

Instruções: Cada questão possui exatamente uma alternativa correta.

Questões

- I) Por que o mecanismo de self-attention padrão precisa de positional encoding?
- a) Porque a função softmax não é diferenciável sem informação de posição.
 - b) Porque o score de atenção $\mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d}$ depende apenas dos valores dos vetores $\mathbf{q}_i$ e $\mathbf{k}_j$, e não dos índices i e j .
 - c) Porque a multiplicação de matrizes não é comutativa.
 - d) Porque os token embeddings são inicializados com zero e precisam de um sinal não nulo.
- II) Qual das seguintes alternativas apresenta a fórmula correta do positional encoding senoidal para a dimensão *par* $2k$ na posição i , dada a dimensão de embedding d ?
- a) $\cos(i \cdot 10000^{-2k/d})$
 - b) $\sin(k/10000^{2i/d})$
 - c) $\sin(i/10000^{2k/d})$
 - d) $\cos(i/10000^{2k/d})$
- III) Qual método de positional encoding injeta a informação posicional *dentro* da cabeça de atenção, modificando os vetores de query e key, em vez de somar um vetor ao token embedding de entrada?
- a) Positional encoding senoidal (Vaswani et al., 2017)
 - b) Positional encoding absoluto aprendido (BERT)
 - c) Rotary Position Embedding (RoPE)
 - d) No Positional Encoding (NoPE)
- IV) No **ALiBi**, qual modificação é feita no logit de atenção pré-softmax $A(i, j) = \mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d}$?
- a) Um vetor aprendido é somado a cada query.
 - b) Um escalar fixo $m_h \cdot |i - j|$ é *subtraído* do logit.
 - c) O logit é multiplicado por um fator $e^{-|i-j|}$.
 - d) O vetor key é rotacionado por um ângulo proporcional a j .
- V) Qual é a principal razão pela qual o **NoPE** (No Positional Encoding) generaliza melhor para seqüências mais longas do que o positional encoding absoluto?
- a) O NoPE utiliza uma dimensão de embedding maior, aumentando a capacidade do modelo.
 - b) O positional encoding absoluto adiciona vetores que ficam fora da distribuição de treinamento para posições além do comprimento treinado, enquanto o NoPE não adiciona nada que possa se tornar fora da distribuição.
 - c) O NoPE aplica um decaimento cosseno a todos os pesos de atenção.

- d) O NoPE requer um embedding aprendido para cada posição, o que é mais flexível.
- VI) Qual estratégia de positional encoding absoluto possui um *limite rígido*: posições além de $L_{\max}$ simplesmente não existem no modelo?
- a) Positional encoding senoidal
 - b) ALiBi
 - c) RoPE
 - d) Positional encoding absoluto aprendido (ex.: BERT, GPT original)
- VII) A memória da matriz de atenção em um Transformer escala como $O(L^2)$ no comprimento da sequência L . Aproximadamente quantas vezes mais memória de atenção é necessária ao dobrar o comprimento de contexto de L para $2L$?
- a) $\sqrt{2}$ × mais memória.
 - b) 2× mais memória.
 - c) 4× mais memória.
 - d) 8× mais memória.
- VIII) No RoPE com $d = 64$ e $\theta_k = 10000^{-2k/d}$, qual dimensão possui a *rotação mais rápida* (maior incremento angular por unidade de posição)?
- a) $k = 0$, com $\theta_0 = 1$.
 - b) $k = 8$, no meio do espectro.
 - c) $k = 31$, com $\theta_{31} \approx 10^{-4}$.
 - d) Todas as dimensões giram à mesma velocidade angular.
- IX) Qual método de positional encoding produz, *por construção*, um decaimento *estritamente linear* da contribuição posicional em função de $|i - j|$?
- a) Positional encoding senoidal (Vaswani et al., 2017).
 - b) RoPE (Su et al., 2024).
 - c) ALiBi (Press et al., 2022).
 - d) NoPE (Kazemnejad et al., 2023).
- X) Para o positional encoding senoidal com $d = 4$, quais são os valores das dimensões 0 e 1 na posição $i = 0$?
- a) $PE(0, 0) = 1$, $PE(0, 1) = 0$
 - b) $PE(0, 0) = 0$, $PE(0, 1) = 1$
 - c) $PE(0, 0) = 0$, $PE(0, 1) = 0$
 - d) $PE(0, 0) = 0.5$, $PE(0, 1) = 0.5$
- XI) Qual propriedade do positional encoding senoidal permite teoricamente que uma camada de atenção *aprenda offsets relativos*, mesmo que o encoding seja absoluto?

- a) Todos os valores são limitados ao intervalo $[-1, 1]$, mantendo os gradientes estáveis.
 - b) O encoding usa base 10 000, compatível com tamanhos típicos de vocabulário.
 - c) $PE(i + \delta)$ pode ser escrito como uma transformação linear fixa (multiplicação de matriz) de $PE(i)$ para qualquer offset δ .
 - d) As funções seno e cosseno são ortogonais, tornando cada dimensão independente.
- XII) No ALiBi com $n = 4$ cabeças de atenção, qual é o slope m_h atribuído à cabeça $h = 1$ usando a fórmula $m_h = 2^{-8h/n}$?
- a) 0.5000
 - b) 0.1250
 - c) 0.2500
 - d) 0.0625
- XIII) Qual é a principal diferença algorítmica entre o **Position Interpolation (PI)** e o **YaRN** para estender a janela de contexto de um modelo baseado em RoPE?
- a) O PI usa busca evolutiva para encontrar escalas por dimensão; o YaRN usa escalonamento uniforme.
 - b) O PI aplica um fator de escala uniforme a todas as posições; o YaRN aplica fatores de escala diferentes por dimensão de frequência do RoPE.
 - c) O PI não requer fine-tuning; o YaRN requer dois estágios de fine-tuning.
 - d) O PI modifica a máscara de atenção; o YaRN modifica apenas os vetores key.
- XIV) Por que a máscara de atenção causal em um decoder autorregressivo funciona como informação posicional implícita no **NoPE**?
- a) Cada token na posição i atende exatamente a $i + 1$ tokens, criando um padrão único por linha que se estende naturalmente além do comprimento de treinamento.
 - b) A máscara introduz um padrão senoidal nos pesos de atenção.
 - c) Mascaram tokens futuros força o modelo a memorizar posições absolutas.
 - d) A máscara triangular inferior é equivalente a subtrair as penalidades do ALiBi.
- XV) Qual método de positional encoding oferece extrapolação de comprimento **zero-shot** (*medida apenas por perplexidade*), ou seja, mantém perplexidade baixa em sequências mais longas do que o comprimento de treinamento *sem nenhum fine-tuning adicional*?
- a) Positional encoding senoidal
 - b) RoPE
 - c) Position Interpolation (PI)
 - d) ALiBi
- XVI) Por que o **NoPE** (No Positional Encoding) *não* pode ser aplicado diretamente a modelos *encoder-only* como o BERT?

- a) Porque o BERT utiliza tokenização de subpalavras, que é incompatível com a ausência de positional encoding.
- b) Porque modelos encoder-only não possuem máscara causal; sem ela, o sinal posicional implícito que torna o NoPE viável em decoders desaparece e o modelo se torna invariante a permutações.
- c) Porque o BERT possui um vocabulário pequeno demais para inferir posição a partir do contexto.
- d) Porque o NoPE requer uma camada de normalização especial ausente nos modelos da família BERT.

XVII) Considerando o fenômeno **Lost in the Middle**, qual estratégia prática é *mais eficaz* para mitigar o vale de desempenho no centro do contexto em sistemas de produção?

- a) Sempre preencher todo o contexto disponível com a maior quantidade de documentos possível.
- b) Trocar o positional encoding do modelo de RoPE para senoidal absoluto.
- c) Usar Retrieval-Augmented Generation (RAG) para trazer apenas os trechos relevantes, ou posicionar deliberadamente o conteúdo crítico no início ou no final do prompt.
- d) Aumentar a temperatura de amostragem para forçar o modelo a explorar mais regiões do contexto.

