见的构造是

$$\theta_i = B^{-2i/d_h}, \qquad B = 10000.$$

当 $d_h = 2m$ 时，$i = 0, \dots, m-1$。例如 $d_h = 8$ 时有

$$\theta_0 = 1, \quad \theta_1 = 0.1, \quad \theta_2 = 0.01, \quad \theta_3 = 0.001.$$

注意，这里 θ_i 常被代码实现成 inverse frequency，即直接用于乘位置得到相位。

频率表一旦固定，模型的“时间尺度分解”也就固定了。训练能学习的是各 band 上内容系数 a_i, b_i 如何配合，而不是频率本身如何重排。

11.2 位置到相位的映射

原始 RoPE 采用

$$\phi_i(p) = p\theta_i.$$

scaled RoPE 则采用

$$\phi_i(p) = g(p)\theta_i,$$

其中 g 可能是：

(1) 线性缩放 $g(p) = p/\alpha$；

(2) 训练长度到推理长度的线性插值；

(3) 分段平滑函数；

(4) 针对不同 band 采用不同缩放率的混合方案。

无论形式多复杂，实质都是在改“位置到相位”的映射，而不是改“二维旋转”这一核心部件。

11.3 head 的二维分解

给定 $\mathbf{x} \in \mathbb{R}^{d_h}$，其二维分解方式是固定的：

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_{d_h-2} \\ x_{d_h-1} \end{bmatrix} \quad \rightsquigarrow \quad \left(\begin{bmatrix} x_0 \\ x_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}, \ldots, \begin{bmatrix} x_{d_h-2} \\ x_{d_h-1} \end{bmatrix} \right).$$

工程里常用如下伪代码思想：

```
for each token position p:
    for each band i:
        x_even = x[2i]
        x_odd  = x[2i+1]
        c = cos(phi_i(p))
        s = sin(phi_i(p))
        x_rot[2i]   = x_even * c - x_odd * s
        x_rot[2i+1] = x_even * s + x_odd * c
```

这正是 RoPE 的实现骨架。

11.4 在 attention logit 中度量距离效应

由于

$$\ell(p, m) = \frac{1}{\sqrt{d_h}} \sum_i c_i(\Delta), \qquad \Delta = m - p,$$

其中

$$c_i(\Delta) = a_i \cos(\Delta\theta_i) + b_i \sin(\Delta\theta_i),$$

所以“距离效应”就是看 Δ 改变时每个 band 的相位如何变化。尤其可以关注：

(1) $\Delta \to \Delta + 1$ 时局部变化有多大；

(2) Δ 很大时是否出现相位回绕；

(3) 多 band 叠加后是否仍能区分不同距离模式。

11.5 频率缩放、插值与长上下文扩展

设训练时支持长度 L_{train}，希望推理到更长长度 L_{test}。若直接沿用原始 p，则高频相位增速可能太快。最简单的线性插值想法是把测试位置 p 投到训练位置坐标中：

$$g(p) = p \cdot \frac{L_{\text{train}}}{L_{\text{test}}}.$$

于是

$$\phi_i(p) = \theta_i \, p \frac{L_{\text{train}}}{L_{\text{test}}}.$$

如果 $L_{\text{test}} > L_{\text{train}}$，则相位整体变慢。这会牺牲一部分近邻精细度，换取更平滑的长程行为。因此，长上下文扩展本质上是在

$$\text{局部分辨率} \quad \text{和} \quad \text{远程稳定性}$$

之间重新分配相位预算。

12 关键引理

12.1 旋转的内积相减性质

引理 12.1 (相对位移折叠). *对任意二维向量 $\mathbf{q}, \mathbf{k} \in \mathbb{R}^2$ 和任意角度 $a, b \in \mathbb{R}$，有*

$$(\mathbf{R}(a)\mathbf{q})^{\mathsf{T}}(\mathbf{R}(b)\mathbf{k}) = \mathbf{q}^{\mathsf{T}}\mathbf{R}((b-a))\mathbf{k}.$$