XVIII) 🦴 Considere o RoPE com $d = 4$ e frequências $\theta_0 = 1.0$, $\theta_1 = 0.01$. A query é $\mathbf{q} = [1, 0, 1, 0]$ na posição $i = 1$, e a key é $\mathbf{k} = [0, 1, 0, 1]$ na posição $j = 3$. Usando $\cos(1.0) \approx 0.540$, $\sin(1.0) \approx 0.841$, $\cos(3.0) \approx -0.990$, $\sin(3.0) \approx 0.141$, qual é $\text{score}(1, 3) = \tilde{\mathbf{q}}_1^\top \tilde{\mathbf{k}}_3$?

- a) +0.929
- b) -0.929
- c) 0.000
- d) -0.540

XIX) 🦴 No RoPE, o produto interno dos vetores de query e key rotacionados para um único bloco de frequência 2D com parâmetro θ se expande como:

$$\tilde{\mathbf{q}}_i \cdot \tilde{\mathbf{k}}_j = (q_0 k_0 + q_1 k_1) \cos((j - i)\theta) + (q_1 k_0 - q_0 k_1) \sin((j - i)\theta).$$

Qual é a principal implicação teórica dessa expressão?

- a) O score é sempre zero quando $i = j$, impedindo o self-attention.
- b) O score depende apenas do offset relativo $(j - i)$, e não das posições absolutas i ou j separadamente, confirmando que o RoPE é um positional encoding relativo.
- c) O score é limitado entre -1 e $+1$ para qualquer query e key.
- d) A expressão mostra que o RoPE é equivalente ao positional encoding senoidal somado à entrada.

- XX) 🦴 O Position Interpolation reescala posições como $i \leftarrow i \cdot (L_{\text{train}}/L_{\text{test}})$. Suponha $L_{\text{train}} = 2048$ e $L_{\text{test}} = 8192$ (fator de escala = 0.25). Quais dimensões de frequência do RoPE são *mais prejudicadas* por essa compressão uniforme, e por quê?
- Dimensões de k alto (baixa frequência), pois não completaram muitas rotações e a compressão as empurra ainda mais para fora do intervalo treinado.
 - Dimensões de k baixo (alta frequência), pois já completam muitas rotações completas dentro de L_{train} ; comprimir o espaço de índices super-comprime essas dimensões e destrói a resolução posicional fina.
 - Todas as dimensões são igualmente afetadas, pois o mesmo fator é aplicado uniformemente.
 - Dimensões de k médio, pois suas frequências estão mais próximas da taxa de Nyquist.
- XXI) 🦴 No ALiBi com $n = 8$ cabeças de atenção, a fórmula dos slopes é $m_h = 2^{-8h/n} = 2^{-h}$ para $h \in \{1, \dots, 8\}$. Seja $h = 1$ a cabeça com maior slope ($m_1 = 2^{-1} = 0.5$). Calcule a penalidade ALiBi ($m_h |i - j|$) para essa cabeça nas distâncias $|i - j| = 128$ e $|i - j| = 512$, e determine a diferença entre as duas penalidades.
- Penalidade em 128: 64; penalidade em 512: 256; diferença: 192.
 - Penalidade em 128: 0.5; penalidade em 512: 2.0; diferença: 1.5.
 - Penalidade em 128: 128; penalidade em 512: 512; diferença: 384.
 - Penalidade em 128: 2.0; penalidade em 512: 8.0; diferença: 6.0.
- XXII) 🦴 Considere o positional encoding senoidal $\text{PE}(i) \in \mathbb{R}^d$ com $d = 512$ para posições inteiras $i \geq 0$. É possível recuperar univocamente a posição i a partir do vetor $\text{PE}(i)$?
- Não, pois as funções seno e cosseno são periódicas com período 2π ; posições separadas por um múltiplo inteiro de 2π teriam o mesmo encoding, causando colisões inevitáveis já para posições inteiras pequenas.
 - Sim, para posições práticas ($i \ll 10^4$): as frequências $\omega_k = 10000^{2k/d}$ crescem geometricamente e as combinações $(\sin(i/\omega_k), \cos(i/\omega_k))$ são virtualmente distintas para posições inteiras dentro do intervalo de uso típico, tornando o mapeamento $i \mapsto \text{PE}(i)$ injetivo nesse domínio.
 - Não, pois a norma de $\text{PE}(i)$ é constante igual a $\sqrt{d/2}$ para todo i , tornando impossível distinguir posições distintas.
 - Sim, mas somente utilizando a primeira dimensão $\text{PE}(i, 0) = \sin(i)$, que determina a posição de forma única para qualquer inteiro não negativo.
- XXIII) 🦴 Um modelo com positional encoding absoluto **aprendido** (estilo BERT), treinado com comprimento máximo $L_{\text{max}} = 512$, recebe na inferência uma sequência de comprimento $L = 600$. O que ocorre com os tokens nas posições 513 a 600?
- O modelo extrapola suavemente os embeddings posicionais usando interpolação linear entre $\text{PE}(511)$ e $\text{PE}(0)$.
 - Essas posições não possuem embeddings treinados; a implementação típica lançará um erro de índice (*index out of bounds*) ou, se o índice for truncado, reutilizará vetores de posições iniciais, gerando representações completamente fora da distribuição de treinamento.

- c) O modelo truncará silenciosamente a entrada para os primeiros 512 tokens e ignorará os demais sem erro.
- d) Os embeddings posicionais para $i > 512$ serão automaticamente inicializados com vetores nulos, preservando o processamento sem degradação de desempenho.

XXIV) 🦋 Tanto o RoPE quanto o positional encoding senoidal (Vaswani et al., 2017) utilizam frequências da forma $\theta_k = 10000^{-2k/d}$. Qual afirmação descreve com mais precisão a relação matemática entre os dois métodos?

- a) São algebricamente equivalentes: rotacionar os vetores de query e key com RoPE produz exatamente o mesmo logit de atenção que somar o PE senoidal ao embedding de entrada.
- b) Compartilham o mesmo espectro de frequências, mas diferem fundamentalmente: o PE senoidal codifica posição *absoluta* por adição ao embedding de entrada (antes das projeções lineares W_Q, W_K), enquanto o RoPE codifica posição *relativa* via rotação dos vetores de query e key (após as projeções). Não existe configuração de pesos que torne os dois métodos equivalentes em geral.
- c) São equivalentes quando $W_Q = W_K = I$ (projeções identidade), pois nesse caso adição e rotação têm o mesmo efeito sobre o produto interno.
- d) O PE senoidal é um caso especial do RoPE aplicado somente às duas primeiras dimensões do embedding.

XXV) 🦋 Para o RoPE com $d = 2$ e parâmetro θ , a matriz de rotação na posição i é

$$R(i) = \begin{pmatrix} \cos(i\theta) & -\sin(i\theta) \\ \sin(i\theta) & \cos(i\theta) \end{pmatrix}.$$

Dado que matrizes de rotação satisfazem $R(i)^\top R(j) = R(j - i)$, e que $\tilde{\mathbf{q}}_i = R(i) \mathbf{q}$ e $\tilde{\mathbf{k}}_j = R(j) \mathbf{k}$, como o score de atenção $\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j$ pode ser reescrito?

- a) $\mathbf{q}^\top \mathbf{k} + (i - j)\theta$, mostrando dependência linear na posição relativa.
- b) $\mathbf{q}^\top R(j - i) \mathbf{k}$, mostrando que o score depende apenas do offset relativo $(j - i)$ e não das posições absolutas i ou j individualmente.
- c) $\mathbf{q}^\top R(i + j) \mathbf{k}$, mostrando dependência na posição absoluta somada.
- d) $\|\mathbf{q}\| \|\mathbf{k}\| \cos((i - j)\theta)$, válido apenas quando $\mathbf{q}$ e $\mathbf{k}$ são vetores unitários paralelos.