证明. 因为

$$(\mathbf{R}(a)\mathbf{q})^{\mathsf{T}}(\mathbf{R}(b)\mathbf{k}) = \mathbf{q}^{\mathsf{T}}\mathbf{R}(a)^{\mathsf{T}}\mathbf{R}(b)\mathbf{k}.$$

又由于

$$\mathbf{R}(a)^{\mathsf{T}} = \mathbf{R}(-a), \qquad \mathbf{R}(-a)\mathbf{R}(b) = \mathbf{R}(b-a),$$

故

$$(\mathbf{R}(a)\mathbf{q})^{\mathsf{T}}(\mathbf{R}(b)\mathbf{k}) = \mathbf{q}^{\mathsf{T}}\mathbf{R}((b-a))\mathbf{k}.$$

证毕。 □

该引理是 RoPE 的灵魂：它把两个绝对位置 a, b 压缩成了一个差值 $b-a$。

12.2 旋转保持长度

引理 12.2 (长度保持). *对任意 $\mathbf{x} = [x_1, x_2]^{\mathsf{T}} \in \mathbb{R}^2$ 与任意 $\phi \in \mathbb{R}$，都有*

$$\|\mathbf{R}(\phi)\mathbf{x}\|_2 = \|\mathbf{x}\|_2.$$

证明. 先写出旋转后的坐标:

$$\mathbf{R}(\phi)\mathbf{x} = \begin{bmatrix} x_1\cos\phi - x_2\sin\phi \\ x_1\sin\phi + x_2\cos\phi \end{bmatrix}.$$

于是平方范数为

$$\begin{aligned} \|\mathbf{R}(\phi)\mathbf{x}\|_2^2 &= (x_1\cos\phi - x_2\sin\phi)^2 + (x_1\sin\phi + x_2\cos\phi)^2 \\ &= x_1^2\cos^2\phi - 2x_1x_2\cos\phi\sin\phi + x_2^2\sin^2\phi \\ &\quad + x_1^2\sin^2\phi + 2x_1x_2\sin\phi\cos\phi + x_2^2\cos^2\phi \\ &= x_1^2(\cos^2\phi + \sin^2\phi) + x_2^2(\sin^2\phi + \cos^2\phi) \\ &\quad + (-2x_1x_2\cos\phi\sin\phi + 2x_1x_2\sin\phi\cos\phi) \\ &= x_1^2 + x_2^2. \end{aligned}$$

因此

$$\|\mathbf{R}(\phi)\mathbf{x}\|_2 = \|\mathbf{x}\|_2.$$

证毕。 □

这个引理保证 RoPE 不会把位置信号变成范数噪声。

12.3 多 band 带来多尺度表示

引理 12.3 (多尺度叠加). *若单头 logit 可写成*

$$\ell(\Delta) = \frac{1}{\sqrt{d_h}} \sum_{i=0}^{m-1} \big(a_i\cos(\Delta\theta_i) + b_i\sin(\Delta\theta_i)\big),$$

且存在不同量级的频率 θ_i, 则 $\ell(\Delta)$ 是快慢不同振荡成分的叠加, 因此具有多尺度位移敏感性。

证明. 每一项

$$a_i\cos(\Delta\theta_i) + b_i\sin(\Delta\theta_i)$$

都是频率为 θ_i 的单尺度谐波。对不同 i, 周期长度

$$T_i = \frac{2\pi}{\theta_i}$$

不同。当 θ_i 覆盖多个数量级时, $\ell(\Delta)$ 由多个不同周期的谐波叠加组成, 因此同时包含快变成分与慢变成分。这正是多尺度表示。

其中重要的一点是: 同一对 token 的相对位移 Δ 并不是只被某一个 band 测量, 而是会在所有 band 上同时进入

$$a_i\cos(\Delta\theta_i) + b_i\sin(\Delta\theta_i).$$

随后这些不同尺度的响应再被加总成单个 logit。所以多 band 的意义不是保留多个彼此独立的最终输出, 而是共同塑造总的距离响应曲线。 □

注 12.4 (短尺子与长尺子). 把高频 band 看成短尺子，它对很小的位移差就会发生明显变化，因而更擅长区分近邻；把低频 band 看成长尺子，它在大距离上变化更平滑，因而更适合作为长程稳定通道。单个整体频率只有一把尺子，而 RoPE 用许多不同长度的尺子同时测量同一对 token，这就是多尺度的直观含义。