Gabarito

I, Resposta:(b)

O score de atenção $\mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d}$ é um produto interno que depende apenas dos *valores* dos dois vetores, e não das posições i ou j . Permutar os tokens em qualquer ordem produz os mesmos padrões de atenção, de modo que o modelo é invariante a permutações sem positional encoding.

II, Resposta:(c)

O artigo original do Transformer (Vaswani et al., 2017) define:

$$\text{PE}(i, 2k) = \sin\left(\frac{i}{10000^{2k/d}}\right), \quad \text{PE}(i, 2k + 1) = \cos\left(\frac{i}{10000^{2k/d}}\right).$$

Dimensões pares usam *seno*; dimensões ímpares usam *cosseno* na mesma frequência. As opções (a) e (d) invertem seno e cosseno; a opção (b) troca i e k .

III, Resposta:(c)

O RoPE (Su et al., 2024) aplica uma matriz de rotação $R_\theta(i)$ diretamente aos vetores de query $\mathbf{q}_i$ e key $\mathbf{k}_j$ dentro de cada cabeça de atenção: $\tilde{\mathbf{q}}_i = R_\theta(i) \mathbf{q}_i$. O token embedding $\mathbf{x}_i$ *não* é modificado. O positional encoding senoidal e o aprendido somam um vetor ao embedding de entrada; o NoPE não faz nada.

IV, Resposta:(b)

O ALiBi (Press et al., 2022) modifica o logit pré-softmax para:

$$A_h(i, j) = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}} - m_h |i - j|.$$

A penalidade $m_h \cdot |i - j|$ *cresce com a distância*, induzindo cada cabeça a focar em tokens próximos. Nenhum vetor é somado ao embedding.

V, Resposta:(b)

O positional encoding absoluto atribui um vetor distinto a cada índice de posição visto durante o treinamento. Posições $i \geq L_{\text{train}}$ nunca aparecem no treinamento, portanto esses vetores ficam fora da distribuição no momento do teste, causando degradação. O NoPE não adiciona vetores posicionais, logo nada fica fora da distribuição; apenas a máscara causal muda, e ela o faz de forma previsível e suave.

VI, Resposta:(d)

O positional encoding absoluto aprendido (usado no BERT e nos primeiros GPTs) armazena uma matriz treinável $E \in \mathbb{R}^{L_{\text{max}} \times d}$. Uma posição $i > L_{\text{max}}$ simplesmente não tem linha em E : o embedding não existe. O encoding senoidal possui fórmula fechada para qualquer posição; RoPE e ALiBi são relativos e escalam naturalmente.

VII, Resposta:(c)

A memória da matriz de atenção contém todos os $L \times L$ pares, escalando como $O(L^2)$. Ao dobrar o comprimento de L para $2L$, o número de pares passa de L^2 para $(2L)^2 = 4L^2$, ou seja, **quatro vezes** mais memória. É exatamente esse crescimento quadrático que torna o treinamento em sequências longas economicamente caro e motiva os métodos de extensão de contexto e PEs com boa extrapolação.

VIII, Resposta:(a)

No RoPE, $\theta_k = 10000^{-2k/d}$. Para $k = 0$ tem-se $\theta_0 = 10000^0 = 1$, o valor *máximo* do espectro. Conforme k aumenta, θ_k decai exponencialmente, atingindo $\theta_{31} \approx 10^{-4}$ em $d = 64$. Portanto, a dimensão $k = 0$ rotaciona o vetor a uma velocidade angular muito maior por unidade de posição (rotação rápida $\rightarrow$ posição relativa fina), enquanto k alto rotaciona lentamente (posição relativa grosseira).

IX, Resposta:(c)

O ALiBi adiciona um termo $-m_h \cdot |i - j|$ ao logit de atenção, ou seja, uma função estritamente linear da distância $|i - j|$. RoPE produz decaimento suave mas dependente da frequência (combinação de senos/cossenos), o senoidal absoluto tem contribuição imprevisível ao multiplicar embeddings, e o NoPE não impõe forma alguma: o decaimento, se existir, é puramente aprendido. Apenas o ALiBi garante decaimento linear por construção.

X, Resposta:(b)

Com $d = 4$ e $i = 0$:

$$\text{PE}(0, 0) = \sin(0 \cdot \omega_0) = \sin(0) = 0, \quad \text{PE}(0, 1) = \cos(0 \cdot \omega_0) = \cos(0) = 1.$$

Na posição 0, toda dimensão de seno vale 0 e toda dimensão de cosseno vale 1, pois todos os argumentos são iguais a 0.

XI, Resposta:(c)

Usando fórmulas de adição trigonométricas, pode-se demonstrar que $\text{PE}(i + \delta) = M(\delta) \text{PE}(i)$ para uma matriz de rotação em blocos *fixa* $M(\delta)$ que depende apenas de δ . Como essa relação linear vale para qualquer δ , uma camada de atenção pode, em princípio, aprender a calcular offsets relativos aplicando a transformação inversa.

XII, Resposta:(c)

$$m_1 = 2^{-8 \cdot 1/4} = 2^{-2} = 0.25.$$

Os quatro slopes para $n = 4$ são: $m_1 = 0.25$, $m_2 = 0.0625$, $m_3 = 0.0156$, $m_4 = 0.0039$. A cabeça 1 tem o *slope mais acentuado* e, portanto, atende mais localmente.

XIII, Resposta:(b)

O Position Interpolation (Chen et al., 2023) reescala todos os índices de posição pelo mesmo fator $L_{\text{train}}/L_{\text{test}}$, comprimindo todas as frequências do RoPE uniformemente. O YaRN (Peng et al., 2024) reconhece que dimensões de alta frequência já estão bem representadas e não precisam de escalonamento, enquanto as de baixa frequência precisam de grande escalonamento. Ele aplica uma estratégia de escala por dimensão (não uniforme) inspirada em interpolação NTK-aware, resultando em melhor retenção do contexto curto.

XIV, Resposta:(a)

Em um decoder causal (autorregressivo), o token i só pode atender às posições $j \leq i$. A linha i da máscara de atenção possui exatamente $i + 1$ entradas não mascaradas, criando um padrão binário *único* para cada posição. Conforme o comprimento da sequência cresce além de L_{train} , novas linhas simplesmente estendem esse padrão suavemente, nada fica fora da distribuição, ao contrário dos vetores posicionais explícitos.

XV, Resposta:(d)

O ALiBi penaliza a atenção com $-m_h \cdot |i - j|$. Quando a sequência ultrapassa L_{train} , a penalidade simplesmente fica maior para tokens distantes, comportamento *inteiramente dentro da distribuição* de treinamento. Press et al. (2022) demonstraram que modelos com ALiBi treinados em 1K tokens *mantêm perplexidade baixa* até 8K tokens sem nenhum fine-tuning. Vale notar que essa afirmação de extrapolação é medida apenas por *perplexidade*: trabalhos posteriores mostraram que o desempenho em tarefas de recuperação de longo alcance (por exemplo, *needle in a haystack*) degrada bem antes do que a perplexidade sugere.

XVI, Resposta:(b)

O NoPE depende inteiramente da máscara causal para gerar um sinal posicional implícito: é a estrutura triangular inferior, com cada linha vendo $i + 1$ tokens, que quebra a invariância a permutações do self-attention. Modelos encoder-only como o BERT usam atenção *bidirecional sem máscara*, de modo que sem nenhum positional encoding eles voltam a ser literalmente invariantes a permutações dos tokens, não conseguindo distinguir “o gato comeu o rato” de “o rato comeu o gato”. Por isso o NoPE é uma técnica restrita a decoders.

XVII, Resposta:(c)

A curva de desempenho em U mostra que o modelo ignora informação no meio do contexto, mesmo dentro da janela suportada. As estratégias eficazes na prática são: (i) **RAG** (*Retrieval-Augmented Generation*), que recupera apenas o trecho relevante e o coloca em uma posição privilegiada do prompt; e (ii) **posicionamento deliberado**, colocando o conteúdo crítico no início ou no final do contexto, onde os efeitos de primazia/recência o favorecem. Apenas aumentar o contexto, trocar o PE ou alterar a temperatura de amostragem não corrige o vale central, que é uma propriedade aprendida da atenção e não do positional encoding.