12.4 二维旋转为何是基本块

引理 12.5 (二维是最小非平凡连续正交块). *在实数域上，若要求位置变换既连续依赖一个角度参数，又保持欧氏长度，则最小的非平凡块维度是 2。*

证明. 在一维实数空间中，正交群只有

$$O(1) = \{1, -1\},$$

它是离散集合，没有连续角度参数，不能表达“位置越往后，相位连续累积”。而在二维空间中，

$$SO(2) = \left\{ \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} : \ \phi \in \mathbb{R} \right\},$$

恰好提供一个连续角度参数 ϕ，同时保持长度。故二维是最小非平凡连续正交块。

更高维旋转当然也存在，但它们要么需要更多参数、解释和实现都更复杂，要么在线性代数上本质上仍可分解成多个二维旋转块（再加上可能的平凡 ± 1 块）。因此把“每两维一组”作为基本单位，正是实数空间里最自然、最省参数、又最容易保持相对位移结构的做法。 □

12.5 这些引理分别解决了什么障碍

这几条引理并不是孤立的数学事实，而是分别对应不同工程障碍：

(1) 内积相减性质解决“如何把绝对位置变成相对位移”；

(2) 长度保持解决“位置注入会不会破坏向量几何”；

(3) 多 band 叠加解决“如何兼顾近距离和远距离”；

(4) 二维基本块解决“为什么结构既简单又够用”。

把这四条放在一起，RoPE 的设计就变得非常自然。

13 核心定理（结构性结论）

13.1 相对位移表示结论

定理 13.1 (相对位移进入 logit)**.** 设 $d_h = 2m$, *RoPE* 频率表为 $\theta_0, \dots, \theta_{m-1}$。则对任意 *query* 位置 p 与 *key* 位置 m，单头 *logit* 可写成

$$\ell(p, m) = \frac{1}{\sqrt{d_h}} \sum_{i=0}^{m-1} \Big[a_i \cos((m-p)\theta_i) + b_i \sin((m-p)\theta_i) \Big],$$

其中

$$a_i = q_{2i} k_{2i} + q_{2i+1} k_{2i+1}, \qquad b_i = q_{2i+1} k_{2i} - q_{2i} k_{2i+1}.$$

证明. 由整头 block-diagonal 定义可知

$$\ell(p, m) = \frac{1}{\sqrt{d_h}} \sum_i (\mathbf{R}(p\theta_i)\mathbf{q}^{(i)})^\mathsf{T} (\mathbf{R}(m\theta_i)\mathbf{k}^{(i)}).$$

由引理 12.1 得

$$(\mathbf{R}(p\theta_i)\mathbf{q}^{(i)})^\mathsf{T} (\mathbf{R}(m\theta_i)\mathbf{k}^{(i)}) = (\mathbf{q}^{(i)})^\mathsf{T} \mathbf{R}((m-p)\theta_i)\mathbf{k}^{(i)}.$$

再按二维坐标逐项展开，即得结论。 □

这一定理给出的不是“位置大概会影响打分”，而是一个完全显式的表达式：位置依赖只出现在 $\cos((m-p)\theta_i)$ 与 $\sin((m-p)\theta_i)$ 中。

13.2 多尺度表示结论

定理 13.2 (多尺度结论)**.** 若频率集合 $\{\theta_i\}$ 覆盖多个数量级，则 *RoPE logit* 是一组不同尺度位移响应函数的线性叠加。

证明. 由上一定理，logit 是

$$\ell(\Delta) = \frac{1}{\sqrt{d_h}} \sum_i c_i(\Delta), \qquad c_i(\Delta) = a_i \cos(\Delta\theta_i) + b_i \sin(\Delta\theta_i).$$

每个 c_i 的周期长度为 $2\pi/\theta_i$。当 θ_i 跨多个数量级时，周期长度也跨多个数量级。故 $\ell(\Delta)$ 包含快慢不同的位移响应，形成多尺度表示。 □

这一点解释了 RoPE 为什么比单频旋转稳健得多。

13.3 二维分解结论

定理 13.3 (二维分解结论)**.** *在实数域与长度保持约束下，RoPE 使用二维 block 是实现连续相位编码的最小可行结构。*

证明. 由引理 12.4，一维正交变换只有 ± 1，无法连续编码位置。二维旋转群 $SO(2)$ 则提供一个连续角度参数，并保持长度。因此二维是最小可行块。 □

这一定理意味着：RoPE 的“每两维一组”不是随手切的，而是有结构必要性的。

13.4 正交不变量结论

定理 13.4 (正交不变量结论)**.** *对任意位置 p 和任意 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{d_h}$，整头旋转算子 $\mathcal{R}_p$ 满足*

$$\|\mathcal{R}_p \mathbf{x}\|_2 = \|\mathbf{x}\|_2, \qquad (\mathcal{R}_p \mathbf{x})^\mathsf{T} (\mathcal{R}_p \mathbf{y}) = \mathbf{x}^\mathsf{T} \mathbf{y}.$$

证明. $\mathcal{R}_p$ 是由一组二维正交矩阵组成的 block-diagonal 矩阵，故整体仍为正交矩阵：

$$\mathcal{R}_p^\mathsf{T} \mathcal{R}_p = I_{d_h}.$$

于是

$$\|\mathcal{R}_p \mathbf{x}\|_2^2 = \mathbf{x}^\mathsf{T} \mathcal{R}_p^\mathsf{T} \mathcal{R}_p \mathbf{x} = \mathbf{x}^\mathsf{T} \mathbf{x} = \|\mathbf{x}\|_2^2.$$

同理

$$(\mathcal{R}_p \mathbf{x})^\mathsf{T} (\mathcal{R}_p \mathbf{y}) = \mathbf{x}^\mathsf{T} \mathcal{R}_p^\mathsf{T} \mathcal{R}_p \mathbf{y} = \mathbf{x}^\mathsf{T} \mathbf{y}.$$

证毕。 □

这个结论说明：若两个向量处在同一位置上一起旋转，它们的夹角与内积保持不变。RoPE 真正改变的是“跨位置比较”。

13.5 数学等价与工程可用性的区别

注 13.5 (等价不等于同样好实现)*.* 有时不同论文或代码会写出不同符号约定：例如有人使用 $\Delta = p - m$，有人使用 $\Delta = m - p$；有人把正弦项写成 $q^\perp \cdot k$，但 $q^\perp$ 的定义符号不同。这些在数学上可能等价，但工程上未必同样清楚、同样好调试。RoPE 的工程价值，不只在于“公式成立”，还在于它可以被写成非常稳定的逐 block 实现。