XVIII, Resposta:(b)

Passo 1: Rotacionar a query $\mathbf{q} = [1, 0, 1, 0]$ em $i = 1$:

$$\text{Bloco } k = 0 \text{ } (i\theta_0 = 1.0 \text{ rad}) : (\cos(1) \cdot 1 - \sin(1) \cdot 0, \sin(1) \cdot 1 + \cos(1) \cdot 0) = (0.540, 0.841)$$

$$\text{Bloco } k = 1 \text{ } (i\theta_1 = 0.01 \text{ rad}) : (\cos(0.01) \cdot 1, \sin(0.01) \cdot 1) \approx (1.000, 0.010)$$

Portanto $\tilde{\mathbf{q}}_1 = [0.540, 0.841, 1.000, 0.010]$.

Passo 2: Rotacionar a key $\mathbf{k} = [0, 1, 0, 1]$ em $j = 3$:

Bloco $k = 0$ ($j\theta_0 = 3.0$ rad) : $(\cos(3) \cdot 0 - \sin(3) \cdot 1, \sin(3) \cdot 0 + \cos(3) \cdot 1) = (-0.141, -0.990)$

Bloco $k = 1$ ($j\theta_1 = 0.03$ rad) : $(\cos(0.03) \cdot 0 - \sin(0.03) \cdot 1, \sin(0.03) \cdot 0 + \cos(0.03) \cdot 1) \approx (-0.030, 1.000)$

Portanto $\tilde{\mathbf{k}}_3 = [-0.141, -0.990, -0.030, 1.000]$.

Passo 3: Produto interno:

$\text{score}(1, 3) = 0.540 \cdot (-0.141) + 0.841 \cdot (-0.990) + 1.000 \cdot (-0.030) + 0.010 \cdot 1.000 \approx -0.076 - 0.832 - 0.030 + 0.010 = -$

XIX, Resposta:(b)

O produto interno expandido:

$$(q_0 k_0 + q_1 k_1) \cos((j - i)\theta) + (q_1 k_0 - q_0 k_1) \sin((j - i)\theta)$$

contém as posições absolutas i e j apenas por meio da combinação $(j - i)$. Os termos $q_0 k_0 + q_1 k_1$ e $q_1 k_0 - q_0 k_1$ dependem apenas do conteúdo vetorial de $\mathbf{q}$ e $\mathbf{k}$, não das posições. Portanto, o score de atenção codifica posição *relativa*: dois pares de tokens com o mesmo offset relativo $\delta = j - i$ sempre recebem a mesma contribuição posicional, independentemente de onde apareçam na sequência.

XX, Resposta:(b)

Com fator de escala $s = L_{\text{train}}/L_{\text{test}} = 0.25$, a posição i é substituída por $0.25 i$. O ângulo de rotação efetivo para a dimensão k torna-se:

$$\phi'(i, k) = 0.25 i \cdot \theta_k.$$

Para dimensões de k **baixo** (alta frequência), $\theta_k \approx 1$: essas dimensões já completam muitas rotações completas dentro de L_{train} . Multiplicar por 0.25 comprime quatro tokens de variação angular em um único token, *super-comprimindo* a resolução e fazendo com que tokens próximos recebam sinais posicionais quase idênticos, destruindo as distinções posicionais finas.

As dimensões de k alto ($\theta_k \ll 1$) mal se movem dentro de qualquer comprimento de sequência razoável e são, portanto, o alvo principal da interpolação. Essa é exatamente a motivação do YaRN e do LongRoPE: manter as dimensões de alta frequência intactas e reescalar apenas as de baixa frequência.

XXI, Resposta:(a)

Com $n = 8$ cabeças e $m_h = 2^{-h}$, a cabeça $h = 1$ possui $m_1 = 2^{-1} = 0.5$.

$$\text{Penalidade em } |i - j| = 128 : m_1 \times 128 = 0.5 \times 128 = 64.$$

$$\text{Penalidade em } |i - j| = 512 : m_1 \times 512 = 0.5 \times 512 = 256.$$

$$\text{Diferença: } 256 - 64 = 192.$$

Para comparação, a cabeça $h = 8$ tem $m_8 = 2^{-8} \approx 0.0039$, produzindo penalidades de apenas ≈ 0.5 e ≈ 2.0 nas mesmas distâncias: essa cabeça tolera contexto longo quase sem penalidade, enquanto a cabeça 1 é fortemente local. Esse espectro de slopes é o mecanismo pelo qual o ALiBi cobre ao mesmo tempo atenção local e global com diferentes cabeças.

XXII, Resposta:(b)

A primeira dimensão de seno usa frequência $\omega_0 = 1$, com período $2\pi \approx 6.28$. Para que dois inteiros $i \neq j$ produzam o mesmo PE, seria necessário que $(j - i)/\omega_k$ fosse múltiplo de 2π para *todos* os k simultaneamente. Como 2π é irracional e as frequências $\omega_k = 10000^{2k/d}$ são multiplicativamente incomensuráveis (para k distintos), isso não acontece para nenhum par de inteiros na faixa de uso prático ($i < 10^4$): o mapeamento é injetivo. A opção (a) erra ao afirmar que inteiros próximos colidem — 2π é irracional, portanto $\lfloor i \rfloor$ nunca repete o seno de $\lfloor j \rfloor$ para i, j inteiros com $|i - j| < 2\pi$. A opção (c) erra ao afirmar norma constante: em geral $\|PE(i)\| \neq \sqrt{d}/2$ exatamente para inteiros finitos. A opção (d) erra porque $\sin(i)$ repete com período irracional; entre inteiros, não há repetição, mas tampouco unicidade garantida por uma única dimensão.

XXIII, Resposta:(b)

O positional encoding aprendido é implementado tipicamente como `nn.Embedding(L_max, d)`, uma tabela com $L_{\max}$ linhas. Consultar o índice $i \geq L_{\max}$ é equivalente a acessar uma linha inexistente:

- Em frameworks como PyTorch, isso gera `IndexError: index out of range`.
- Em implementações que truncam o índice (*clamping*), a posição 600 receberia o embedding da posição 511, repetindo-o para todos os tokens além de $L_{\max}$: o modelo passa a ver múltiplos tokens com a mesma “posição 511”, produzindo representações incoerentes e fora de distribuição.

Esse limite rígido é a principal desvantagem do PE aprendido em relação ao PE senoidal (fórmula fechada para qualquer i), ao ALiBi e ao RoPE, que extrapolam naturalmente sem consultar tabelas.

XXIV, Resposta:(b)

O PE senoidal e o RoPE compartilham o espectro $\theta_k = 10000^{-2k/d}$ por design — ambos precisam de frequências variadas para distinguir um amplo intervalo de posições. Contudo as operações realizadas são fundamentalmente distintas:

- **PE senoidal:** adiciona $PE(i)$ ao token embedding *antes* das projeções W_Q, W_K . O score de atenção contém termos cruzados conteúdo-posição e termos de posição absoluta.
- **RoPE:** aplica $R_\theta(i)$ *após* a projeção $W_Q \mathbf{x}_i$, resultando em $\mathbf{q}^\top R(j - i) \mathbf{k}$ — dependência exclusivamente relativa, sem termos de posição absoluta no score.

Mesmo com $W_Q = W_K = I$, os dois scores diferem: o PE senoidal ainda gera termos cruzados $\mathbf{x}_i^\top PE(j)$ que o RoPE não possui. Não existe, portanto, equivalência algébrica entre os dois métodos.