14 评价标准

14.1 什么算好

对位置编码而言，“好”至少意味着四件事：

(1) 能稳定区分顺序；

(2) 能让 relative displacement 比较自然地进入模型；

(3) 在训练长度外不过早崩坏；

(4) 参数和算力代价可接受。

RoPE 的好处在于，它在这四点上都给出均衡答案，特别是“relative 进入打分公式”这一点非常漂亮。

14.2 什么算坏

“坏”的典型表现包括：

(1) 位置只在输入层模糊地混入，没有清晰结构；

(2) 长度扩大后迅速失效；

(3) 参数量、缓存开销或实现复杂度太高；

(4) 近距离和远距离只能顾一头。

如果一个方法只是换了很多符号，最后却没有解决这些问题，那就只是形式变化，不是结构改进。

14.3 什么算深刻

“深刻”的机制，不只是能用，而且能说明为什么这样设计刚好命中问题本身。RoPE 的深刻之处在于：

- 它把位置从输入层问题变成比较器问题；
- 它把绝对位置差值化；
- 它用二维旋转这个最小几何对象达成目标；
- 它在数学结构与工程成本之间非常平衡。

14.4 什么算无效

若某种“改进”做了大量复杂设计，但最终没有改善 relative position 的表达，也没有显著提升长上下文稳定性，却让实现、训练、推理更麻烦，那就可以视为无效或至少性价比很低。

14.5 RoPE 的工程评价维度

工程上可以从以下维度评价 RoPE 或其变体：

(1) **长度外推：**超过训练长度时是否稳定；

(2) **局部分辨率：**相邻 token 是否容易区分；

(3) **缓存兼容：**是否方便做 KV cache；

(4) **数值稳定：**大相位下是否容易退化；

(5) **实现简洁：**是否只需逐维旋转和 cos/sin 表。

从这些维度看，原始 RoPE 很强，但 scaled RoPE 往往在长上下文上更实用。

15 流派争论

15.1 RoPE 与其他位置编码机制的比较

从结构上比较：

(1) learned absolute embedding：简单，但更偏 absolute；

(2) sinusoidal additive embedding：无参数，但 relative 不是显式主角；

(3) relative bias / ALiBi 一类：relative 明确，但更像在分数层加偏置；

(4) RoPE：把 relative phase 直接写进 query-key 比较器。

RoPE 相比纯偏置法的优势是几何结构更强，相比纯加法法的优势是 relative dependence 更显式。

15.2 原始 RoPE 与 scaled RoPE

原始 RoPE 指

$$\phi_i(p) = p\theta_i.$$

scaled RoPE 则泛指

$$\phi_i(p) = g(p)\theta_i$$

中的 $g \neq p$。二者争论的焦点不是“谁更正统”，而是“在给定训练长度和推理长度差距下，哪个更稳”。

应当坦率地说：原始 RoPE 并不保证无限长度外推；scaled RoPE 也不是免费午餐，它通常通过压缩相位换取长程稳定，但可能损失局部分辨率。这本质是一个 trade-off。

15.3 显式解耦与频率混合

原始 RoPE 采用显式二维解耦。一些变体尝试做 band 间混合、学习相位、学习频率、引入额外调制因子。支持者认为这样表达力更强；反对者认为这会损失 RoPE 原本最宝贵的可控性与简洁性。

在大模型工程里，一个重要经验是：

> 可解释、可控、易实现的结构，往往比更复杂但难调的结构更有生命力。

15.4 表达力、外推性与工程可控性的平衡

位置机制没有绝对赢家，只有任务与资源约束下的折中。RoPE 之所以流行，是因为它在三件事之间取得了不错平衡：

(1) 表达力：多尺度 relative phase；

(2) 外推性：比纯 learned absolute 更自然；

(3) 工程可控性：代价低、实现稳定。

很多“RoPE 改进”本质上都在这三角形里重新找平衡点。

16 应用迁移

16.1 语言模型中的应用

在语言模型中，RoPE 的核心价值是：token 的顺序信息直接进入 attention 打分，而不是仅仅作为输入层提示。这对句法依赖、照应、长句跨越、段落级上下文都很有帮助。尤其在 decoder-only 架构中，RoPE 与 causal attention 的结合非常自然。

16.2 代码模型中的应用

代码比自然语言更强调相对结构：括号匹配、缩进层级、变量定义与使用距离、调用栈局部模式等，都很适合 relative-aware 的位置机制。RoPE 的多尺度特性在这里很有价值：高速 band 捕捉局部语法，慢速 band 支持更长的引用关系。