XXV, Resposta:(b)

Aplicando a definição dos vetores rotacionados:

$$\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j = (R(i) \mathbf{q})^\top (R(j) \mathbf{k}) = \mathbf{q}^\top R(i)^\top R(j) \mathbf{k}.$$

Usando $R(i)^\top = R(-i)$ (matrizes de rotação são ortogonais) e a propriedade de composição $R(-i) R(j) = R(j - i)$:

$$\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j = \mathbf{q}^\top R(j - i) \mathbf{k}.$$

As posições absolutas i e j aparecem apenas por meio do offset relativo $(j - i)$. Dois pares de tokens com o mesmo offset relativo $\delta = j - i$ recebem exatamente a mesma contribuição posicional, independentemente de onde estejam na sequência. Isso distingue o RoPE do PE senoidal e é a base teórica para sua generalização zero-shot a sequências mais longas que as do treinamento: o produto $\mathbf{q}^\top R(\delta) \mathbf{k}$ é bem definido para qualquer δ , não dependendo de uma tabela de embeddings pré-treinada.

Gabarito Detalhado — Questões Difíceis de Positional Encoding

Sobre este apêndice. Esta seção apresenta derivações completas para as questões de nível **Difícil** (marcadas com 🧠). Para cada questão são fornecidos: o enunciado, a alternativa correta e todos os passos intermediários.

Os temas centrais são:

- Prova algébrica da propriedade de posição relativa do **RoPE** (questões XIX, XX e XXVI);
- Cálculo dos *slopes* e das penalidades do **ALiBi** (questão XXII);
- Argumento de injetividade do **positional encoding senoidal** (questão XXIII);
- Análise das consequências da compressão uniforme no **Position Interpolation** (questão XXI);
- Comportamento do PE aprendido além de $L_{\max}$ (questão XXIV);
- Comparação matemática entre RoPE e PE senoidal (questão XXV).

XIX 🧠 — Cálculo Numérico do Score RoPE

Enunciado. Considere o RoPE com $d = 4$ e frequências $\theta_0 = 1.0$, $\theta_1 = 0.01$. A query é $\mathbf{q} = [1, 0, 1, 0]$ na posição $i = 1$, e a key é $\mathbf{k} = [0, 1, 0, 1]$ na posição $j = 3$. Usando os valores trigonométricos

$$\cos(1.0) \approx 0.540, \quad \sin(1.0) \approx 0.841, \quad \cos(3.0) \approx -0.990, \quad \sin(3.0) \approx 0.141,$$

calcule $\text{score}(1, 3) = \tilde{\mathbf{q}}_1^\top \mathbf{k}_3$.

Resposta correta

Alternativa (b): -0.929 .

Derivação completa. O RoPE com $d = 4$ opera em dois blocos 2D independentes, um para cada par de dimensões $(2k, 2k + 1)$.

Bloco $k = 0$ — ângulo de rotação $\alpha = i \theta_0 = 1 \times 1.0 = 1.0$ rad:

A rotação no plano $[q_0, q_1]$ aplica a matriz

$$R(i\theta_0) = \begin{pmatrix} \cos(1.0) & -\sin(1.0) \\ \sin(1.0) & \cos(1.0) \end{pmatrix} = \begin{pmatrix} 0.540 & -0.841 \\ 0.841 & 0.540 \end{pmatrix}.$$

Como $\mathbf{q}_{[0:2]} = [1, 0]^\top$:

$$\tilde{\mathbf{q}}_{1,[0:2]} = \begin{pmatrix} 0.540 & -0.841 \\ 0.841 & 0.540 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.540 \\ 0.841 \end{pmatrix}.$$

Bloco $k = 1$ — ângulo de rotação $\alpha = i \theta_1 = 1 \times 0.01 = 0.01$ rad:

Para ângulos pequenos, $\cos(0.01) \approx 1.000$ e $\sin(0.01) \approx 0.010$. Como $\mathbf{q}_{[2:4]} = [1, 0]^\top$:

$$\tilde{\mathbf{q}}_{1,[2:4]} = \begin{pmatrix} 1.000 & -0.010 \\ 0.010 & 1.000 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1.000 \\ 0.010 \end{pmatrix}.$$

Query rotacionada: $\tilde{\mathbf{q}}_1 = [0.540, 0.841, 1.000, 0.010]$.

Bloco $k = 0$ para key — ângulo de rotação $\beta = j\theta_0 = 3 \times 1.0 = 3.0$ rad:

$$R(j\theta_0) = \begin{pmatrix} -0.990 & -0.141 \\ 0.141 & -0.990 \end{pmatrix}.$$

Como $\mathbf{k}_{[0:2]} = [0, 1]^\top$:

$$\tilde{\mathbf{k}}_{3,[0:2]} = \begin{pmatrix} -0.990 & -0.141 \\ 0.141 & -0.990 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -0.141 \\ -0.990 \end{pmatrix}.$$

Bloco $k = 1$ para key — ângulo de rotação $\beta = j\theta_1 = 3 \times 0.01 = 0.03$ rad:

$\cos(0.03) \approx 1.000$, $\sin(0.03) \approx 0.030$. Como $\mathbf{k}_{[2:4]} = [0, 1]^\top$:

$$\tilde{\mathbf{k}}_{3,[2:4]} = \begin{pmatrix} 1.000 & -0.030 \\ 0.030 & 1.000 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -0.030 \\ 1.000 \end{pmatrix}.$$

Key rotacionada: $\tilde{\mathbf{k}}_3 = [-0.141, -0.990, -0.030, 1.000]$.

Produto interno:

$$\text{score}(1, 3) = \underbrace{0.540 \times (-0.141)}_{\approx -0.076} + \underbrace{0.841 \times (-0.990)}_{\approx -0.832} + \underbrace{1.000 \times (-0.030)}_{=-0.030} + \underbrace{0.010 \times 1.000}_{=+0.010} \approx -0.929.$$

Verificação via expansão analítica. O bloco $k = 0$ sozinho contribui com $(q_0k_0 + q_1k_1) \cos((j - i)\theta_0) + (q_1k_0 - q_0k_1) \sin((j - i)\theta_0)$:

$$(1 \cdot 0 + 0 \cdot 1) \cos(2) + (0 \cdot 0 - 1 \cdot 1) \sin(2) = 0 - \sin(2) \approx -0.909.$$

O bloco $k = 1$ contribui com valor próximo de zero (ângulo muito pequeno):

$$(1 \cdot 0 + 0 \cdot 1) \cos(0.02) + (0 \cdot 0 - 1 \cdot 1) \sin(0.02) \approx -0.020.$$

Soma: $-0.909 + (-0.020) \approx -0.929 \checkmark$

XX 🦴 — Expansão Algébrica do Score RoPE e Posição Relativa

Enunciado. No RoPE, o produto interno dos vetores de query e key rotacionados para um único bloco de frequência 2D com parâmetro θ se expande como:

$$\tilde{\mathbf{q}}_i \cdot \tilde{\mathbf{k}}_j = (q_0k_0 + q_1k_1) \cos((j - i)\theta) + (q_1k_0 - q_0k_1) \sin((j - i)\theta).$$

Qual é a principal implicação teórica dessa expressão?

Resposta correta

Alternativa **(b)**: O score depende apenas do offset relativo $(j - i)$, e não das posições absolutas i ou j separadamente, confirmando que o RoPE é um positional encoding relativo.

Derivação completa da expansão. Definindo a rotação para o bloco 2D:

$$\tilde{\mathbf{q}}_i = \begin{pmatrix} \cos(i\theta) & -\sin(i\theta) \\ \sin(i\theta) & \cos(i\theta) \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \end{pmatrix} = \begin{pmatrix} q_0 \cos(i\theta) - q_1 \sin(i\theta) \\ q_0 \sin(i\theta) + q_1 \cos(i\theta) \end{pmatrix},$$

$$\tilde{\mathbf{k}}_j = \begin{pmatrix} \cos(j\theta) & -\sin(j\theta) \\ \sin(j\theta) & \cos(j\theta) \end{pmatrix} \begin{pmatrix} k_0 \\ k_1 \end{pmatrix} = \begin{pmatrix} k_0 \cos(j\theta) - k_1 \sin(j\theta) \\ k_0 \sin(j\theta) + k_1 \cos(j\theta) \end{pmatrix}.$$

Calculando o produto interno componente a componente:

$$\begin{aligned} \tilde{\mathbf{q}}_i \cdot \tilde{\mathbf{k}}_j &= (q_0 \cos(i\theta) - q_1 \sin(i\theta))(k_0 \cos(j\theta) - k_1 \sin(j\theta)) \\ &\quad + (q_0 \sin(i\theta) + q_1 \cos(i\theta))(k_0 \sin(j\theta) + k_1 \cos(j\theta)). \end{aligned}$$

Expandindo e agrupando os termos com $q_0 k_0$, $q_1 k_1$, $q_0 k_1$ e $q_1 k_0$:

$$\begin{aligned} q_0 k_0 &: \cos(i\theta) \cos(j\theta) + \sin(i\theta) \sin(j\theta) = \cos((j-i)\theta), \\ q_1 k_1 &: \sin(i\theta) \sin(j\theta) + \cos(i\theta) \cos(j\theta) = \cos((j-i)\theta), \\ q_0 k_1 &: -\cos(i\theta) \sin(j\theta) + \sin(i\theta) \cos(j\theta) = -\sin((j-i)\theta), \\ q_1 k_0 &: -\sin(i\theta) \cos(j\theta) + \cos(i\theta) \sin(j\theta) = \sin((j-i)\theta), \end{aligned}$$

onde foram usadas as identidades trigonométricas $\cos(A-B) = \cos A \cos B + \sin A \sin B$ e $\sin(A-B) = \sin A \cos B - \cos A \sin B$.

Portanto:

$$\tilde{\mathbf{q}}_i \cdot \tilde{\mathbf{k}}_j = (q_0 k_0 + q_1 k_1) \cos((j-i)\theta) + (q_1 k_0 - q_0 k_1) \sin((j-i)\theta).$$

Implicação teórica. Os dois coeficientes $(q_0 k_0 + q_1 k_1)$ e $(q_1 k_0 - q_0 k_1)$ dependem apenas do *conteúdo* vetorial de $\mathbf{q}$ e $\mathbf{k}$. As posições absolutas i e j aparecem exclusivamente no argumento $(j-i)$ das funções trigonométricas. Portanto:

1. O score é invariante a translações: substituir (i, j) por $(i + \Delta, j + \Delta)$ não altera o resultado, pois o argumento $(j + \Delta) - (i + \Delta) = j - i$ é idêntico.
2. Dois pares de tokens com o mesmo *offset relativo* $\delta = j - i$ recebem exatamente a mesma contribuição posicional, onde quer que estejam na sequência.
3. Isso confirma que o RoPE é um *positional encoding relativo*: a informação posicional codificada não é “estou na posição absoluta i ”, mas sim “a distância relativa entre mim e o outro token é δ ”.

XXI 🧠 — Compressão Uniforme no Position Interpolation

Enunciado. O Position Interpolation reescala posições como $i \leftarrow i \cdot (L_{\text{train}}/L_{\text{test}})$. Supondo $L_{\text{train}} = 2048$ e $L_{\text{test}} = 8192$ (fator = 0.25), quais dimensões de frequência do RoPE são *mais prejudicadas* por essa compressão?

Resposta correta

Alternativa **(b)**: Dimensões de k **baixo** (alta frequência), pois já completam muitas rotações em L_{train} ; a compressão super-comprime essas dimensões e destrói a resolução posicional fina.

Análise detalhada. O ângulo de rotação acumulado na dimensão k para a posição i é:

$$\phi(i, k) = i \cdot \theta_k = i \cdot 10000^{-2k/d}.$$

Após a interpolação $i \rightarrow s i$ com fator $s = 0.25$:

$$\phi'(i, k) = s i \cdot \theta_k = 0.25 \cdot i \cdot \theta_k.$$

O número de rotações completas realizadas até a posição L_{train} , na dimensão k , é:

$$N_k = \frac{L_{\text{train}} \cdot \theta_k}{2\pi}.$$

Dimensões de k baixo (alta frequência, $\theta_k \approx 1$):

$$N_0 = \frac{2048 \times 1}{2\pi} \approx 326 \text{ rotações completas dentro de } L_{\text{train}}.$$

Cada token ocupa um arco de ≈ 1.0 rad, proporcionando alta resolução posicional local. Com o fator de compressão $s = 0.25$, cada token passa a ocupar apenas 0.25 rad: quatro tokens que antes eram distinguíveis por posição agora ficam “comprimidos” no mesmo intervalo que antes representava um único token, tornando os sinais posicionais indistinguíveis.

Dimensões de k alto (baixa frequência, $\theta_k \ll 1$): Para $k = d/2 - 1$ com $d = 64$, $\theta_{31} = 10000^{-2 \times 31/64} \approx 10^{-4}$.

$$N_{31} = \frac{2048 \times 10^{-4}}{2\pi} \approx 0.033 \text{ rotações em } L_{\text{train}}.$$

Essas dimensões mal se movem e não fornecem resolução posicional fina de qualquer forma; a compressão afeta muito menos a qualidade do sinal nestas dimensões.

Motivação do YaRN. Essa assimetria motivou o YaRN (Peng et al., 2024): ao invés de comprimir todas as dimensões uniformemente, o YaRN aplica:

- **Dimensões de k baixo** (alta frequência): mantém sem compressão, pois já fornecem resolução fina e seriam super-comprimidas.
- **Dimensões de k alto** (baixa frequência): aplica compressão agressiva, pois mal se movem e toleram a reescala.

Esse escalonamento não-uniforme permite que o modelo mantenha a resolução posicional de curto alcance (dependente das dimensões de alta frequência) enquanto estende o alcance global.

XXII 🧠 — Cálculo dos Slopes e Penalidades ALiBi

Enunciado. No ALiBi com $n = 8$ cabeças de atenção, a fórmula dos slopes é $m_h = 2^{-8h/n} = 2^{-h}$ para $h \in \{1, \dots, 8\}$. Para $h = 1$ (cabeça com maior slope), calcule as penalidades em $|i - j| = 128$ e $|i - j| = 512$ e a diferença entre elas.

Resposta correta

Alternativa (a): Penalidade em 128: 64; penalidade em 512: 256; diferença: 192.

Derivação detalhada. Passo 1 — Tabela completa de slopes.

Para $n = 8$ cabeças, $m_h = 2^{-8h/n} = 2^{-8h/8} = 2^{-h}$:

Cabeça h	Slope $m_h = 2^{-h}$	Forma fracionária
1	0.500000	1/2
2	0.250000	1/4
3	0.125000	1/8
4	0.062500	1/16
5	0.031250	1/32
6	0.015625	1/64
7	0.007813	1/128
8	0.003906	1/256

A cabeça $h = 1$ possui o maior slope ($m_1 = 0.5$) e, portanto, a maior penalidade por unidade de distância: essa cabeça é a *mais local* do conjunto. A cabeça $h = 8$ tem o menor slope ($m_8 \approx 0.0039$) e é a *mais global*.

Passo 2 — Cálculo das penalidades para $h = 1$.

$$\text{Penalidade}(|i - j|) = m_1 \times |i - j| = \frac{1}{2} \times |i - j|.$$

$$\text{Em } |i - j| = 128 : \quad \frac{1}{2} \times 128 = \mathbf{64}.$$

$$\text{Em } |i - j| = 512 : \quad \frac{1}{2} \times 512 = \mathbf{256}.$$

$$\text{Diferença: } 256 - 64 = \mathbf{192}.$$

Passo 3 — Interpretação no logit de atenção.

O logit pré-softmax com ALiBi é:

$$A_h(i, j) = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}} - m_h \cdot |i - j|.$$

Para $h = 1$ e $|i - j| = 128$, a penalidade de -64 subtrai 64 do logit bruto. Após o softmax, se o logit sem penalidade fosse ~ 0 , a atenção efetiva para esse token seria $\propto e^{-64}$, praticamente zero: a cabeça 1 ignora tokens a mais de ~ 10 posições de distância de forma efetiva.

Para comparação, a cabeça $h = 8$ subtrairia apenas ≈ 0.5 em $|i - j| = 128$: essa cabeça consegue “ver” contexto de centenas de tokens com penalidade mínima.

Por que espectro de slopes? O conjunto de slopes $\{2^{-1}, 2^{-2}, \dots, 2^{-8}\}$ cobre múltiplas escalas de atenção: atenção local extrema (cabeça 1) até atenção quase global (cabeça 8). Isso permite que o modelo, em camadas profundas, componha relações locais (sintaxe) e globais (semântica de longo alcance) simultaneamente, sem nenhum parâmetro adicional além dos já existentes em cada cabeça de atenção.

XXIII 🦴 — Injetividade do Positional Encoding Senoidal

Enunciado. Considere o positional encoding senoidal $\text{PE}(i) \in \mathbb{R}^d$ com $d = 512$ para posições inteiras $i \geq 0$. É possível recuperar univocamente a posição i a partir do vetor $\text{PE}(i)$?

Resposta correta

Alternativa (b): Sim, para posições práticas ($i \ll 10^4$): as frequências crescem geometricamente e as combinações $(\sin(i/\omega_k), \cos(i/\omega_k))$ são virtualmente distintas para posições inteiras dentro do intervalo de uso típico.

Argumento formal de injetividade. Definição. O positional encoding senoidal é:

$$\text{PE}(i, 2k) = \sin\left(\frac{i}{\omega_k}\right), \quad \text{PE}(i, 2k+1) = \cos\left(\frac{i}{\omega_k}\right), \quad \omega_k = 10000^{2k/d}, \quad k = 0, 1, \dots, \frac{d}{2} - 1.$$

Condição de colisão. Dois inteiros $i \neq j$ com $i, j \geq 0$ produzem $\text{PE}(i) = \text{PE}(j)$ se e somente se, para todo k :

$$\frac{i}{\omega_k} \equiv \frac{j}{\omega_k} \pmod{2\pi},$$

ou seja, $\frac{i-j}{\omega_k}$ é múltiplo inteiro de 2π para todo k .

Análise para $k = 0$. Com $\omega_0 = 10000^0 = 1$:

$$(i - j) \cdot 1 = 2\pi m \quad \text{para algum inteiro } m.$$

Portanto $i - j = 2\pi m$. Como $i - j$ é um inteiro (diferença de dois inteiros) e $2\pi \approx 6.28318\dots$ é **irracional**, a única solução é $m = 0$, o que implica $i = j$.

Conclusão. A frequência $k = 0$ *por si só* é suficiente para garantir que nenhuma colisão existe entre posições inteiras não-negativas distintas quaisquer. O mapeamento $i \mapsto \text{PE}(i)$ é **estritamente injetivo** para todo $i, j \in \mathbb{Z}_{\geq 0}$ com $i \neq j$.

Por que a alternativa (b) diz “para posições práticas”? A formulação “para posições práticas ($i \ll 10^4$)” na alternativa (b) é conservadora e reflete a abordagem do artigo original (Vaswani et al., 2017), que usou bases de frequência $\omega_k = 10000^{2k/d}$ esperando que posições $i \lesssim 10^4$ fossem suficientes para aplicações de NLP. O argumento acima mostra que a injetividade é, na verdade, *universal* sobre os inteiros não-negativos — a irracionalidade de 2π garante isso matematicamente.

Análise das alternativas incorretas.

- (a) — **Errada.** A alternativa afirma que posições separadas por um múltiplo inteiro de 2π teriam o mesmo encoding. Contudo, 2π é irracional, então nenhum múltiplo não-nulo de 2π é inteiro. A premissa da opção (a) nunca é satisfeita para posições inteiras.
- (c) — **Errada.** A norma de $\text{PE}(i)$ é:

$$\|\text{PE}(i)\|^2 = \sum_{k=0}^{d/2-1} \left[\sin^2\left(\frac{i}{\omega_k}\right) + \cos^2\left(\frac{i}{\omega_k}\right) \right] = \sum_{k=0}^{d/2-1} 1 = \frac{d}{2}.$$

A norma é constante igual a $\sqrt{d/2}$ para *toda* i . Isso é correto e decorre da identidade pitagórica, mas a norma constante é indiferente à questão de injetividade: o que importa para distinguir i de j é a *direção* do vetor $PE(i)$, não sua magnitude. Dois vetores podem ter a mesma norma e ainda assim ser distintos.

- **(d) — Errada.** $\sin(i)$ é periódica em $i \in \mathbb{R}$ com período 2π irracional; portanto, para inteiros i, j , $\sin(i) \neq \sin(j)$ sempre que $i \neq j$ e $|i - j| < 2\pi$, mas é possível que $\sin(i) = \sin(j)$ para inteiros distantes — a única dimensão não é suficiente por si só para toda a faixa de inteiros. Contudo, o par $(\sin(i), \cos(i))$ identifica $i \bmod 2\pi$ univocamente, e como 2π é irracional, nenhum inteiro positivo tem a mesma representação que $i = 0$.

XXIV 🦴 — PE Aprendido Além de $L_{\max}$

Enunciado. Um modelo com positional encoding absoluto aprendido (estilo BERT), treinado com $L_{\max} = 512$, recebe na inferência uma sequência de comprimento $L = 600$. O que ocorre com os tokens nas posições 513 a 600?

Resposta correta

Alternativa **(b)**: Essas posições não possuem embeddings treinados; a implementação típica lançará um erro de índice ou, se o índice for truncado, reutilizará vetores de posições iniciais, gerando representações fora da distribuição de treinamento.

Análise da implementação. O positional encoding aprendido é implementado como uma tabela de consulta (*lookup table*):

```
position_embeddings = nn.Embedding(L_max, d)
```

Essa camada armazena uma matriz treinável $E \in \mathbb{R}^{L_{\max} \times d}$, com uma linha por posição $i \in \{0, 1, \dots, L_{\max} - 1\}$.

Cenário 1 — Erro de índice. Em frameworks padrão como PyTorch, consultar `position_embeddings[512]` (com $L_{\max}=512$) acessa o índice 512 de uma tabela com 512 linhas (índices 0–511), produzindo:

```
IndexError: index out of range in self
```

Cenário 2 — Clamping de índice. Algumas implementações aplicam `position_ids = position_ids.clamp(max=L-1)`, que força qualquer posição $i \geq L_{\max}$ a usar o vetor treinado para $i = L_{\max} - 1 = 511$. Os tokens nas posições 512 a 599 recebem todos o *mesmo* embedding posicional (o da posição 511), resultando em:

- Perda completa da distinção posicional para os últimos 88 tokens;
- Atenção entre esses tokens afetada como se todos fossem equidistantes (mesma posição);
- Representações fortemente fora da distribuição de treinamento, pois o modelo nunca viu dois tokens simultâneos com posição 511.

Cenário 3 — Truncamento silencioso (alternativa (c)). Bibliotecas como Hugging Face Transformers, ao tokenizar com `tokenizer(..., truncation=True, max_length=L_max)` sem o usuário pedir *warnings*, podem cortar a entrada para os primeiros $L_{\max}$ *tokens* antes mesmo de chegar ao modelo (geralmente acompanhado de um *warning* discreto). Isso não é o que a alternativa (c) descreve estritamente (“sem erro” e “preservando o processamento”), mas ilustra que o comportamento exato depende muito do *wrapper* usado, em produção é comum ver truncamento silencioso em vez do `IndexError` do PyTorch puro. Em todos os cenários, dados ou representações são degradados, por isso (b) continua sendo a resposta que captura o **problema fundamental** do PE aprendido fora do intervalo de treino.

Contraste com outros métodos.

- **PE senoidal:** possui fórmula fechada $\sin(i/\omega_k)$, calculável para qualquer i . Não consulta tabela.
- **RoPE:** aplica matrizes de rotação parametrizadas por θ_k , válidas para qualquer i .
- **ALiBi:** computa $m_h|i - j|$ para qualquer distância, sem tabela.

XXV 🦴 — Relação Matemática entre RoPE e PE Senoidal

Enunciado. Tanto o RoPE quanto o PE senoidal utilizam frequências da forma $\theta_k = 10000^{-2k/d}$. Qual afirmação descreve com mais precisão a relação matemática entre os dois métodos?

Resposta correta

Alternativa **(b)**: Compartilham o mesmo espectro de frequências, mas diferem fundamentalmente: o PE senoidal codifica posição *absoluta* por adição ao embedding de entrada, enquanto o RoPE codifica posição *relativa* via rotação dos vetores de query e key. Não existe configuração de pesos que torne os dois métodos equivalentes em geral.

Análise comparativa. Espectro de frequências compartilhado. Ambos os métodos utilizam o conjunto de frequências $\{\theta_k\}_{k=0}^{d/2-1}$ com $\theta_k = 10000^{-2k/d}$. Isso não é coincidência: ambos precisam de um espectro que permita distinguir posições em um amplo intervalo (da resolução de token único até o comprimento máximo da sequência).

Diferenças fundamentais.

Característica	PE Senoidal (Vaswani et al.)	RoPE (Su et al.)
Ponto de aplicação	Antes de W_Q, W_K (na entrada)	Após W_Q, W_K
Operação	Adição: $\mathbf{x}_i + \text{PE}(i)$	Rotação: $R(i) \mathbf{q}_i$
Tipo de encoding	Posição absoluta	Posição relativa
Score de atenção	Contém termos cruzados conteúdo-posição	Apenas $\mathbf{q}^\top R(j - i) \mathbf{k}$

Por que não são equivalentes mesmo com $W_Q = W_K = I$?

Com PE senoidal, o logit de atenção é:

$$A_{\text{seno}}(i, j) = (\mathbf{x}_i + \text{PE}(i))^\top (\mathbf{x}_j + \text{PE}(j)) = \mathbf{x}_i^\top \mathbf{x}_j + \mathbf{x}_i^\top \text{PE}(j) + \text{PE}(i)^\top \mathbf{x}_j + \text{PE}(i)^\top \text{PE}(j).$$

O último termo $\text{PE}(i)^\top \text{PE}(j)$ se simplifica via identidades trigonométricas:

$$\text{PE}(i)^\top \text{PE}(j) = \sum_k \cos\left(\frac{j-i}{\omega_k}\right),$$

que depende apenas de $(j-i)$. Contudo, os termos cruzados $\mathbf{x}_i^\top \text{PE}(j)$ e $\text{PE}(i)^\top \mathbf{x}_j$ introduzem dependência de posição absoluta misturada com conteúdo vetorial.

Com RoPE:

$$A_{\text{RoPE}}(i, j) = (R(i)\mathbf{x}_i)^\top (R(j)\mathbf{x}_j) = \mathbf{x}_i^\top R(j-i)\mathbf{x}_j.$$

Não há termos cruzados conteúdo-posição absoluta: a posição entra exclusivamente como offset relativo $R(j-i)$.

Portanto, mesmo quando $W_Q = W_K = I$, os dois logits contêm termos estruturalmente distintos e os dois métodos não são equivalentes para nenhuma escolha de pesos.

XXVI 🦋 — Prova da Propriedade de Posição Relativa do RoPE

Enunciado. Para o RoPE com $d = 2$ e parâmetro θ , a matriz de rotação é

$$R(i) = \begin{pmatrix} \cos(i\theta) & -\sin(i\theta) \\ \sin(i\theta) & \cos(i\theta) \end{pmatrix}.$$

Dado que $R(i)^\top R(j) = R(j-i)$ e que $\tilde{\mathbf{q}}_i = R(i)\mathbf{q}$ e $\tilde{\mathbf{k}}_j = R(j)\mathbf{k}$, como o score $\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j$ pode ser reescrito?

Resposta correta

Alternativa **(b)**: $\mathbf{q}^\top R(j-i)\mathbf{k}$, mostrando que o score depende apenas do offset relativo $(j-i)$.

Prova completa em três partes. Parte 1 — Ortogonalidade de $R(i)$.

$R(i)$ é uma matriz de rotação plana e, portanto, ortogonal:

$$R(i)^\top R(i) = I \implies R(i)^\top = R(i)^{-1} = R(-i).$$

Verificação explícita:

$$R(i)^\top = \begin{pmatrix} \cos(i\theta) & \sin(i\theta) \\ -\sin(i\theta) & \cos(i\theta) \end{pmatrix} = R(-i).$$

Parte 2 — Propriedade de composição $R(a)R(b) = R(a+b)$.

$$\begin{aligned} R(a)R(b) &= \begin{pmatrix} \cos a\theta & -\sin a\theta \\ \sin a\theta & \cos a\theta \end{pmatrix} \begin{pmatrix} \cos b\theta & -\sin b\theta \\ \sin b\theta & \cos b\theta \end{pmatrix} \\ &= \begin{pmatrix} \cos a\theta \cos b\theta - \sin a\theta \sin b\theta & -\cos a\theta \sin b\theta - \sin a\theta \cos b\theta \\ \sin a\theta \cos b\theta + \cos a\theta \sin b\theta & -\sin a\theta \sin b\theta + \cos a\theta \cos b\theta \end{pmatrix} \\ &= \begin{pmatrix} \cos(a+b)\theta & -\sin(a+b)\theta \\ \sin(a+b)\theta & \cos(a+b)\theta \end{pmatrix} = R(a+b). \end{aligned}$$

Parte 3 — Reescrita do score de atenção.

$$\begin{aligned}
\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j &= (R(i) \mathbf{q})^\top (R(j) \mathbf{k}) \\
&= \mathbf{q}^\top R(i)^\top R(j) \mathbf{k} && \text{(transposição de produto)} \\
&= \mathbf{q}^\top R(-i) R(j) \mathbf{k} && \text{(Parte 1)} \\
&= \mathbf{q}^\top R(-i + j) \mathbf{k} && \text{(Parte 2 com } a = -i, b = j) \\
&= \mathbf{q}^\top R(j - i) \mathbf{k}.
\end{aligned}$$

Interpretação teórica. O resultado $\mathbf{q}^\top R(j - i) \mathbf{k}$ mostra que:

1. As posições absolutas i e j cancelam-se durante a derivação; apenas o offset relativo $\delta = j - i$ permanece.
2. A matriz $R(\delta)$ é uma rotação *fixa* por ângulo $\delta\theta$ que pondera o produto interno entre $\mathbf{q}$ e $\mathbf{k}$ de forma dependente da separação entre os tokens.
3. Para $\delta = 0$ (self-attention em $i = j$): $R(0) = I$, e o score reduz-se ao produto interno simples $\mathbf{q}^\top \mathbf{k}$ sem penalidade posicional.
4. O resultado generaliza para $d > 2$: o RoPE aplica blocos de rotação 2D independentes $\{R_k(\delta)\}_{k=0}^{d/2-1}$ a cada par de dimensões, e a mesma argumentação se aplica a cada bloco individualmente.

Extensão para d arbitrário. Para $d = 2M$ (dimensão geral), o RoPE aplica:

$$\tilde{\mathbf{q}}_i^\top \tilde{\mathbf{k}}_j = \sum_{k=0}^{M-1} [\mathbf{q}_{[2k:2k+2]}]^\top R_k(j - i) [\mathbf{k}_{[2k:2k+2]}],$$

onde $R_k(\delta) = R(\delta\theta_k)$ usa a frequência $\theta_k = 10000^{-2k/d}$. Cada bloco k contribui com uma dependência relativa de frequência θ_k distinta, e a soma sobre todos os blocos é o score completo de atenção RoPE, cujo valor depende apenas de $\delta = j - i$ e dos vetores de conteúdo $\mathbf{q}$ e $\mathbf{k}$.