16.3 视觉与多模态中的迁移

在视觉模型里，位置不再是一维序列，而可能是二维网格。常见做法是做轴向扩展：对 x 轴和 y 轴分别构造旋转，或者把二维位置拆成多个轴向 phase 后再拼回。多模态中也常见类似思想，例如文本轴、图像轴、时间轴分别处理。

需要注意的是：视觉中的相对位置不只是“前后”，还包括“上下左右”。因此一维 RoPE 的思想通常要做轴向推广，而不是直接照搬。

16.4 时间序列中的迁移

时间序列天然适合 RoPE，因为很多序列任务都关注相对滞后、周期、局部模式与长程趋势。高频 band 可以捕捉短周期与局部突变，低频 band 可以保留慢变化趋势。但如果序列具有非平稳采样间隔或强外生时间标记，单纯 RoPE 往往还需要与显式时间特征结合。

16.5 RoPE 失效或变差的条件

RoPE 并不是在所有条件下都同样强。常见退化情形包括：

(1) 推理长度远超训练长度，高频 band 相位严重回绕；

(2) 任务需要严格非周期的绝对位置识别；

(3) head 维度太小，band 数量不够；

(4) 位置缩放与训练分布不匹配；

(5) 多轴场景中，位置结构比一维序列复杂得多。

理解这些边界条件，比盲目神化 RoPE 更重要。

17 训练闭环

17.1 能否预测

学会 RoPE 的第一个检验是：能否预测公式形态。如果给你一个新的位置机制，你应该能问：

它最后进入 logit 时，是绝对位置、相对位移，还是两者混合?

对 RoPE，你应能一眼判断：最终会出现 $\cos((m-p)\theta_i)$ 与 $\sin((m-p)\theta_i)$。

17.2 能否举例

第二个检验是：能否自己举出二维与多 band 数值例子，并手算出旋转后的坐标、内积和最终分数。如果只能背定义，不能算出

$$\tilde{q}_{2i} = q_{2i}\cos\phi_i - q_{2i+1}\sin\phi_i$$

这种原子项，那就还没有真正掌握。

17.3 能否反驳

第三个检验是：能否指出反例为什么不行。例如你应能解释：

- learned absolute embedding 为什么不是 clean relative；
- 单频旋转为什么不能多尺度；
- 稠密大旋转为什么工程代价大且不透明。

17.4 能否证明

第四个检验是：能否独立证明最关键的两三条引理。至少应能从头推导出：

$$(\mathbf{R}(a)\mathbf{q})^{\mathsf{T}}(\mathbf{R}(b)\mathbf{k}) = \mathbf{q}^{\mathsf{T}}\mathbf{R}(b-a)\mathbf{k},$$

以及

$$\|\mathbf{R}(\phi)\mathbf{x}\|_2 = \|\mathbf{x}\|_2.$$

这两条证明一旦能自己写出来，RoPE 的骨架就立住了。

17.5 能否应用

第五个检验是：能否把 RoPE 用到具体模型设计中。比如当你面对更长上下文时，是否知道可以从

$$\phi_i(p) = p\theta_i \quad \rightarrow \quad \phi_i(p) = g(p)\theta_i$$

这个角度思考，而不是只会机械调参。

17.6 能否教给别人

最后一个检验是：能否把它讲给别人听。如果你能只用一句话解释 “RoPE 就是每两维一个小表盘，位置就是转角，attention 比较的是相位差”，再用一页纸把公式推到原子项，那就说明你已经真正形成了闭环理解。

18 总结

18.1 RoPE 的一句话本质

RoPE 的一句话本质是：

> 把位置索引变成相位旋转，把绝对位置比较变成相对位移比较。

它不是把位置“加”进去，而是把向量“转”过去。

18.2 RoPE 的数学核心

数学核心只有三层：

(1) 每两维做一次二维旋转；

(2) 旋转矩阵是正交矩阵，保持长度；

(3) 旋转内积满足

$$\mathbf{R}(a)^{\top}\mathbf{R}(b) = \mathbf{R}(b-a),$$

因此位置差自然进入 attention logit。

最终单头 logit 的原子项公式是

$$\ell(p,m) = \frac{1}{\sqrt{d_h}} \sum_{i=0}^{m-1} \Big[(q_{2i}k_{2i} + q_{2i+1}k_{2i+1}) \cos((m-p)\theta_i) + (q_{2i+1}k_{2i} - q_{2i}k_{2i+1}) \sin((m-p)\theta_i) \Big].$$

18.3 RoPE 的工程核心

工程核心也很简洁：

(1) 零或极少额外参数；

(2) 与标准 attention 管线兼容；

(3) 多 band 覆盖多尺度；

(4) 可通过缩放 $g(p)$ 扩展长上下文。

这就是它在大模型里极具生命力的原因。

18.4 后续可展开主题

如果继续往下学，可以展开至少四个方向：

(1) scaled RoPE 的不同相位映射 $g(p)$；

(2) RoPE 与 ALiBi、relative bias 的系统比较；

(3) 二维/多轴 RoPE 在视觉和多模态中的推广；

(4) 超长上下文中相位回绕、外推失效与补救策略。

最后的记忆钩子： *RoPE 不是在 token 上贴位置标签，而是在比较两个 token 时，让它们的相位差自己说话。*

A 附录：单 band 公式的一次性原子项展开

为了便于查阅，这里把单 band 公式完整地再写一遍。设

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \end{bmatrix}, \qquad \mathbf{k} = \begin{bmatrix} k_0 \\ k_1 \end{bmatrix}, \qquad \Delta = m - p.$$

则

$$\ell_{\text{1-band}}(p, m) = (\mathbf{R}(p\theta)\mathbf{q})^{\mathsf{T}}(\mathbf{R}(m\theta)\mathbf{k}).$$

完全展开是

$$\begin{aligned}
\ell_{\text{1-band}}(p, m) &= (q_0 \cos p\theta - q_1 \sin p\theta)(k_0 \cos m\theta - k_1 \sin m\theta) \\
&\quad + (q_0 \sin p\theta + q_1 \cos p\theta)(k_0 \sin m\theta + k_1 \cos m\theta) \\
&= q_0 k_0 \cos p\theta \cos m\theta - q_0 k_1 \cos p\theta \sin m\theta \\
&\quad - q_1 k_0 \sin p\theta \cos m\theta + q_1 k_1 \sin p\theta \sin m\theta \\
&\quad + q_0 k_0 \sin p\theta \sin m\theta + q_0 k_1 \sin p\theta \cos m\theta \\
&\quad + q_1 k_0 \cos p\theta \sin m\theta + q_1 k_1 \cos p\theta \cos m\theta \\
&= (q_0 k_0 + q_1 k_1)\big(\cos p\theta \cos m\theta + \sin p\theta \sin m\theta\big) \\
&\quad + (q_1 k_0 - q_0 k_1)\big(\cos p\theta \sin m\theta - \sin p\theta \cos m\theta\big) \\
&= (q_0 k_0 + q_1 k_1) \cos((m - p)\theta) + (q_1 k_0 - q_0 k_1) \sin((m - p)\theta).
\end{aligned}$$

B 附录：原始 RoPE 与线性缩放 RoPE 的对照

原始 RoPE：

$$\phi_i(p) = p\theta_i.$$

线性缩放版：

$$\phi_i(p) = \frac{p}{\alpha}\theta_i.$$

于是相对相位差分别为

$$\Delta\phi_i^{\text{raw}} = (m - p)\theta_i, \qquad \Delta\phi_i^{\text{scaled}} = \frac{m - p}{\alpha}\theta_i.$$

可见缩放并不改变“只依赖位移差”这一结构，它改变的是位移差进入相位的速度。

参考文献

[1] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, and Bo Wen. RoFormer: Enhanced Transformer with Rotary Position Embedding.

[2] Ashish Vaswani et al. Attention Is All You Need.

[3] Ofir Press, Noah A. Smith, and Mike Lewis. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation.