FRAYED ROPE AND LONG INPUTS: A GEOMETRIC PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Rotary Positional Embedding (RoPE) is a widely adopted technique for encoding position in language models, which, while effective, causes performance breakdown when input length exceeds training length. Prior analyses assert (rightly) that long inputs cause channels to rotate "out of distribution," but it is not clear how extra rotation relates to or causes pathological behavior. Through empirical analysis we advance a unified geometric understanding of attention behavior with RoPE. We find that attention induces tight clustering of separated key and query latent point clouds, allowing for creation of sink tokens: placeholders that allow attention heads to avoid token mixing when not required. RoPE applied to longer inputs damages this key/query cluster separation, producing pathological behavior by inhibiting sink token functionality. From this geometric perspective, we propose RoPE-ID (In Distribution), a straightforward modification that allows attention layers to generalize to longer inputs out of the box: apply RoPE with high frequency to a subset of channels. We demonstrate the effectiveness of RoPE-ID for extended inputs using 1B and 3B parameter Transformers on the LongBench and RULER information retrieval benchmarks.

1 INTRODUCTION

Transformer models form the backbone of modern large language models (LLMs), enabling them to capture complex dependencies across long sequences. The attention mechanism in transformers maps inputs into queries, keys, and values: queries and keys determine token relevance through similarity scores, while values provide the content to be aggregated. This separation allows the model to learn both where to attend and what information to extract, producing context-aware representations that drive the success of transformers in natural language processing and beyond.

To enhance the interaction between queries and keys, positional encoding is used to distinguish token order, constituting a fundamental component of transformer design. Rotary Positional Embedding (RoPE) (Su et al., 2023) has emerged as the predominant approach and is now implemented in most state-of-the-art LLMs, including LLaMA (Grattafiori et al., 2024), GPT (OpenAI et al., 2025), and DeepSeek (DeepSeek-AI et al., 2025). However, a key limitation of RoPE is performance degradation when input length exceeds training context. Most attempts to analyze and address this issue attribute the failure to channels rotating "out of distribution," leading to frequency rescaling as a workaround (Chen et al., 2023; bloc97, 2023b;a; Peng et al., 2023; Ding et al., 2024).

Another important phenomenon in transformers is the attention sink, which has been shown to influence long-context generalization (Xiao et al., 2023). The attention sink, typically the first input token, possesses little semantic meaning but consistently large attention scores. Its presence is considered crucial to prevent over-mixing of information, and empirical evidence shows that attention sinks must be preserved when extending context lengths (Xiao et al., 2023; Han et al., 2024).

The relationship between attention, RoPE, and attention sinks – three seemingly disconnected concepts – is the focus of this paper. We propose a unified geometric perspective, based on analysis of popular LLM families including LLaMA, Gemma (Team et al., 2024), and Olmo (Groeneveld et al., 2024). We find that, contrary to the common intuition of attention as a soft nearest-neighbor lookup, queries and keys form tight clusters with minimal overlap, while the sink token resides near the origin (Fig. 1, left). Within the training context length, this separation ensures that the sink token, with its small norm, naturally absorbs the majority of attention weight. Beyond the training context, RoPE disperses and overlaps the query/key clusters. This geometric disruption prevents the sink token from functioning, leading to long-context performance breakdowns (Fig. 1, right).

To address this, we propose RoPE-ID (RoPE In Distribution), a simple yet effective plug-in replacement for RoPE. RoPE-ID combines high-frequency RoPE channels with RoPE-free channels to preserve stable query-key cluster geometry and sink token functionality. RoPE-free channels lower-bound the degree of cluster overlap, while high-frequency RoPE channels encode position while avoiding out-of-distribution effects. We validate RoPE-ID with trained 1B- and 3B-parameter decoder models, evaluated on the LongBench and RULER benchmarks, demonstrating strong context length generalization and improvements over prior tuning-free methods.

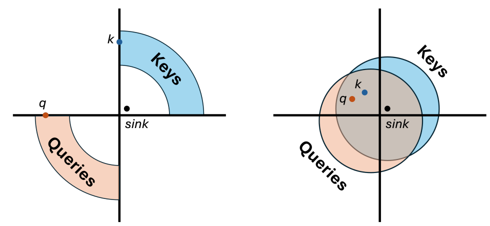


Figure 1: A 2D diagram of our observed latent geometry. **Left:** Keys and queries cluster tightly into opposing point clouds with negative dot products. The sink token has small norm and therefore the greatest dot product. Assigned key/query pair q, k are aligned orthogonally, allowing their dot product to approach and exceed the sink's. **Right:** Beyond the training length, RoPE causes key/query clouds to disperse and overlap, introducing spurious alignment. Sink token no longer functions.

2 Background and Related Work

Position Encoding techniques can be broadly divided into two categories: absolute position embeddings (APE) and relative position embeddings (RPE). APE directly injects position information into latent representations using token-index-dependent vectors, in a fixed sinusoidal (Vaswani et al., 2023) or learnable form (Devlin et al., 2019). APE exhibits limited generalization beyond the training context length, which RPE addresses by encoding distances between token pairs rather than their absolute indices. Notable RPE approaches include T5's relative bias (Raffel et al., 2023) and Alibi (Press et al., 2022), which add position-dependent linear biases to attention logits, and RoPE (Rotary Position Embeddings) (Su et al., 2023), which encodes relative distance as latent angular displacement and has since become the de facto standard for LLMs.

The key insight behind RoPE is that relative position, through properties of rotation, decomposes into independent key and query transformations. RoPE encodes relative position via angular displacement proportional to token distance, interposing a block-diagonal matrix of 2×2 rotations into the key/query dot product. Each submatrix has a constant frequency θ scaling token distance m:

$$\mathbf{R}_{\Theta}^{m} = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ 0 & 0 & \cdots & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix} \tag{1}$$

In practice, this decomposes into independent rotations on query q_i and key k_j at positions i, j:

$$\text{RoPE}(\langle q_i, k_j \rangle) = q_i R_{\Theta}^{i-j} k_j^{\top} = q_i R_{\Theta}^{i} R_{\Theta}^{-j} k_j^{\top} = q_i R_{\Theta}^{i} (k_j R_{\Theta}^{j})^{\top} = \langle r(q_i, i), r(k_j, j) \rangle \tag{2}$$

where $r(\cdot, i)$ represents rotation by $\mathbf{R}_{\Theta}^{i}$. Understanding the impact of this progressive rotation on latent keys and queries is crucial to understanding out-of-distribution failures on long contexts.

Context Length Extension: As large language models are increasingly applied to long-context tasks, substantial research has focused on extending their usable sequence lengths without retraining. Although RPEs such as RoPE are designed to improve long-context generalization, extending beyond the training length often results in performance degradation. To address this, Position Interpolation (PI) (Chen et al., 2023) linearly interpolates position indices within the pre-trained sequence length. NTK-by-parts (bloc97, 2023b) and NTK-aware (bloc97, 2023a) introduce non-linear interpolation schemes inspired by Neural Tangent Kernel dynamics. These methods scale RoPE frequencies based on three groups of frequency dimensions and the target sequence length, and outperform simple PI. YaRN (Yet Another RoPE Extension) (Peng et al., 2023) further integrates previous NTK-based approaches with temperature scaling on attention logits, achieving a 2×

context window extension without fine-tuning. More recently, LongRoPE (Ding et al., 2024) employs evolutionary search to optimize the frequency rescale factors for each dimension. LongRoPE extends context window to beyond 2 million tokens, albeit with multi-step long-context fine-tuning. Here we focus on tuning-free generalization and leave tuning to future work.

Some studies find that RoPE's low-frequency components induce high-norm semantic bands, which become unstable in long contexts (Barbero et al., 2024b) or hinder the encoding of semantic information (Chen et al., 2024). They propose limiting RoPE to a subset of channels, finding this improves performance. Our analysis provides novel perspective and caveats for this technique.

Sink Tokens, or attention sinks, refer to tokens with disproportionately high attention despite a lack of semantic meaning (Xiao et al., 2023). This phenomenon is widely observed in LLMs and plays a critical role in preserving model behavior, especially in long contexts (Xiao et al., 2023; Han et al., 2024; Yang et al., 2024). The sink is usually the first token of a sequence, i.e. the $\langle \texttt{bos} \rangle$ beginning-of-sequence token (Xiao et al., 2023; Cancedda, 2024), and has also been linked to massive activations observed across LLMs (Sun et al., 2024). Gu et al. (2025) systematically investigate when and how attention sinks emerge during pretraining, and Barbero et al. (2025) provide theoretical and empirical evidence that attention sinks prevent over-mixing of information, to avoid representation collapse (Barbero et al., 2024a). In this work we relate sink tokens to RoPE and attention geometry, pinpointing the sink token as the failure mechanism for transformers in long contexts.

3 Analysis

We perform a geometric analysis of attention with RoPE, showing that keys and queries cluster tightly in opposing directions, while RoPE inhibits this behavior, with clusters dispersing and overlapping over time. Alongside small sink token ℓ_2 norm, these separated clusters produce a learned bias toward the sink. However, as RoPE disperses and overlaps key and query clusters, this mechanism becomes tenuous. We claim that the breakdown of transformers in long contexts is a breakdown of the sink token: past the training length, models begin attending to the wrong token(s) by default.

Analysis is conducted on Llama3-8B-Instruct, with additional trials on Olmo-7b and Gemma-7b for verification. When not otherwise specified, the relevant model is Llama3. Input text is drawn from the Wikitext2 dataset (Merity et al., 2016). Further details are provided in A.4.

3.1 Key/Query Clustering

An intuitive understanding of the attention operation is that it functions as a soft nearest-neighbor lookup. A query is oriented in latent space to align with one or more contextually relevant key vectors, and the degree of alignment defines the mixing ratios for corresponding values. The curse of dimensionality ensures that random IID latent points are orthogonal by default, so directional alignment in high-dimensional space is difficult. Thus we can imagine that keys and queries form overlapping point clouds around the origin. Key/query matching is accomplished by high directional alignment: activated pairs should have large, positive dot products. Keys and non-matching queries, meanwhile, should be orthogonal, with small dot products, to keep retrieval discriminative.

This model of attention, while intuitive, is also wrong, at least for RoPE models. Instead of overlapping clouds on the origin, keys and queries form tight clusters away from the origin, with minimal overlap. Further, such clusters are generally *un*aligned directionally, with the origin sitting between the clusters. Fig. 2 shows the mean intra- and inter-cluster pairwise cosine distances (ℓ_2-normalized dot products) for keys and queries, averaged over layers and heads. Before RoPE, intra-cluster distances (key-key and query-query), bounded to ± 1, are generally close to 1. Key and query point clouds are situated within a tight arc - i.e. in small clusters displaced from the origin. At the same time, these clusters are largely aligned *against* each other, with negative mean key-query dot product in Fig. 2 (right). This paints an entirely different picture of attention: instead of overlapped point clouds, envision keys and queries in opposing quadrants. Queries avoid attending to most keys via negative dot products. If a query and key land on aligned quadrant boundaries, though, the resulting zero dot product exceeds the baseline and yields a large attention weight. Softmax shift-invariance ensures that this arrangement (negative baseline and orthogonal "aligned" pairs) produces identical mixing behavior to the original conception (orthogonal baseline and positive products for aligned pairs). Fig. 1 (left) illustrates the proposed geometry. In practice, key/query clusters are not *directly* across the origin (dot products in Fig. 2 (right) are negative but small), but the intuition holds.

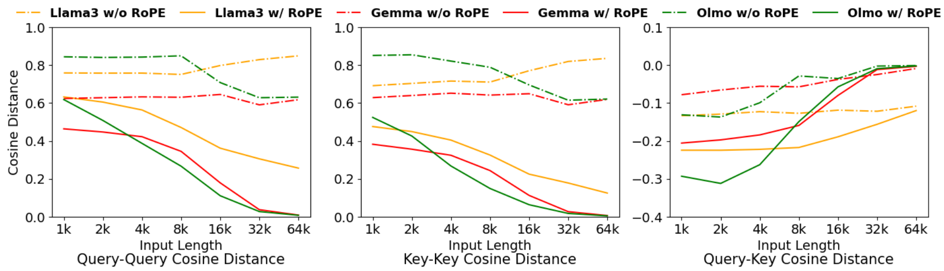


Figure 2: Effect of RoPE across context length on pairwise angular distances within heads for Llama3-8B, Gemma-7B and OLMo-7B.

3.1.1 The Impact of RoPE on Clustering

RoPE sends points through a predetermined trajectory of orbits around the origin, so any tightly-clustered point cloud, displaced from the origin, should inevitably disperse. This is indeed the case for transformers: Fig. 2 shows that RoPE decreases intra-cluster alignment, with further decrease over time as points spiral further away. The model compensates for this by positioning key/query clusters such that RoPE misaligns them *further*: key-query dot product *also* decreases, and does not rise until after the training length (2k for Olmo, 8k Llama/Gemma). Meanwhile, the clusters without RoPE mostly maintain their behavior across context lengths. It thus appears that RoPE weakens the clustering, but does not eliminate it until the training context length is exceeded.

Fig. 3 illustrates this visually, by taking a PCA "snapshot" of the point clouds without RoPE at time $t = 4096$, and applying the same projection with RoPE and at time $t = 65536$ (more plots are available in A.12). Prior observations are confirmed visually: in the first and third views, points form tight clusters displaced from the origin, and key and query clusters are located across from each other (with four queries per key cloud due to GQA (Ainslie et al., 2023)). RoPE causes clusters within training length to disperse slightly, but at length 64k eliminates cluster separation entirely (in this projection). The exact impact of this overlap in key/query clouds is discussed in § 3.2, but it is clear that query-key "alignment" works differently at length 64k compared to the other scenarios.

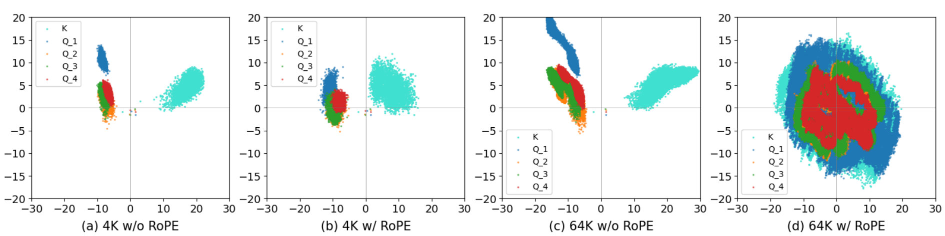


Figure 3: 2D PCA projections of Llama3 representations under different context lengths and RoPE settings (3rd key head of layer 21 and its queries). RoPE at long contexts destroys cluster separation.

3.1.2 A Singular Value Perspective on Clustering

While the visual analysis is striking, it only captures a 2D projection of a 128-dimensional latent space. Similarly, Fig. 2 reports pairwise relationships, an incomplete picture of global behavior. We therefore corroborate our findings with a holistic analysis based on singular values. In an attention head, the set of key or query vectors forms an $n \times d$ matrix, where n is sequence length and d is head width. Singular values of this matrix correspond to the principal components of the point cloud, an ordered set of directions maximizing variance along the earliest directions up to orthogonality constraints. When singular values are equal, variance is constant in all directions, and the point cloud forms a ball around the origin. Unequal values indicate uneven spread. In practice, the first singular value (FSV) of key and query clusters (before RoPE) is large, accounting for over 75% of total cluster variance on average for Llama3. Fig. 4 (left) plots the distribution across individual heads and layers, and the degree of variance covered by the first principal component ranges from about half to nearly all of it. Thus the majority of spread around the origin occurs in a single direction: either the cluster is a long thin needle, or the point cloud is displaced along this direction, and otherwise clustered tightly. Given the intra-cluster dot products in Fig. 2, the latter is the case.

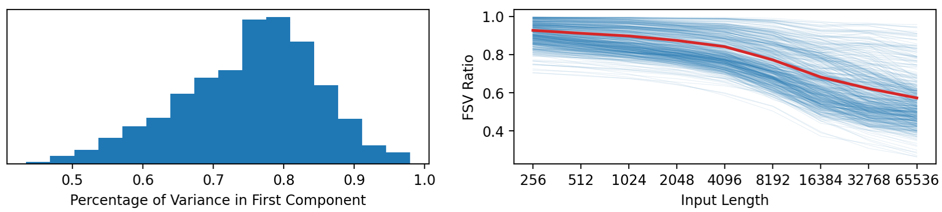


Figure 4: **Left**: Histogram across layers and heads showing the percentage of variance (relative to origin) explained by the first principal component of latent key/query clusters in Llama3. **Right**: Ratio of first singular value before and after RoPE, as a function of input length. Blue lines plot individual key/query heads, red plots the average trend. RoPE shrinks the first singular value, causing clusters to disperse, but impact accelerates beyond the training length.

When RoPE is applied, we expect principal components to skew more evenly: RoPE throws channel pairs through decorrelated rotations (ensured by irrational frequency ratios), so in the limit, a point cloud under RoPE maps to a shell of IID points orbiting the origin. In practice, this behavior does start to appear: FSV shrinks as sequence length increases [1]. Fig. 4 (right) shows the ratio of the FSV before and after RoPE. In all cases, the FSV decreases when RoPE is applied, but the decrease is limited for much of the training context, falling almost linearly up to length 4k (in log space, so really decaying exponentially). The decrease is more aggressive for inputs above the 8k training length. This implies that cluster means drift toward the origin as RoPE is applied over longer sequences.

At the same time, Proposition 1 (with proof in A.1) shows that RoPE preserves the sum of the point cloud matrix's squared singular values. This is relevant because, as the FSV falls due to RoPE, other singular values (representing other principal components) must grow to compensate. We thus demonstrate analytically that clusters expand and disperse as RoPE pulls them to the origin, exactly as depicted in Fig. 3.

Proposition 1. *Let $X \in \mathbb{R}^{n \times d}$ be a key or query matrix, with each row corresponding to a token. Then applying RoPE preserves the sum of squared singular values of X, i.e., $\sum_{i=1}^{\min\{n,d\}} \sigma_i(X)^2 = \sum_{i=1}^{\min\{n,d\}} \sigma_i(r(X))^2$, where $r(X)$ denotes the application of RoPE to X.*

3.2 Sink Tokens

Latent keys and queries cluster tightly into unaligned point clouds, and RoPE causes the point clouds to disperse and overlap over time, particularly when input length exceeds training length. But how does this produce out-of-distribution behavior in the attention mechanism itself, and for a transformer model as a whole? We claim that the effect of clustering is mediated by the sink token.

Prior work establishes that transformers attend heavily to the first token, regardless of input (Xiao et al., 2023). Prevailing wisdom is that this prevents over-mixing: information retrieval is not always useful, so attention heads must formulate a null operation (Barbero et al., 2025). Softmax normalizes attention scores to sum to 1, so heads cannot avoid attending. Instead, they "sink" attention into a placeholder key conveying no information – in practice the first token, typically a beginning-of-sequence indicator. When an attention head does perform meaningful information retrieval, it borrows weight from the sink token and reallocates it to the chosen key, as shown in A.2.

Under the intuitive overlapping-clouds model of latent keys/queries, sink token behavior is hard to reconcile. How can a single embedding align to *all* directions by default? Why not default to the current token – where keys and queries, projections of the same input, are easy to align – as in Mamba and other linear attention layers? (Gu & Dao, 2023; Katharopoulos et al., 2020) Our observations help to explain this behavior. Same-token key-query self-alignment is difficult to

[1]We account for the variable sequence length and its effect on singular value magnitude by computing the singular values as the eigenvalues of $\frac{XX^\top}{n}$, where the point cloud matrix $X \in \mathbb{R}^{n \times d}$.

consistently impose when keys and queries are distant. It is easier to assign a sink, and place its key near the origin, granting it a near-zero dot product with all queries. When average key-query product is negative, the sink becomes the most-attended by default. This is borne out in practice: key vectors for the first input token have unusually small ℓ_2 norm, as shown in Fig. 5 (left). Smaller norm produces larger dot products, as shown in Fig. 5 (right), which displays the average dot product of each key across subsequent queries, normalized by the largest product in the set. In this case, that's consistently the sink token, with recency bias responsible for the later upward trend.

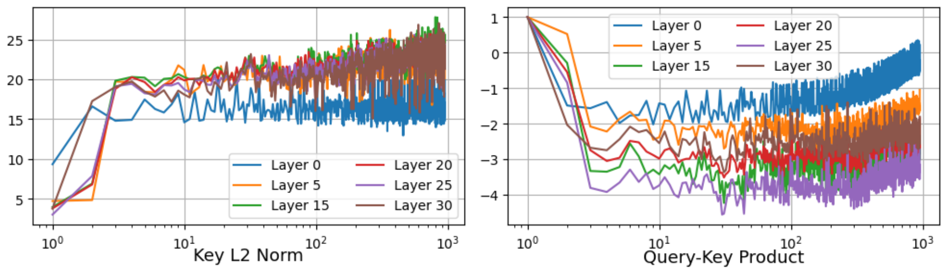


Figure 5: **Left:** Key ℓ_2 norm across layers over time. Sink token is consistently small. **Right:** Keys have low dot product against subsequent queries in expectation, except for the first and most recent tokens. Scores are normalized by the highest value, in this case always the sink.

3.2.1 The Impact of RoPE on Sink Tokens

Within the training context length, key and query clusters are separated to the point that a sink token with small key norm can absorb the majority of attention weight. Beyond the training length, however, RoPE causes clusters to disperse and overlap. When this happens, key and query points begin to obtain positive dot products. This stops the sink token from functioning, and we claim this is the cause of out-of-distribution behavior when transformers are exposed to long inputs.

Fig. 6 illustrates this behavior in Llama3. The left-hand plot captures the attention weight allocated to the sink token as a function of input length. The share of attention weight (with RoPE applied) varies widely but stably within the training length of 8k, but then falls sharply, decaying to zero over time as clusters progressively overlap and highly-aligned point pairs accumulate. Meanwhile, the activation of the sink token without RoPE stays roughly constant. We hypothesize that the activation starts off lower within the training length due to the observation in Fig. 2 (right): key/query clouds are more opposed *after* applying RoPE, lowering the average dot product relative to the sink. In any case, it is clear that RoPE causes catastrophic collapse of sink token weight for long inputs.

Fig. 6 (right) confirms that the decay in sink token attention weight is a function of key/query cluster overlap. As two point clouds approach the origin and disperse, the chance for high directional alignment between point pairs increases, and so the maximum dot pairwise product should increase as well. In practice, the maximum key/query dot product across all keys, per-query, does rise steadily over sequence length when RoPE is applied. Without RoPE, cluster behavior is stable over time, and so too, therefore, is the maximum degree of alignment between key and query points.

3.2.2 A Unified Theory of RoPE Attention

We now establish a unified geometric understanding of attention, RoPE, and sink tokens. The shift-invariance of softmax induces keys and queries to gather into opposing clouds across the origin. Sink tokens are implemented by positioning the first key near the origin, making that key's dot product with any query small. Because the clusters are opposed, average key/query dot product is negative, defaulting attention to the sink and mixing tokens only in the case of particularly aligned pairs.

RoPE complicates this arrangement by spinning points around and across the origin. Some channel pairs rotate much faster than others, but over time more and more channel pairs drift meaningfully from their original locations. Eventually all channels shift into orbit, transforming previously well-separated key and query clusters into dispersed, overlapping balls. This produces positive dot products between keys and queries, overwhelming the small, but previously dominant sink token logit. Transformers with RoPE fail on long inputs because they effectively lose access to the sink token, broadcasting an excess of information from the wrong tokens forward through time.

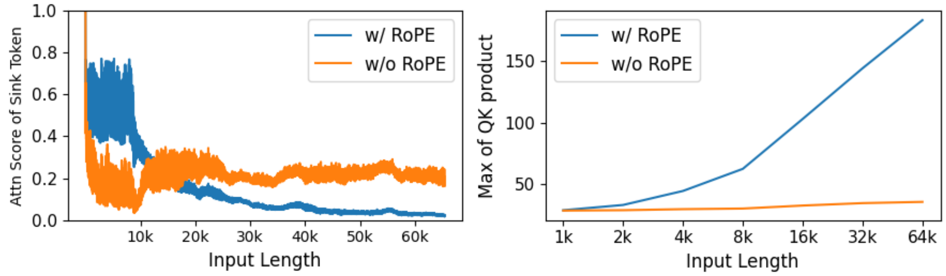


Figure 6: **Left:** Sink token attention weight vs. input length. With RoPE, sink-token attention decays to zero beyond the training length; without RoPE it remains stable. **Right:** Maximum Query–Key dot product vs. input length. The max QK product increases with length only when RoPE is active.

4 Method

Insight into RoPE's out-of-distribution behavior informs mitigation techniques. We want to encode relative position efficiently via rotation, but must still ensure that key and query point clouds remain separated through time, preserving sink token functionality. Many scaling techniques exist that (perhaps unintentionally) follow this heuristic. PI (Chen et al., 2023) and YaRN (Peng et al., 2023), for example, both limit the degree of drift in low-frequency channels to that seen during training. This maintains the separation of key and query clusters in those channels over extended contexts.

Training a new model from scratch offers greater design freedom. (Barbero et al., 2024b) suggest that information is better preserved when RoPE is limited to a fraction of channels, as positional and contextual information can be embedded in the RoPE subspace, while long-term semantic content can be allocated to the stable RoPE-free subspace. However, we still expect this approach to fail on extended contexts, as low-frequency channels still exist, reproducing the issues observed in § 3.

An obvious solution is to directly eliminate the low frequencies: raise the lowest value enough to complete a full cycle within the training length. This ensures that clusters "finish" drifting to the origin, as they devolve into uncorrelated rotations – overlapping shells around the origin – that the model can expect to persist indefinitely. Phrased differently, channel pairs with RoPE cannot rotate "out of distribution" if the entire rotation arc is covered. Liu et al. (2024) make this exact suggestion, but limit their evaluation to perplexity. Perplexity is improved by the newfound stability over long contexts, but it does not capture the loss of distant information through several cycles of uncorrelated rotation. Long-context information retrieval is likely difficult in this setting.

We hypothesize that *combining* high RoPE frequency and partial application is crucial for generalization to long contexts. Both changes are required for stable, discriminative behavior: key/query clusters fully merge in the RoPE channel subspace, but preserve sink token functionality via continued separation in RoPE-free subspace. We name this approach RoPE-ID (In Distribution). Fig. 7 (left) repeats the singular value analysis from Fig. 4 on a synthetic point cloud, for RoPE-ID and three baselines: RoPE with base frequency $\theta = 500k$, high-frequency RoPE with $\theta = 652$ (the lowest frequency to complete a cycle in 4k steps), and standard RoPE over half the channels. Base RoPE decays the FSV, pulling the cluster to the origin (and dispersing it) as input length increases. Applying RoPE to half the channels mitigates but does not resolve the issue: FSV still falls out of distribution after 4k tokens. High frequency RoPE dodges the issue by decaying within the training length to nearly zero, making further decay impossible. This produces stable behavior on long inputs, but preserves little information from the original embedding. Meanwhile, RoPE-ID lower-bounds FSV decay, maintaining cluster separation and sink token functionality by construction.

4.1 Implementation

We apply RoPE to half the channels of each attention head, and adjust RoPE frequencies to attain desired behaviors. RoPE frequencies interpolate exponentially between 1 and $\frac{1}{\theta}$, where θ is the base

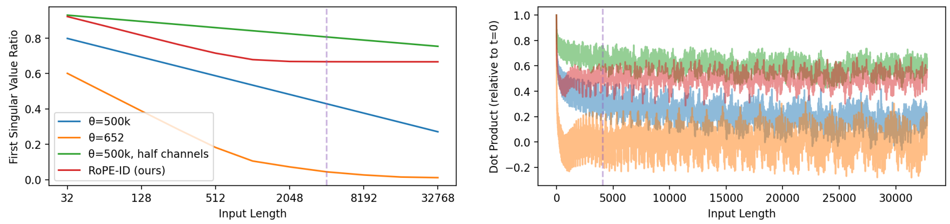


Figure 7: Expected RoPE behavior for our proposed method and three baselines. Dotted line indicates hypothetical training length of 4k and head dimension is 128. **Left:** Repeats the Fig. 4 singular value ratio before/after RoPE, for a synthetic point cloud of ones vectors. **Right:** Long-term RoPE decay for the same techniques, showing similar behaviors.

frequency hyperparameter. We adjust the low end of this scale to two full rotations per training length, as one rotation may still preserve correlation between low frequency channels (i.e. some decay still occurs after the purple line in the high frequency curve of Fig. 7 (left)). Maximum rotation speed is set to one cycle every 32 tokens, to better preserve information over short windows.

Taking a softmax over many IID logits increases the denominator but not the numerator, resulting in the mixture distribution becoming artificially smoother over time. We account for this via temperature scaling based on input length, borrowing from (Peng et al., 2023). Details, code and further discussion for RoPE-ID can be found in A.3. We evaluate both with and without temperature scale.

To evaluate RoPE-ID, we pretrain example 1B- and 3B-parameter decoders. Models use the Llama3 tokenizer and Dolma v1.7 dataset (Soldaini et al., 2024), reweighted per (Chu et al., 2024). Training proceeds over 21 billion tokens, with hyperparameter and architecture details provided in A.4.

5 Results

We compare RoPE-ID against four baselines. First is a vanilla decoder with RoPE base frequency $500k$, which we expect to fail beyond the training length. Second and third are approaches from § 4: increase RoPE frequency so that all channel pairs complete a rotation, or apply the original RoPE on half the channels per head. The former should yield stable, but poor, performance over context lengths, while the latter should mitigate but not prevent performance degradation. Tuning-free extension of the vanilla model using YaRN (Peng et al., 2023) with default hyperparameters acts as a state-of-the-art comparison. Across several benchmarks, our method is comparable to or outperforms all baselines, while generalizing gracefully out of the box, requiring no model adjustment.

Table 1: Average RULER benchmark scores by sequence length. Highest average score for each sequence length and model size is in **bold**; runner-up is underlined.

	Llama-1B			Llama-3B		
Method	4k	8k	16k	4k	8k	16k
RoPE	39.72	0.01	0.03	46.19	0.14	0.01
High frequency	16.04	7.60	2.37	28.02	14.31	5.14
HalfRoPE	**43.07**	0.14	0	**51.28**	0.4	0.03
YaRN	40.24	35.55	30.25	43.90	**45.09**	40.14
RoPE-ID	39.15	29.71	14.29	44.86	37.51	21.95
RoPE-ID (scaling)	39.15	**35.64**	**30.83**	44.86	43.39	**42.0**

RULER (Hsieh et al., 2024) measures long context performance on a number of synthetic tasks, such as needle-in-a-haystack retrieval and word counting. Table 1 shows that baselines perform as expected: RoPE and HalfRoPE perform well at 4k training length (with HalfRoPE even delivering a boost from its improved semantic encoding), but immediately fall to near-zero performance beyond that. High frequency RoPE degrades less in comparison, but also starts from a much lower score at 4k, as RoPE is scrambling stored information aggressively. YaRN delivers robust extrapolation to lengths 8k and 16k, without major penalty to baseline performance at 4k. Meanwhile, RoPE-ID with temperature scaling is comparable to YaRN with slight gains for longer sequences. It is marginally the strongest evaluated approach overall, but clearly surpasses RoPE-ID without temperature scaling, so we omit the non-scaled version from further analysis. A full breakdown of scores by task is provided in A.5.

LongBench (Bai et al., 2024) corroborates our RULER results, with reported averages over five task categories: single document question-answering, multi-document question-answering, few-shot learning, code completion and summarization. We exclude non-English tasks as our models are trained on English data. A full breakdown of scores is provided in A.6. Results in Table 2 mirror Table 1: RoPE and HalfRoPE drop immediately after 4k, with HalfRoPE delivering a small boost within 4k. High frequency performs stably but poorly, while YaRN is able to bring up performance for long inputs. Our method trails YaRN at 3B scale, but is superior at 1B. We conclude that RoPE-ID successfully generalizes to longer inputs out of the box.

Table 2: LongBench scores, averaged over 14 English tasks, by sequence length. Highest average score for each sequence length and model size is in **bold**; runner-up is <u>underlined</u>.

	Llama-1B			Llama-3B		
Method	4k	8k	16k	4k	8k	16k
RoPE	14.61	8.23	8.73	<u>18.62</u>	11.36	10.42
High frequency	11.8	11.44	11.04	14.19	13.82	13.78
HalfRoPE	<u>15.38</u>	8.73	8.86	**19.42**	10.7	10.62
YaRN	14.84	<u>14.54</u>	<u>14.09</u>	15.87	**19.29**	**19.63**
RoPE-ID (scaling)	**15.83**	**15.83**	**15.80**	15.92	<u>17.13</u>	<u>17.94</u>

Commonsense Reasoning tasks act as a sanity check in Table 3. Model scores are all similar: RoPE frequency and number of channels have little impact on expressivity within the training length. To the degree that scores differ, RoPE-ID is in the top-3 for all tasks and settings.

Table 3: Standard evaluation of common sense reasoning tasks

	Llama-1B				Llama-3B			
Method	ARC-C	HellaSwag	PIQA	**Avg.**	ARC-C	HellaSwag	PIQA	**Avg.**
RoPE	25.77	**44.00**	**69.26**	46.34	29.18	**53.95**	**72.74**	51.96
High frequency	25.17	42.99	**69.26**	45.81	29.61	53.32	71.87	51.60
HalfRoPE	**26.45**	**44.00**	68.77	**46.41**	**32.17**	53.87	72.03	**52.69**
YaRN	25.60	41.89	68.61	45.37	29.10	52.46	72.20	51.25
RoPE-ID	25.60	43.58	68.88	46.02	30.03	**53.95**	72.25	52.08

Reapplying Analysis from § 3.1 confirms that our trained models obey our geometric framework. We recreate Fig. 2 and 3 for baseline and RoPE-ID models in A.7. Our 1B example decoder mirrors the behavior of established LLMs, while RoPE-ID behaves as expected, delivering stable, and stably separated, key and query clusters over time.

6 Discussion

From empirical analysis we develop a unified understanding of attention geometry and long-context failure modes. Keys and queries form tight clusters in opposing directions, allowing sink tokens to absorb attention weight by default via small ℓ_2 norm. RoPE inhibits this behavior by merging and dispersing the point clouds, particularly beyond the training length. Overlapped point clouds inflate key/query alignment, preventing the sink token from functioning. From this understanding, we produce stable model behavior by applying RoPE with high frequency to a fraction of channels, and demonstrate strong long-context performance on downstream tasks out of the box.

Beyond RoPE-ID, other approaches based on this analysis are possible (e.g., applying high frequency RoPE to a fraction of *heads*, or manually injected sink tokens as in Hymba (Dong et al., 2025)). It is also possible to combine RoPE-ID with inference-time model adjustments. We leave this, as well as long-context fine-tuning of RoPE-ID models, to future work.

Reproducibility: Code for RoPE-ID is provided in A.3 with training details in A.4. Empirical analysis techniques are straightforward, and evaluation uses standard benchmarks.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2024. URL `https://arxiv.org/abs/2308.14508`.

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João G. M. Araújo, Alex Vitvitskyi, Razvan Pascanu, and Petar Veličković. Transformers need glasses! information over-squashing in language tasks, 2024a. URL `https://arxiv.org/abs/2406.04267`.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? *arXiv preprint arXiv:2410.06205*, 2024b.

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*, 2025.

bloc97. Tk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation., 2023a. URL `https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/`.

bloc97. Add ntk-aware interpolation ”by parts” correction, 2023b. URL `https://github.com/jquesnelle/scaled-rope/pull/1`.

Nicola Cancedda. Spectral filters, dark signals, and attention sinks, 2024. URL `https://arxiv.org/abs/2402.09221`.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023. URL `https://arxiv.org/abs/2306.15595`.

Yuhan Chen, Ang Lv, Jian Luan, Bin Wang, and Wei Liu. Hope: A novel positional encoding without long-term decay for enhanced context awareness and extrapolation, 2024. URL `https://arxiv.org/abs/2410.21216`.

Linsong Chu, Divya Kumari, Tri Dao, Albert Gu, Raghu Ganti, Dakshi Agrawal, Mudhakar Srivatsa, Davis Wertheimer, Yu Chin Fabian Lim, Antoni Viros, Nelson Gonzalez, Tuan HoangTrong, Ofir Arviv, Yotam Perlitz, Michal Shmueli, Haochen Shen, Minjia Zhang, Gabe Goodhart, Naigang Wang, Nick Hill, Joshua Rosenkranz, Chi-Chun Liu, Adnan Hoque, Chih-Chieh Yang, Sukriti Sharma, Anh Uong, Jay Gala, Syed Zawad, and Ryan Gordon. Bamba: Inference-efficient hybrid mamba2 model. `https://huggingface.co/blog/bamba#bamba-inference-efficient-hybrid-mamba2-model-%F0%9F%90%8D`, December 2024. Accessed: 2025-05-06.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kai ge Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, ZIJIA CHEN, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. Hymba: A hybrid-head architecture for small language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=A1ztozypga.

Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan

Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj

Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view, 2025.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models, 2024. URL https://arxiv.org/abs/2308.16137.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Xiaoran Liu, Hang Yan, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of roPE-based extrapolation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JO7k0SJ5V6.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models, 2023. URL https://arxiv.org/abs/2309.00071.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022. URL https://arxiv.org/abs/2108.12409.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models, 2024. URL https://arxiv.org/abs/2402.17762.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Shuai Yang, Yuying Ge, Yang Li, Yukang Chen, Yixiao Ge, Ying Shan, and Yingcong Chen. Seed-story: Multimodal long story generation with large language model, 2024. URL https://arxiv.org/abs/2407.08683.

A APPENDIX

A.1 PROOF OF PROPOSITION 1

Proof. First, a standard result in linear algebra (Golub & Van Loan, 2013) shows that

$$\sum_{i=1}^{\min\{n,d\}} \sigma_i(X)^2 = \|X\|_F^2 := \sum_{i=1}^{n}\sum_{j=1}^{d} X_{i,j}^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Similarly,

$$\sum_{i=1}^{\min\{n,d\}} \sigma_i(r(X))^2 = \|r(X)\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{d} r(X)_{i,j}^2.$$

Therefore, it suffices to show that for any fixed token index i, $\sum_{j=1}^{d} X_{i,j}^2 = \sum_{j=1}^{d} r(X)_{i,j}^2$. By the definition of RoPE, $\forall 1 \le l \le d/2$, we have

$$[r(X)_{i,2l-1}, r(X)_{i,2l}] = [\cos(i\theta_l)X_{i,2l-1} - \sin(i\theta_l)X_{i,2l}, \sin(i\theta_l)X_{i,2l-1} + \cos(i\theta_l)X_{i,2l}],$$

and it is easy to see that $r(X)_{i,2l-1}^2 + r(X)_{i,2l}^2 = X_{i,2l-1}^2 + X_{i,2l}^2$. Finally,

$$\sum_{j=1}^{d} X_{i,j}^2 = \sum_{l=1}^{d/2} X_{i,2l-1}^2 + X_{i,2l}^2 = \sum_{l=1}^{d/2} r(X)_{i,2l-1}^2 + r(X)_{i,2l}^2 = \sum_{j=1}^{d} r(X)_{i,j}^2,$$

which completes the proof. □

A.2 ATTENTION MAP AND SINK TOKEN VISUALIZATION

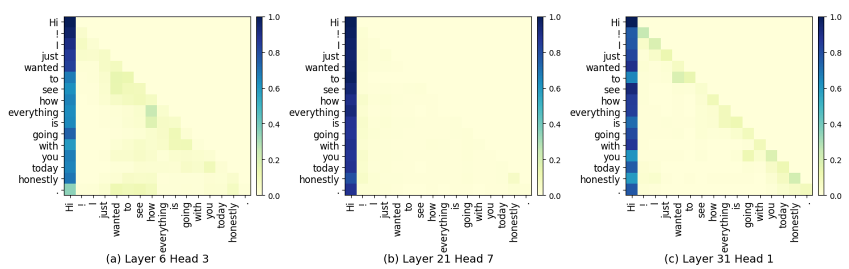


Figure 8: Attention patterns of three heads in LLaMA3-8B. Sink token behavior is clearly observed, even when performing non-trivial token mixing.

A.3 IMPLEMENTATION DETAILS

In our approach we apply RoPE to half the channels of each attention head, and adjust the RoPE frequencies to attain desired behaviors. Standard RoPE frequencies interpolate exponentially between 1 and $\frac{1}{\theta}$, where θ is the base frequency hyperparameter, typically $10k$ or $500k$. We adjust both endpoints of this interpolated scale. First, we must ensure that all frequencies are high enough to complete at least one rotation within our training length of 4k tokens. We set two full rotations as the minimum, as one rotation may not be sufficient to fully eliminate correlation between low frequency channels (i.e. some decay still occurs in the high frequency curve of Fig. 7 (Left), even after the slowest channel pair finishes a full rotation at the purple line). We therefore update the minimum frequency scale value from $\frac{1}{\theta}$ to $\frac{4\pi}{4096}$. Second, we also pull the maximum frequency scale of 1 toward a more conservative value. Since we apply RoPE to only a fraction of available channels, it is important that the channels be discriminative. RoPE's fastest channel pair completes a cycle in $2\pi \approx 6$ tokens, after which information is effectively lost, as it becomes impossible to disentangle relative position modulo 2π from content. An effective 6-token window is very aggressive, so we pull back the max frequency to $\frac{2\pi}{32}$, completing a cycle in 32 tokens.

While applying high-frequency RoPE to a fraction of channels ensures stable clustering and sink token behavior for long inputs, out-of-distribution behavior can still occur via the softmax activation. Key/query dot products are stable over time by construction, so taking a softmax over an increasing number of IID key/query pairs will increase the softmax denominator, without a corresponding increase to the numerator. The result is the mixture distribution becoming smoother than expected over time. We account for this by introducing temperature scaling based on input length, borrowing from (Peng et al., 2023). The adjustment is $(1+0.1*\ln(\min(4096,n)))^2$, where 4096 is the training length and n is the given input length.

Example code for this approach in HuggingFace Transformers is provided below.

Codeblock 1: Modification of scaling factor within the attention_interface

```
# src/transformers/models/llama/modeling_llama.py

class LlamaAttention(nn.Module):
    ...
    def forward(...):
        ...
        attn_output, attn_weights = attention_interface(
            self,
            query_states,
            key_states,
            value_states,
            attention_mask,
            dropout=0.0 if not self.training else self.attention_dropout,
            scaling=self.scaling \
            * (0.1 * math.log(max(current_position, 4096) / 4096) + 1)**2,
            **kwargs,
        )
```

Codeblock 2: Modification of modeling_rope_utils with our method

```
# src/transformers/modeling_rope_utils.py

ROPE_INIT_FUNCTIONS = {
    ...
    "ourmethod": _compute_our_method_parameters,
}

def _compute_our_method_parameters(
    config, device, seq_len = None, **rope_kwargs
):
    if config is not None and len(rope_kwargs) > 0:
        raise ValueError(...)
    if len(rope_kwargs) > 0:
        base = rope_kwargs["base"]
        dim = rope_kwargs["dim"]
    elif config is not None:
        base = config.rope_theta
        partial_rotary_factor = config.partial_rotary_factor \
            if hasattr(config, "partial_rotary_factor") else 1.0
        head_dim = getattr(config, "head_dim", None) \
            or config.hidden_size // config.num_attention_heads
        dim = int(head_dim * partial_rotary_factor)

    attention_factor = 1.0  # Unused in this type of RoPE

    logstart = math.log(2 * math.pi / base)  # 1 cycle in ratio steps
    logend = math.log(4 * math.pi / 4096)  # 2 cycles in 4k steps
    pos = torch.arange(0, dim // 2, device=device) / (dim // 2 - 1)
    logfreq = pos * (logend - logstart) + logstart
    inv_freq = logfreq.exp()
    return inv_freq, attention_factor
```

Codeblock 3: Applying RoPE to only half the channels

```
# src/transformers/models/llama/modeling_llama.py

def apply_rotary_pos_emb(q, k, cos, sin, position_ids=None, unsqueeze_dim=1):
    cos = cos.unsqueeze(unsqueeze_dim)
    sin = sin.unsqueeze(unsqueeze_dim)

    q_quartile_size = q.shape[-1] // 4
    q1, q2, q3, q4 = torch.split(q, \
        split_size_or_sections=q_quartile_size, dim=-1)
    k_quartile_size = k.shape[-1] // 4
    k1, k2, k3, k4 = torch.split(k, \
        split_size_or_sections=k_quartile_size, dim=-1)

    q_rot = torch.cat((q1, q3), dim=-1)
    k_rot = torch.cat((k1, k3), dim=-1)

    q_rot_embed = (q_rot * cos) + (rotate_half(q_rot) * sin)
    k_rot_embed = (k_rot * cos) + (rotate_half(k_rot) * sin)

    q1_updated, q3_updated = torch.split(q_rot_embed, \
        split_size_or_sections=q_quartile_size, dim=-1)
    k1_updated, k3_updated = torch.split(k_rot_embed, \
        split_size_or_sections=k_quartile_size, dim=-1)

    q_embed = torch.cat((q1_updated, q2, q3_updated, q4), dim=-1)
    k_embed = torch.cat((k1_updated, k2, k3_updated, k4), dim=-1)


    return q_embed, k_embed
```

A.4 Training, Model, and Evaluation Details

Evaluation: During evaluation, we take care to avoid inducing out-of-distribution behavior not related to extended context length. In particular, we report point cloud behaviors "with" and "without" RoPE. Both cases are drawn from the same single forward pass from a given model, with the model unaltered and RoPE applied. Strictly speaking, these point clouds come from *after* and *before* the application of RoPE, respectively. This keeps observations within-distribution, even when discussing un-RoPEd point clouds inside of a RoPE-using model. Performing the actual attention without RoPE would cascade errors through the model.

Training: Model training proceeds over 21 billion tokens, with a context length of 4096 and half a million tokens per minibatch. All models are trained across 16 NVIDIA A100s in parallel. We employ a learning rate of $3e-4$, with warmup over 2k steps and cosine decay. Optimizer is AdamW with $\lambda = (.9, .95)$ and weight decay 0.1. Model architecture follows Llama3, with details provided in Table 4.

Table 4: Model architectures used for pretraining and evaluation

Parameters	1B	3B
Vocab	128256	128256
Width	1280	2048
Layers	32	48
Heads	16	16
KV heads	4	4
Head dim	80	128
Inner dim	4096	7168

A.5 RULER BENCHMARK

Table 5: Performance of different methods on the RULER benchmark. Highest average score for each sequence length and model size is in **bold**; runner-up is <u>underlined</u>.

Method	Seq.	N-S1	N-S2	N-S3	N-MK1	N-MK2	N-MK3	N-MV	N-MQ	VT	CWE	FWE	QA-1	QA-2	Avg.
1B Models															
	4k	100	100	96.6	73.6	2.4	3.4	23.45	24.65	10.24	21.76	27.4	22.67	10.24	39.72
RoPE	8k	0	0	0	0	0	0	0	0	0	0	0	0.13	0	0.01
	16k	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0.03
	4k	42	44.8	35	37.4	0.2	3.6	7.05	4.85	0	0.1	1	16.95	15.6	16.04
High Frequency	8k	18.8	15.6	15	19.4	0	0.4	6.35	1.2	0	0.04	2.87	6.13	13	7.60
	16k	9	3	1	3	0	0	1.2	0.1	0	0	0.2	4.73	8.6	2.37
	4k	100	100	96	76.4	22	17.2	12.2	10.95	3.88	48.72	27.73	22.42	22.4	**43.07**
HalfRoPE	8k	0	0	0	0	0	0	0	0	0	0	0.07	1.6	0.2	0.14
	16k	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4k	100	100	96.6	73.6	2.4	3.4	23.45	24.65	10.24	21.76	27.4	22.67	17	<u>40.24</u>
YaRN	8k	97.6	92.2	92.4	62	1.6	1.8	22.2	23.1	21.8	2.8	17.07	11.57	16	<u>35.55</u>
	16k	99.8	85.8	68	44.2	1.6	1.4	20.35	17.7	18.24	0.34	12	11.07	12.8	<u>30.25</u>
	4k	100	100	85.6	62	3	2.6	29.7	30.85	0.52	12.12	35.4	27.52	19.6	39.15
RoPE-ID	8k	100	100	33.4	40.4	0.2	0.6	21.8	20.55	11.44	2.44	29.33	10.28	15.8	29.71
	16k	96.4	19.4	1.2	14.2	0	0	9.65	2.1	2.36	0	20.33	8.73	11.4	14.29
	4k	100	100	85.6	62	3	2.6	29.7	30.85	0.52	12.12	35.4	27.52	19.6	39.15
RoPE-ID (Scaling)	8k	100	100	70.6	50.8	1.4	1.2	26.3	34.55	12.8	6.16	33.13	11.02	15.4	**35.64**
	16k	100	98.6	37	40.8	1.2	0.4	29.95	23.5	6.64	0.38	32.13	15.82	14.4	**30.83**
3B Models															
	4k	100	100	93.4	56	7.2	3.8	39.55	53.55	12.96	41.28	36.07	30.25	26.4	<u>46.19</u>
RoPE	8k	0	0	0	0	0	0	0	0	0	0	0.07	1.8	0	0.14
	16k	0	0	0	0	0	0	0	0	0	0	0	0.13	0	0.01
	4k	59.8	57.4	47.8	26.89	40.6	18.4	18.4	17.6	0	13.9	13.2	29.53	20.8	28.02
High Frequency	8k	27.8	24.4	25.6	23.8	1.8	4.8	17.45	15.4	0	10.72	8.2	9.4	16.6	14.31
	16k	8.8	9.2	5.2	7.8	0	0.4	7.65	2.05	0	1.58	5.53	7.27	11.4	5.14
	4k	100	100	99.6	85	5.6	4.6	58.9	55.7	15.8	44	38.4	33.4	25.6	**51.28**
HalfRoPE	8k	0	0	0	0	0	0	0	0	0	0.82	3.2	0.93	0.2	0.40
	16k	0	0	0	0	0	0	0	0	0	0	0	0.13	0.2	0.03
	4k	100	98.2	91.8	57.2	9.8	4.8	64.3	38.6	20.88	14.66	21.27	28	21.2	43.90
YaRN	8k	100	100	98.6	71	1.8	4.4	42.25	60.25	20.32	13.94	29.8	17.18	26.6	**45.09**
	16k	100	99.8	92.6	63.4	7.2	1.4	34.55	47.65	11.76	8.66	18.8	15.82	20.2	<u>40.14</u>
	4k	100	100	97.8	77.4	9.4	8.4	28.2	37.25	12.84	13.18	42.27	30.88	25.6	44.86
RoPE-ID	8k	100	99	88.6	45.4	0.6	1.8	23.7	36.5	13.4	6.54	36.4	14.28	21.4	37.51
	16k	73.6	73.4	13.2	32.2	0	0.2	8.25	8.4	18.72	3.72	26.8	12.87	14	21.95
	4k	100	100	97.8	77.4	9.4	8.4	28.2	37.25	12.84	13.18	42.27	30.88	25.6	44.86
RoPE-ID (Scaling)	8k	100	100	96	55.8	6.6	3.4	43.4	48.9	20.6	12.36	37	15.62	24.4	<u>43.39</u>
	16k	90.8	99.2	86	59.6	7.2	0.4	48	42.4	36.56	10.24	31.07	17.07	17.4	**42.0**

A.6 LongBench Benchmark

14 English Tasks for 5 Different Categories used for evaluation:

Single Document QA: NarrativeQA, Qasper, and MultiFieldQA-en
Multi-Document QA: 2WikiMultihopQA, HotpotQA, MuSiQue
Summarization: GovReport, MultiNews, QMSum
Few-shot Learning: SAMSum, TREC, TriviaQA
Code Completion: LCC, RepoBench-P

Table 6: Performance of different methods on LongBench, averaged by task type. Highest total average score for each sequence length and model size is in **bold**; runner-up is underlined.

Methods	Seq.	Single-Doc QA	Multi-Doc QA	Summarization	Few-Shot Learning	Code	Avg.
1B Models							
RoPE	4k	6.65	4.32	14.22	27.27	23.55	14.61
	8k	4.53	1.69	11.48	7.55	19.75	8.23
	16k	4.84	2.08	12.74	8.19	19.33	8.73
High Frequency	4k	5.26	4.22	12.97	19.3	19.96	11.8
	8k	5.31	4.22	12.61	17.61	20.47	11.44
	16k	5.2	3.92	12.49	16.71	19.77	11.04
HalfRoPE	4k	6.94	4.72	16.07	28.73	22.99	<u>15.38</u>
	8k	4.96	1.94	15.89	6.78	16.77	8.73
	16k	5.25	2	16.17	6.5	17.17	8.86
YaRN	4k	6.86	4.35	14.94	28.07	22.54	14.84
	8k	6.83	4.77	15.29	26.76	21.33	<u>14.54</u>
	16k	6.78	5.18	15.26	25.59	19.42	<u>14.09</u>
RoPE-ID (scaling)	4k	6.71	4.80	14.61	31.68	24.11	**15.83**
	8k	7.18	5.23	14.69	31.39	23.11	**15.83**
	16k	7.01	5.35	15.40	30.55	23.12	**15.80**
3B Models							
RoPE	4k	7.32	4.81	14.79	35.13	37.28	<u>18.62</u>
	8k	5.19	2.57	13	11.03	31.84	11.36
	16k	5.12	2.51	11.67	11.22	27.15	10.42
High Frequency	4k	5.42	4.52	12.61	25.85	26.69	14.19
	8k	5.82	4.6	12.64	23.06	27.58	13.82
	16k	5.93	4.51	12.51	22.7	28.02	13.78
HalfRoPE	4k	7.75	4.81	17.14	40.68	30.4	**19.42**
	8k	5.13	2.23	16.81	9.84	23.89	10.7
	16k	4.99	2.23	16.11	10.87	23.03	10.62
YaRN	4k	6.5	4.96	15.2	29.74	26.50	15.87
	8k	7.96	5.46	15.57	40.30	31.12	**19.29**
	16k	8	5.94	16.45	42	28.81	**19.63**
RoPE-ID (scaling)	4k	7.71	5.23	13.51	32.83	22.50	15.92
	8k	8.52	5.65	14.55	36.37	22.24	<u>17.13</u>
	16k	8.96	6.23	15.90	37.93	22.03	<u>17.94</u>

A.7 Repeated Analysis for Trained Models

We repeat our original analysis on our trained 1B models (baseline and RoPE-ID). Our baseline model exhibits the same behavior observed in state of the art LLMs, while our RoPE-ID model exhibits the desired stable behavior and consistent clustering across sequence lengths.

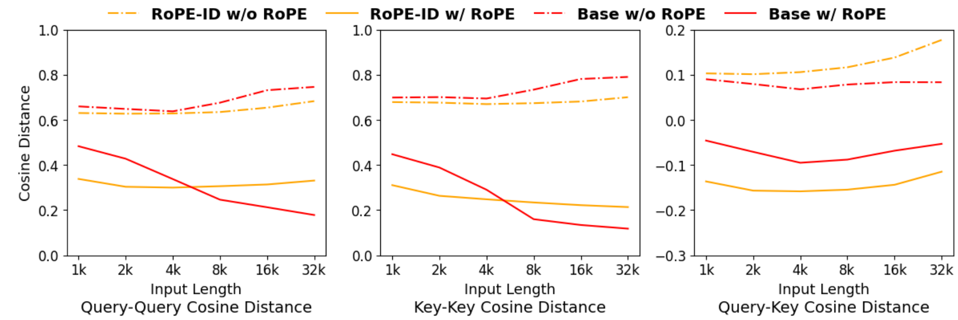


Figure 9: Pairwise angular distances within- and between-clusters for our trained models. Base model matches prior LLM observations, including an inflection point for key-query product with RoPE at the training length (4k). RoPE-ID maintains stable behavior over time.

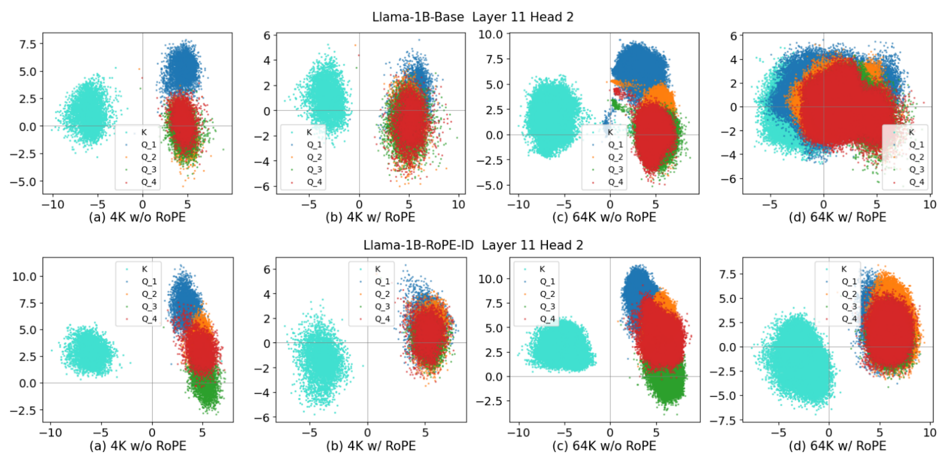


Figure 10: Clustering behavior for our trained models. Vanilla transformer (top) matches prior LLM observations; RoPE-ID (bottom) maintains stable cluster separation.

A.8 RoPE-ID HyperParameter Ablations

We perform additional ablations on the hyperparameters of RoPE-ID, namely the high and low frequencies, the number of channels with RoPE applied, and the degree of temperature scaling. We train several additional 1B demo models using the same training procedure, and report average scores from the RULER benchmark.

We observe a consistent general trend: reducing the number of high-frequency channels improves performance, especially on shorter contexts. However, the model eventually reaches a threshold beyond which length generalization drops sharply. We hypothesize that without enough high-frequency channels, the model instead learns to encode position based on learned patterns of latent drift, which do not generalize to longer contexts. This explains the patterns observed in Table 8 and Fig. 11, where decreasing the number of RoPE channels, and increasing the wavelength of the highest frequency, gradually improves performance, until triggering a catastrophic collapse at longer contexts. We conclude that our hyperparameter choices for channel fraction and highest frequency (50% of channels, shortest wavelength 32) represent a safe middle-ground.

Halving the wavelength of the lowest frequency is highly beneficial, showing that one period is indeed not sufficient to decorrelate all rotating channels. This aligns with Fig. 7, where the orange curve (RoPE with high enough frequency to complete one period) still performs some FSV decay beyond the training length. Meanwhile, RoPE-ID, the red curve, holds the FSV ratio constant beyond the training length.

The impact of temperature scaling, shown in Table 8, is minimal. Here we raise and lower the exponent of the YaRN temperature scaling formula, and find that the default value of 2 works fine.

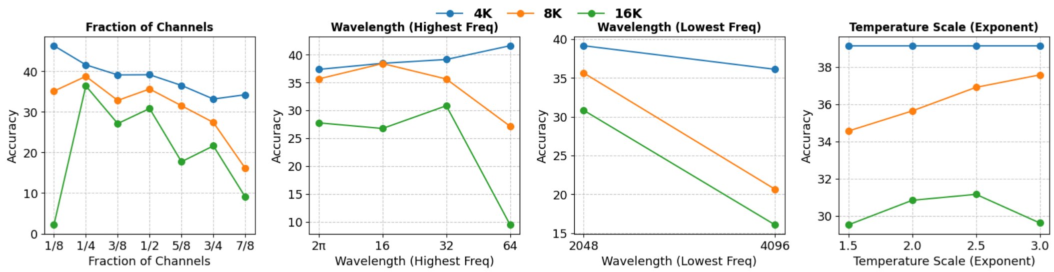


Figure 11: Llama 1B ablation studies with average RULER scores on the y-axis. The x-axis of the 4 plots cover fraction of channels, wavelength (high frequency), wavelength (low frequency) and temperature scaling, respectively. The color coded legend represents context lengths 4k, 8k and 16k.

Table 7: RULER scores for LlaMA 1B models up to 128k context length, where experiments cover tuning with YaRN, RoPE-ID, and RoPE-ID + YaRN.

Llama-1B Models			
Fine-Tuning	**YaRN**	**RoPE-ID**	**RoPE-ID + YaRN**
4K	43.29	39.49	47.09
8K	39.27	35.81	41.94
16K	34.55	30.63	38.59
32K	32.74	30.66	36.16
64K	21.90	27.34	31.48
128K	12.25	19.78	29.23

A.9 Long-Context Fine-Tuning with RoPE-ID

While our analysis focuses on RoPE-ID as a tuning-free approach to length generalization, we can also combine it with fine-tuning to further extend the effective context length. Here we take our trained Llama 1B models and tune them to 128k context length in stages: first, we load the 4k model checkpoint and apply any relevant RoPE frequency scaling. Then, we continue pretraining for 5k steps, with sequence length increased to 32k. The total tokens per batch is held constant at 500k, and learning rate warms up over 250 steps until it reaches the final LR of the previous checkpoint ($3e-5$), where it is held constant. We then repeat the process for another 5k steps, going from 32k sequence length to 128k.

We extend three models in this fashion: first, we apply YaRN scaling to the baseline RoPE model during each jump in sequence length. Second, we tune the RoPE-ID model with no adjustment to RoPE frequencies. Third, we tune the RoPE-ID model, but with YaRN-style scaling also applied. YaRN cannot be applied directly to RoPE-ID models as the default YaRN hyperparameters do not work for such high frequencies. We therefore set the scaling thresholds to the highest and lowest frequencies and interpolate in-between (the highest frequency is unchanged, and the lowest frequency scales up by L'/L, where L', L are the new and old sequence lengths, respectively).

Results for RULER are given in Table 7 and Fig. 12. The tuned RoPE-ID model exhibits better length generalization at context length 64k and above. Notably, the combination of RoPE-ID and YaRN-style scaling achieves superior performance at all context lengths, compared to either method alone.

A.10 Additional Clustering Analysis

Here we repeat the analysis performed in Fig. 2 for other distance and clustering metrics. Fig. 13 shows inter- and intra-cluster alignment as measured by dot product rather than cosine distance, better reflecting the actual attention logits. This introduces noise, as embedding norms can shift over time without affecting clustering behavior, and cosine distance is norm-invariant whereas dot

Table 8: RULER scores for LlaMA 1B ablations as function of fraction of channels, wavelength, and temperature scaling

Llama-1B Models			
Fraction of Channels	**4K**	**8K**	**16K**
1/8	46.30	35.12	2.18
1/4	41.59	38.75	36.41
3/8	39.11	32.83	27.12
1/2	39.15	35.64	30.83
5/8	36.52	31.52	17.72
3/4	33.17	27.40	21.63
7/8	34.23	16.11	9.05
Wavelength (Highest Freq)	**4K**	**8K**	**16K**
2π	37.37	35.66	27.77
16	38.46	36.52	26.76
32	39.15	35.64	30.84
64	41.62	27.16	9.46
Wavelength (Lowest Freq)	**4K**	**8K**	**16K**
2048	39.15	35.64	30.83
4096	36.11	20.66	16.07
Temperature Scale (Exponent)	**4K**	**8K**	**16K**
1.5	39.15	34.56	29.51
2.0	39.15	35.64	30.83
2.5	39.15	36.91	31.15
3.0	39.15	37.58	29.61

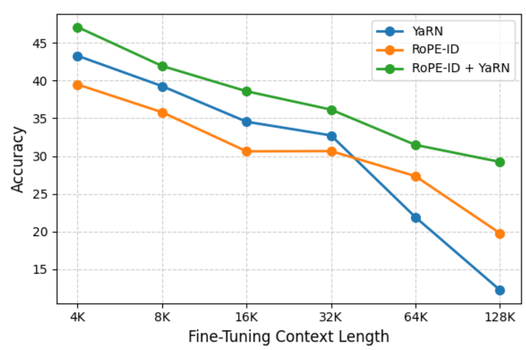


Figure 12: A plot of average RULER scores from Table 7 better visualizing overall trends. Y-axis represents average RULER scores and x-axis is context length.

products are not. This also introduces large differences between different models. Nevertheless, overall trends are roughly the same.

Fig. 14 directly quantifies the degree of clustering, using Silhouette Score (left) and Davies-Bouldin Index (right) in Llama3-8B. Results again mirror Fig. 2: clustering is consistent across sequence lengths prior to RoPE (and even increases beyond the training length), but falls over time once RoPE is applied (Silhouette Score decreases, while Davies-Bouldin Index increases as sequences become long). We conclude that clusters are behaving as described in the main paper.

Fig. 15 shows the correlation between Geometric Structure and Stable Rank. To further investigate the impact of RoPE, we visualize the PCA projections of Key and Query states, explicitly annotated

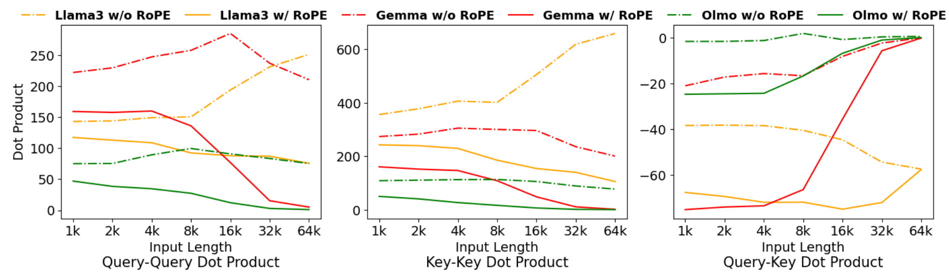


Figure 13: Mean of the pairwise dot product of query and key values across all heads for Llama3-8B, Gemma-7B and OLMo-7B which showcases the effect of RoPE across various context lengths showing similar trend as Fig. 2

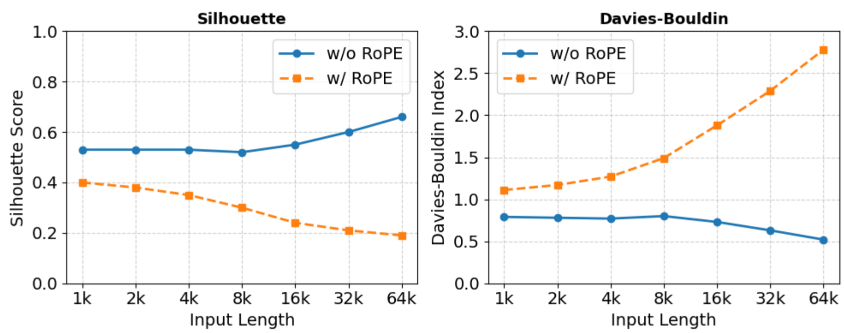


Figure 14: Silhouette Score (left) and Davies-Bouldin Index (right) showcasing the effect of RoPE on Internal Representation Clustering for Llama3-8B-Instruct. Lower Silhouette Score and higher Davies-Bouldin Index represents more overlap.

with their corresponding stable ranks. Crucially, the quantitative analysis is consistent with the visual phenomenon: the stable rank precipitates a sharp rise (from ~ 1.1 to ~ 7.1). The simultaneity of cluster dispersion and rank inflation provides strong evidence that RoPE acts as a destabilizing factor, destroying the intrinsic low-dimensional geometric structure of the Key/Query states during inference extrapolation.

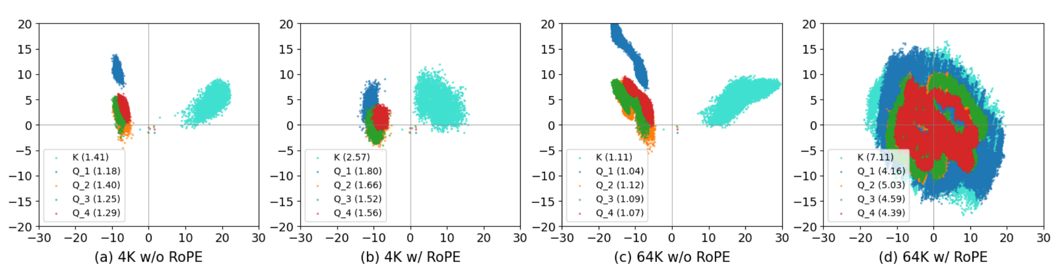


Figure 15: 2D PCA projections of Llama3 representations and their Stable Rank under different context lengths and RoPE settings (same as Figure 3). RoPE at long contexts destroys cluster separation and significantly inflates the stable rank.

Figure 16 presents the stable rank of the K matrices, averaged across all layers of Llama3-8B-Instruct, with context lengths varying from 1k to 64k. The error bars indicate the standard deviation across layers. When RoPE is applied (blue line), the stable rank increases monotonically as the context length extends. Beyond the training length (8k), the stable rank is significantly higher than the baseline.

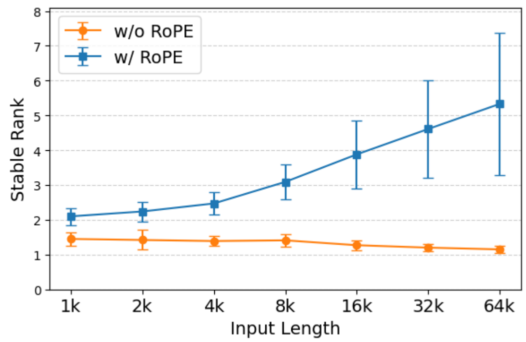


Figure 16: The stable rank of Key states across varying input lengths (1k to 64k). The plot compares the stable rank with and without RoPE. We report the mean stable rank of Key across all layers. Error bars denote standard deviation.

A.11 Additional Theoretical Analysis: Stable Rank

Let us first fix the notations: $\mathcal{R}(\boldsymbol{X})$ denotes the application of RoPE to a key/query matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times d}$. Specifically, $\mathcal{R}(\boldsymbol{X}) := [\boldsymbol{R}_1\boldsymbol{x}_1, \dots, \boldsymbol{R}_n\boldsymbol{x}_n]^\top \in \mathbb{R}^{n\times d}$, where $\boldsymbol{R}_j = \mathrm{diag}(\boldsymbol{R}_{j,\theta_1}, \dots, \boldsymbol{R}_{j,\theta_{d/2}}) \in \mathbb{R}^{d\times d}$ with $\boldsymbol{R}_{j,\theta_k} = \begin{bmatrix}\cos(j\theta_k) & -\sin(j\theta_k)\\ \sin(j\theta_k) & \cos(j\theta_k)\end{bmatrix} \in \mathbb{R}^{2\times 2}$ and $\theta_k = \theta^{-2(k-1)/d}$ for all $1 \le k \le d/2$. For any $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{x}^{(k)} := [x_{2k-1}, x_{2k}]^\top \in \mathbb{R}^2$ denotes the subvector in the k^{th} rotation plane. $\|\cdot\|_2$ denotes the Euclidean or ℓ_2 norm of a vector or denotes the spectral norm of a matrix, and $\|\cdot\|_F$ denotes the Frobenius matrix norm. The stable rank of a matrix $\boldsymbol{X}$ is defined as $\mathrm{sr}(X) := \frac{\|X\|_F^2}{\|X\|_2^2}$. Finally, $\odot$ denotes the element-wise product.

Lemma 1. *For any $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, applying RoPE preserves its Frobenius norm, i.e., $\|\mathcal{R}(\boldsymbol{X})\|_F = \|\boldsymbol{X}\|_F$.*

Proof of Lemma 1. Since $\|\boldsymbol{X}\|_F^2 = \sum_{j=1}^n \|\boldsymbol{x}_j\|_2^2$ and $\|\mathcal{R}(\boldsymbol{X})\|_F^2 = \sum_{j=1}^n \|\boldsymbol{R}_j\boldsymbol{x}_j\|_2^2$, it suffices to show that for all j, $\|\boldsymbol{x}_j\|_2^2 = \|\boldsymbol{R}_j\boldsymbol{x}_j\|_2^2$. Since every diagonal block of $\boldsymbol{R}_j$ is a 2×2 rotation matrix, $\boldsymbol{R}_j$ is also a rotation matrix and thus norm preserving, which completes the proof. □

Theorem 1. *Suppose $\boldsymbol{X} = \boldsymbol{u}\boldsymbol{v}^\top \in \mathbb{R}^{n\times d}$, where $\boldsymbol{u} \in \mathbb{R}^n$, $\boldsymbol{v} \in \mathbb{R}^d$ and $\|\boldsymbol{v}\|_2 = 1$. If $\forall j$, $u_j = \Theta(1)$, and $\boldsymbol{u} \odot \boldsymbol{u}$ has sublinear growth of total variation in the sequence length n, i.e., $\sum_{j=1}^{n-1} |u_{j+1}^2 - u_j^2| = o(n)$, then as n increases, we have*

$$\frac{\|\mathcal{R}(\boldsymbol{X})\|_2}{\|\boldsymbol{X}\|_2} = \frac{1}{\sqrt{2}} \max_{1\le k\le d/2} \alpha_k + o(1)$$

where $\alpha_k := \|\boldsymbol{v}^{(k)}\|_2 = \sqrt{v_{2k-1}^2 + v_{2k}^2}$ satisfying $\max_k \alpha_k \in [\sqrt{2/d}, 1]$.

Remark 1. *The assumption that $\sum_{j=1}^{n-1} |u_{j+1}^2 - u_j^2| = o(n)$ implies that the sequence $\{u_j^2\}$ must exhibit a certain degree of monotonicity. Indeed, if $\{u_j^2\}$ is strictly monotonic, then $\sum_{j=1}^{n-1} |u_{j+1}^2 - u_j^2| = \Theta(1)$. In contrast, if $\{u_j^2\}$ is highly oscillatory, then $\sum_{j=1}^{n-1} |u_{j+1}^2 - u_j^2| = \Theta(n)$, which violates the assumption.*

Proof of Theorem 1. The pre-RoPE spectral norm

$$\|\boldsymbol{X}\|_2 = \sqrt{\lambda_{\max}(\boldsymbol{X}^\top\boldsymbol{X})} = \|\boldsymbol{u}\|_2\sqrt{\lambda_{\max}(\boldsymbol{v}\boldsymbol{v}^\top)} = \|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2 = \|\boldsymbol{u}\|_2 = \Theta(\sqrt{n}),$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue.

In what follows, we estimate the growth of post-RoPE spectral norm, i.e., $\sqrt{\lambda_{\max}(\mathcal{R}(\boldsymbol{X})^\top \mathcal{R}(\boldsymbol{X}))}$. Since $\|\boldsymbol{v}^{(k)}\|_2 = \alpha_k$, due to rational invariance, without loss of generality, we assume $\boldsymbol{v}^{(k)} = [\alpha_k, 0]^\top$ for simplicity.

Consider the Gram matrix $\boldsymbol{G} := \mathcal{R}(\boldsymbol{X})^\top \mathcal{R}(\boldsymbol{X}) \in \mathbb{R}^{d\times d}$, since $\mathcal{R}(\boldsymbol{X}) = [u_1\boldsymbol{R}_1\boldsymbol{v}, \cdots, u_n\boldsymbol{R}_n\boldsymbol{v}]^\top \in \mathbb{R}^{n\times d}$, we have

$$\boldsymbol{G} = \mathcal{R}(\boldsymbol{X})^\top \mathcal{R}(\boldsymbol{X}) = [u_1\boldsymbol{R}_1\boldsymbol{v},\ \cdots,\ u_n\boldsymbol{R}_n\boldsymbol{v}] \begin{bmatrix} u_1(\boldsymbol{R}_1\boldsymbol{v})^\top \\ \vdots \\ u_n(\boldsymbol{R}_n\boldsymbol{v})^\top \end{bmatrix} = \sum_{j=1}^n u_j^2 (\boldsymbol{R}_j\boldsymbol{v})(\boldsymbol{R}_j\boldsymbol{v})^\top$$

$$= \sum_{j=1}^n u_j^2 \begin{bmatrix} \boldsymbol{R}_{j,\theta_1}\boldsymbol{v}^{(1)} \\ \vdots \\ \boldsymbol{R}_{j,\theta_{d/2}}\boldsymbol{v}^{(d/2)} \end{bmatrix} \left[(\boldsymbol{R}_{j,\theta_1}\boldsymbol{v}^{(1)})^\top,\ \cdots,\ (\boldsymbol{R}_{j,\theta_{d/2}}\boldsymbol{v}^{(d/2)})^\top\right],$$

Diagonal Blocks of $\boldsymbol{G}$. For $1 \le k \le d/2$, the k^{th} diagonal block of $\boldsymbol{G}$ is given by

$$\boldsymbol{G}_{k,k} = \sum_{j=1}^n u_j^2 (\boldsymbol{R}_{j,\theta_k}\boldsymbol{v}^{(k)})(\boldsymbol{R}_{j,\theta_k}\boldsymbol{v}^{(k)})^\top \in \mathbb{R}^{2\times 2}.$$

Recall that

$$\boldsymbol{R}_{j,\theta_k} = \begin{bmatrix} \cos(j\theta_k) & -\sin(j\theta_k) \\ \sin(j\theta_k) & \cos(j\theta_k) \end{bmatrix},$$

we have $\boldsymbol{R}_{j,\theta_k}\boldsymbol{v}^{(k)} = \alpha_k[\cos(j\theta_k), \sin(j\theta_k)]^\top$, and thus

$$\begin{aligned} \boldsymbol{G}_{k,k} &= \alpha_k^2 \sum_{j=1}^n u_j^2 \begin{bmatrix} \cos^2(j\theta_k) & \cos(j\theta_k)\sin(j\theta_k) \\ \cos(j\theta_k)\sin(j\theta_k) & \sin^2(j\theta_k) \end{bmatrix} \\ &= \frac{\alpha_k^2}{2} \sum_{j=1}^n u_j^2 \begin{bmatrix} 1+\cos(2j\theta_k) & \sin(2j\theta_k) \\ \sin(2j\theta_k) & 1-\cos(2j\theta_k) \end{bmatrix} \\ &= \frac{\alpha_k^2}{2}\|\boldsymbol{u}\|_2^2\, \boldsymbol{I}_2 + \boldsymbol{E}_{k,k}, \end{aligned}$$

where

$$\boldsymbol{E}_{k,k} := \frac{\alpha_k^2}{2} \sum_{j=1}^n u_j^2 \begin{bmatrix} \cos(2j\theta_k) & \sin(2j\theta_k) \\ \sin(2j\theta_k) & -\cos(2j\theta_k) \end{bmatrix}.$$

Off-Diagonal Blocks of $\boldsymbol{G}$. Similarly, for $k \ne l$, the $(k,l)^{\text{th}}$ block is

$$\begin{aligned} \boldsymbol{E}_{k,l} := \boldsymbol{G}_{k,l} &= \sum_{j=1}^n u_j^2 (\boldsymbol{R}_{j,\theta_k}\boldsymbol{v}^{(k)})(\boldsymbol{R}_{j,\theta_l}\boldsymbol{v}^{(l)})^\top \\ &= \alpha_k^2 \sum_{j=1}^n u_j^2 \begin{bmatrix} \cos(j\theta_k)\cos(j\theta_l), & \cos(j\theta_k)\sin(j\theta_l) \\ \sin(j\theta_k)\cos(j\theta_l), & \sin(j\theta_k)\sin(j\theta_l) \end{bmatrix} \\ &= \frac{\alpha_k^2}{2} \sum_{j=1}^n u_j^2 \begin{bmatrix} \cos(j(\theta_k+\theta_l)) + \cos(j(\theta_k-\theta_l)), & \sin(j(\theta_k+\theta_l)) - \sin(j(\theta_k-\theta_l)) \\ \sin(j(\theta_k+\theta_l)) + \sin(j(\theta_k-\theta_l)), & -\cos(j(\theta_k+\theta_l)) + \cos(j(\theta_k-\theta_l)) \end{bmatrix} \end{aligned}$$

Therefore,

$$\boldsymbol{G} = \frac{\|\boldsymbol{u}\|_2^2}{2} \operatorname{diag}\left(\alpha_1^2\boldsymbol{I}_2, \cdots, \alpha_{d/2}^2\boldsymbol{I}_2\right) + \boldsymbol{E}.$$

Bounding $\lambda_{\max}(\boldsymbol{G})$. We want to show that $\boldsymbol{E}$ is a subleading term and is entry-wise $o(n)$. Note that $\alpha_k \le 1$, for any entry of $\boldsymbol{E}_{k,k}$, $\forall k$, it is upper bounded by

$$\sqrt{\left(\sum_{j=1}^n u_j^2 \cos(2j\theta_k)\right)^2 + \left(\sum_{j=1}^n u_j^2 \sin(2j\theta_k)\right)^2} = \left|\sum_{j=1}^n u_j^2 e^{2j\theta_k i}\right|.$$

Denoting $S_j := \sum_{t=1}^{j} e^{2t\theta_k i}$ and using Abel's summation formula, we have

$$\left|\sum_{j=1}^{n} u_j^2 e^{2j\theta_k i}\right| = \left|u_n^2 S_n - \sum_{j=1}^{n-1}(u_{j+1}^2 - u_j^2) S_j\right|$$
$$\leq u_n^2 \, |S_n| + \max_{1\leq j\leq n-1} |S_j| \sum_{j=1}^{n-1} |u_{j+1}^2 - u_j^2|,$$

where $|S_j| = |e^{2\theta_k i}| \left|\frac{1-e^{2j\theta_k i}}{1-e^{2\theta_k i}}\right| \leq \frac{2}{|1-e^{2\theta_k i}|} = \frac{1}{|\sin(\theta_k)|} = O(1)$ for all j, since $\theta_k = \theta^{-2(k-1)/d} \in [\frac{1}{\theta}, 1]$ is an irrational multiple of π. Moreover, we have assumed $\sum_{j=1}^{n-1} |u_{j+1}^2 - u_j^2| = o(n)$, so $\left|\sum_{j=1}^{n} u_j^2 e^{2j\theta_k i}\right| = o(n)$.

Similarly, for any entry of the off-diagonal block $\boldsymbol{E}_{k,l}$, we can also show that it is upper bounded by

$$\left|\sum_{j=1}^{n} u_j^2 e^{j(\theta_k+\theta_l)i}\right| + \left|\sum_{j=1}^{n} u_j^2 e^{j(\theta_k-\theta_l)i}\right| = o(n),$$

since $(\theta_k \pm \theta_l)/2$ are still irrational multiples of π.

By Weyl's inequality, we have

$$\frac{\|\boldsymbol{u}\|_2^2}{2} \max_k \alpha_k^2 - \|\boldsymbol{E}\|_2 \leq \lambda_{\max}(\boldsymbol{G}) \leq \frac{\|\boldsymbol{u}\|_2^2}{2} \max_k \alpha_k^2 + \|\boldsymbol{E}\|_2.$$

Note that since d is fixed, $\|\boldsymbol{E}\|_2 \leq \|\boldsymbol{E}\|_F = o(dn) = o(n)$, and thus

$$\lambda_{\max}(\boldsymbol{G}) = \frac{\|\boldsymbol{u}\|_2^2}{2} \max_k \alpha_k^2 + o(n).$$

Finally, since $\|\boldsymbol{u}\|_2 = \Theta(\sqrt{n})$, we have

$$\frac{\|\mathcal{R}(\boldsymbol{X})\|_2}{\|\boldsymbol{X}\|_2} = \frac{\sqrt{\lambda_{\max}(\boldsymbol{G})}}{\|\boldsymbol{u}\|_2} = \frac{1}{\sqrt{2}} \max_{1\leq k\leq d/2} \alpha_k + o(1).$$

□

Theorem 2. *Suppose $\boldsymbol{X} = \boldsymbol{u}\boldsymbol{v}^\top \in \mathbb{R}^{n\times d}$, where $\boldsymbol{u} \in \mathbb{R}^n$, $\boldsymbol{v} \in \mathbb{R}^d$ and $\|\boldsymbol{v}\|_2 = 1$. Under the same assumptions on $\boldsymbol{u}$ as in Theorem 1, we have*

$$\lim_{n\to\infty} \frac{\mathrm{sr}(\mathcal{R}(\boldsymbol{X}))}{\mathrm{sr}(\boldsymbol{X})} = \frac{2}{\max_{1\leq k\leq d/2} \alpha_k^2} \in [2, d],$$

where $\alpha_k := \|\boldsymbol{v}^{(k)}\|_2$.

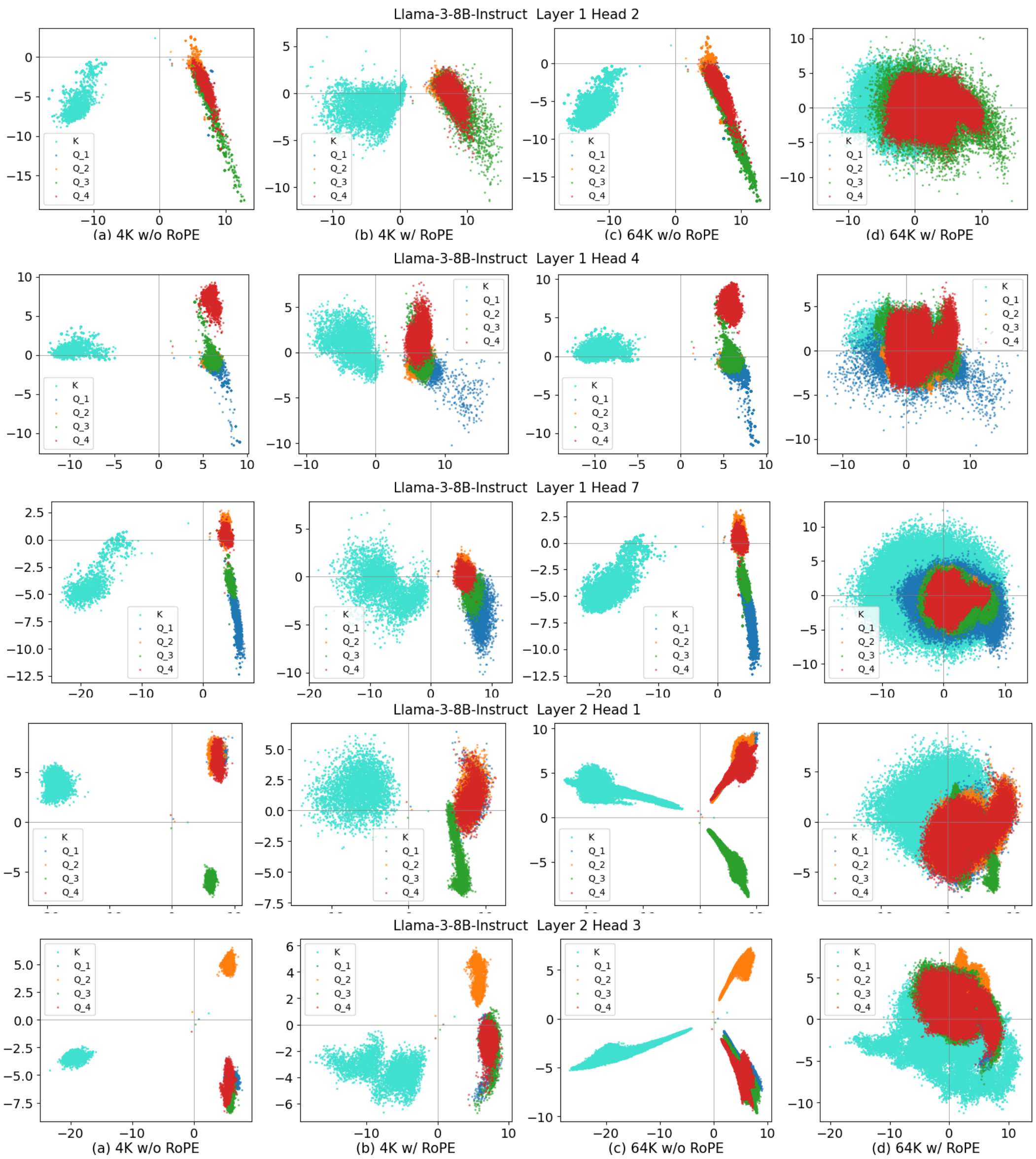
Proof of Theorem 2. Using Lemma 1 and Theorem 1, we have

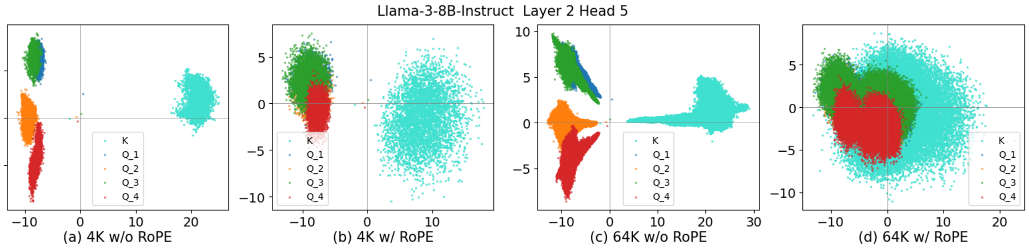
$$\frac{\mathrm{sr}(\mathcal{R}(\boldsymbol{X}))}{\mathrm{sr}(\boldsymbol{X})} = \left(\frac{\|\mathcal{R}(\boldsymbol{X})\|_F}{\|\boldsymbol{X}\|_F}\right)^2 \left(\frac{\|\boldsymbol{X}\|_2}{\|\mathcal{R}(\boldsymbol{X})\|_2}\right)^2 = \frac{2}{\max_{1\leq k\leq d/2} \alpha_k^2 + o(1)}.$$

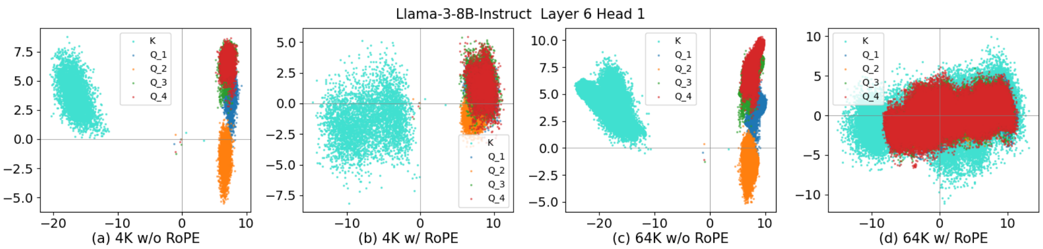
Since $\sum_{k=1}^{d/2} \alpha_k^2 = \|\boldsymbol{v}\|_2^2 = 1$, we have $\max_k \alpha_k^2 \in [2/d, 1]$. Taking $n \to \infty$ completes the proof. □

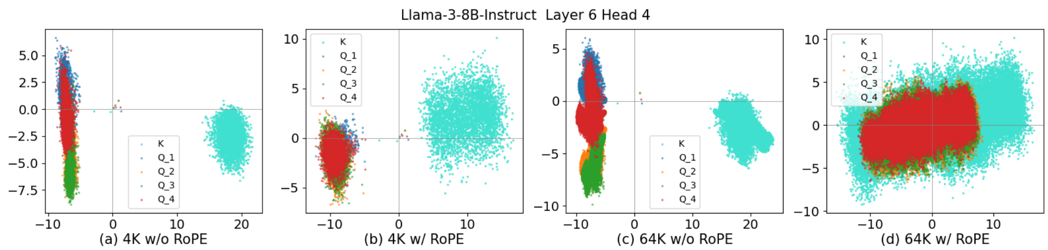
A.12 Additional Cluster Visualizations

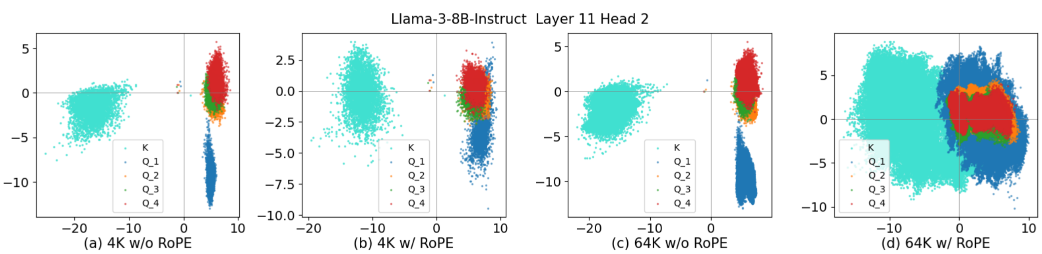
We repeat Fig. 3 for additional layers and heads. The same general trend can be observed, where separated clusters disperse and overlap when RoPE is applied at longer contexts. We randomly sample 16 Key heads and their corresponding Query heads from Llama3-8B-Instruct, and 8 Query-Key head pairs from Gemma-7b and Olmo-7b.

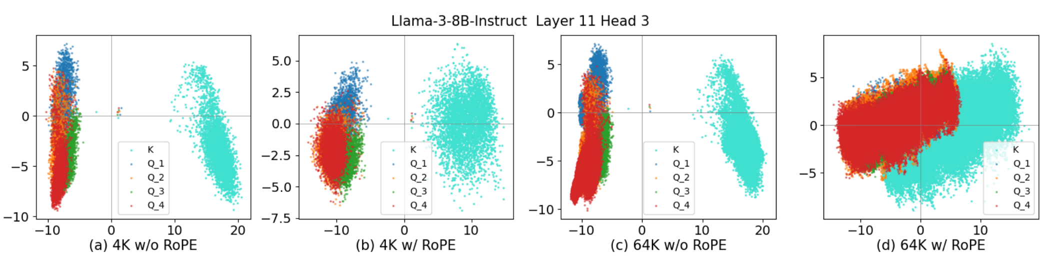


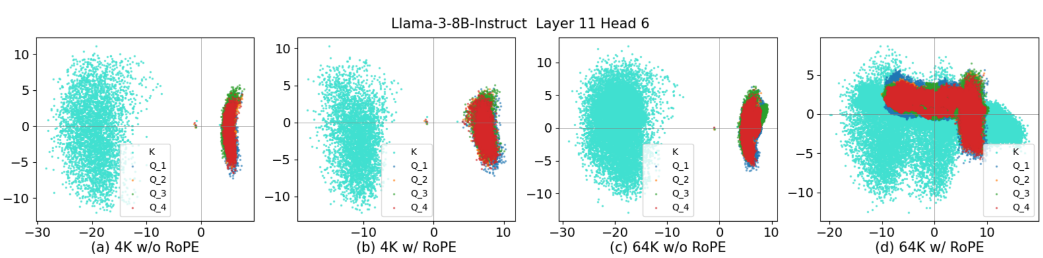

Llama-3-8B-Instruct Layer 2 Head 5
K
Q_1
Q_2
Q_3
Q_4
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE

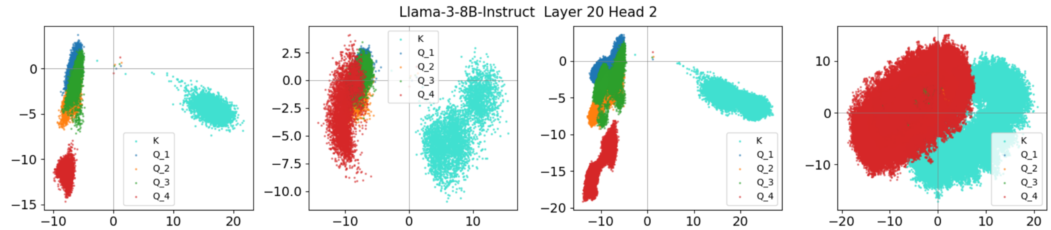

Llama-3-8B-Instruct Layer 6 Head 1
K
Q_1
Q_2
Q_3
Q_4
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE


Llama-3-8B-Instruct Layer 6 Head 4
K
Q_1
Q_2
Q_3
Q_4
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE

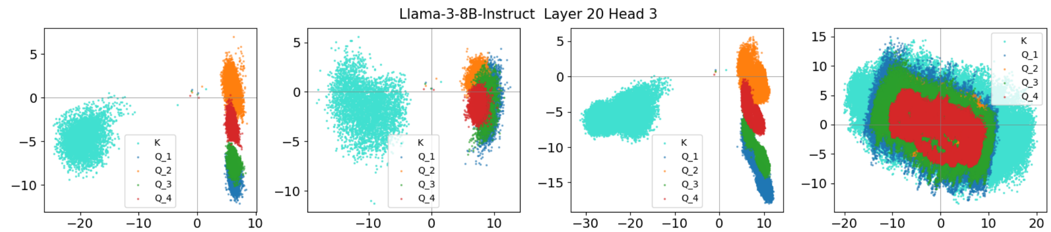

Llama-3-8B-Instruct Layer 11 Head 2
K
Q_1
Q_2
Q_3
Q_4
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE


Llama-3-8B-Instruct Layer 11 Head 3
K
Q_1
Q_2
Q_3
Q_4
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE

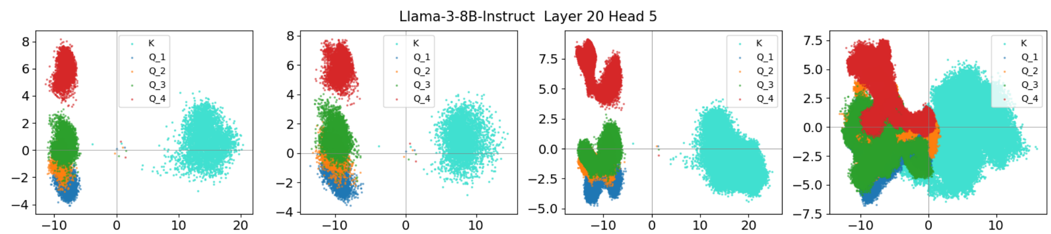

Llama-3-8B-Instruct Layer 11 Head 6
K
Q_1
Q_2
Q_3
Q_4
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE



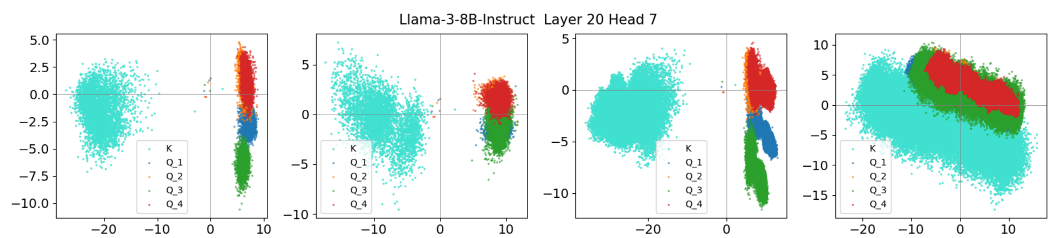
(a) 4K w/o RoPE (b) 4K w/ RoPE (c) 64K w/o RoPE (d) 64K w/ RoPE



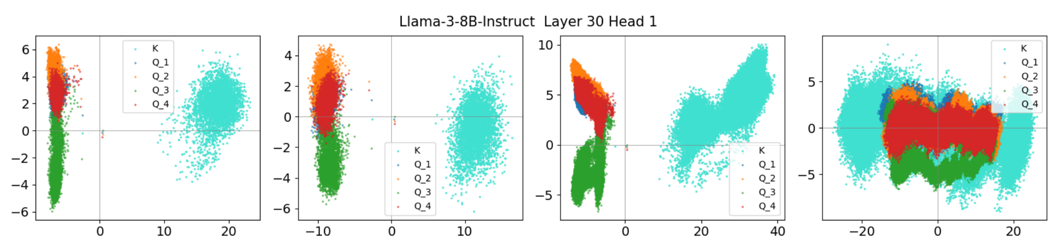
(a) 4K w/o RoPE (b) 4K w/ RoPE (c) 64K w/o RoPE (d) 64K w/ RoPE



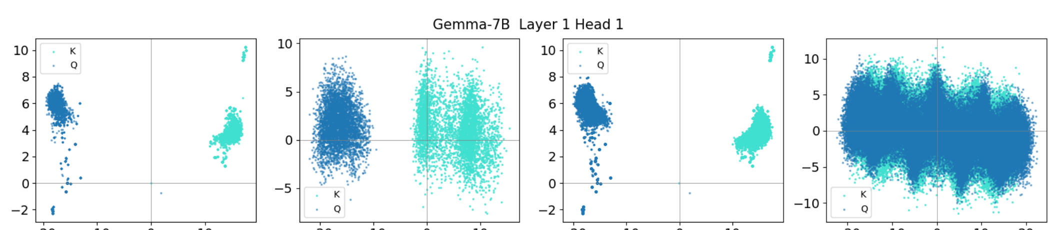
(a) 4K w/o RoPE (b) 4K w/ RoPE (c) 64K w/o RoPE (d) 64K w/ RoPE



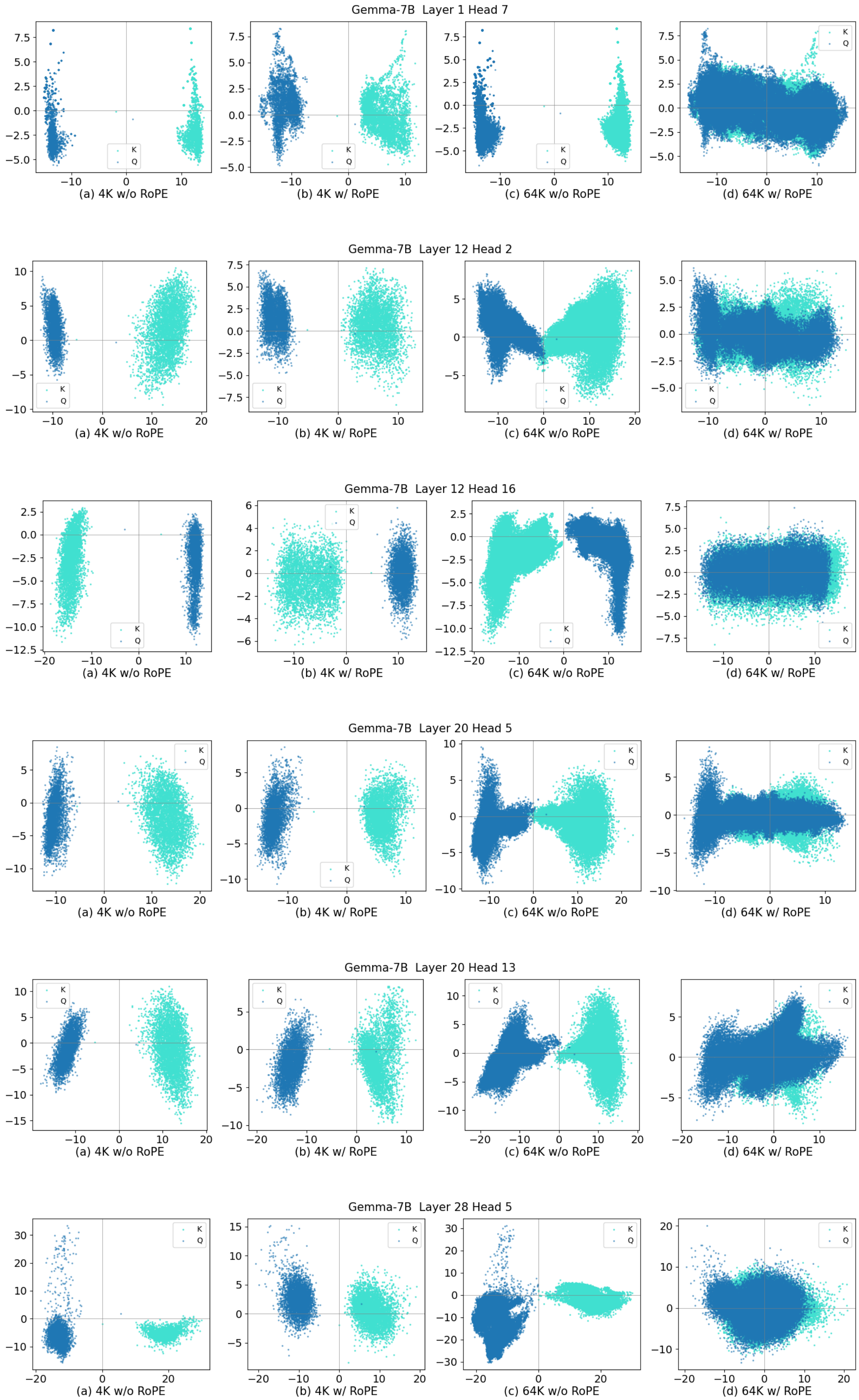
(a) 4K w/o RoPE (b) 4K w/ RoPE (c) 64K w/o RoPE (d) 64K w/ RoPE

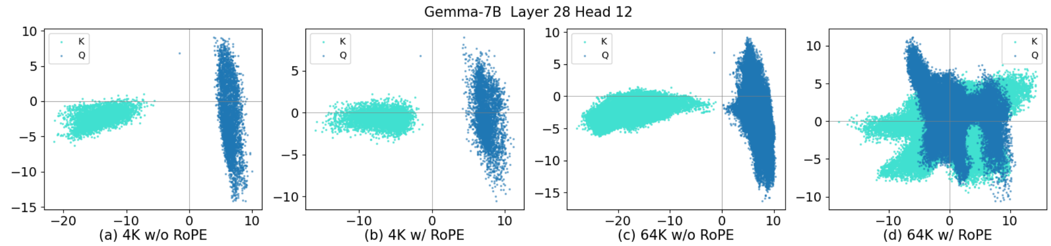


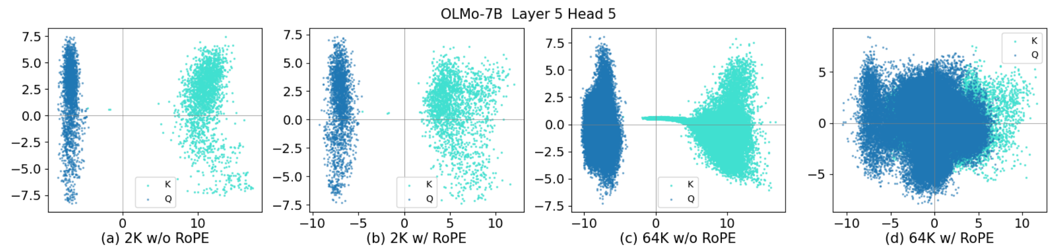
(a) 4K w/o RoPE (b) 4K w/ RoPE (c) 64K w/o RoPE (d) 64K w/ RoPE

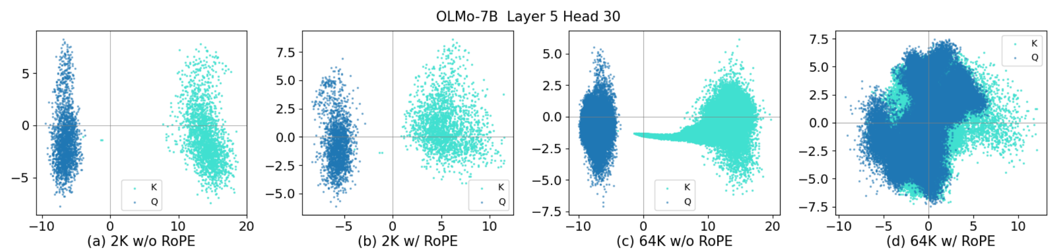


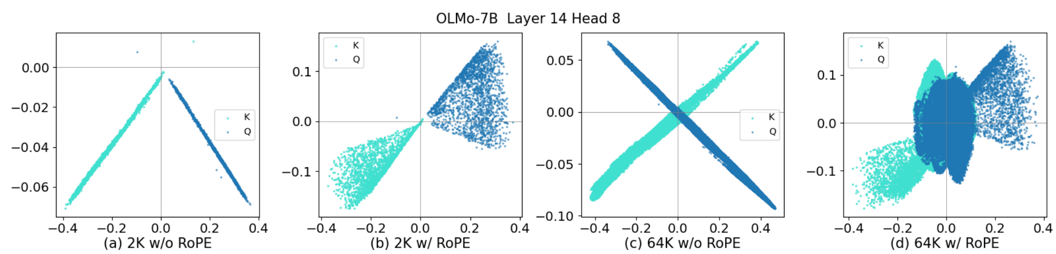
(a) 4K w/o RoPE (b) 4K w/ RoPE (c) 64K w/o RoPE (d) 64K w/ RoPE

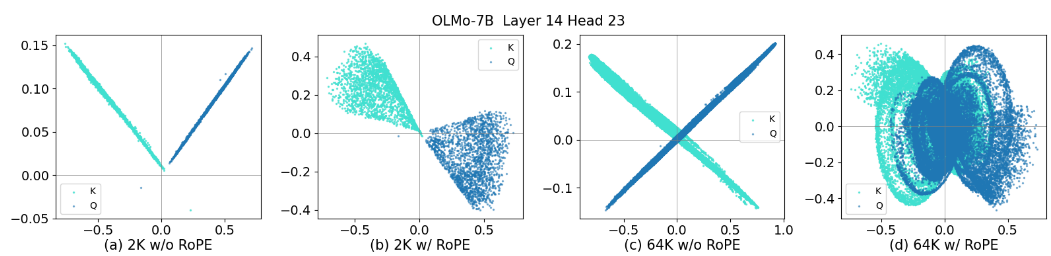

Gemma-7B Layer 1 Head 7
K
Q
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE
Gemma-7B Layer 12 Head 2
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE
Gemma-7B Layer 12 Head 16
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE
Gemma-7B Layer 20 Head 5
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE
Gemma-7B Layer 20 Head 13
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE
Gemma-7B Layer 28 Head 5
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE

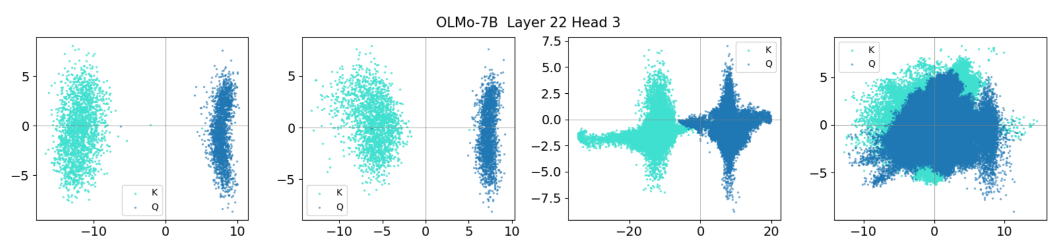

Gemma-7B Layer 28 Head 12
K
Q
(a) 4K w/o RoPE
(b) 4K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE

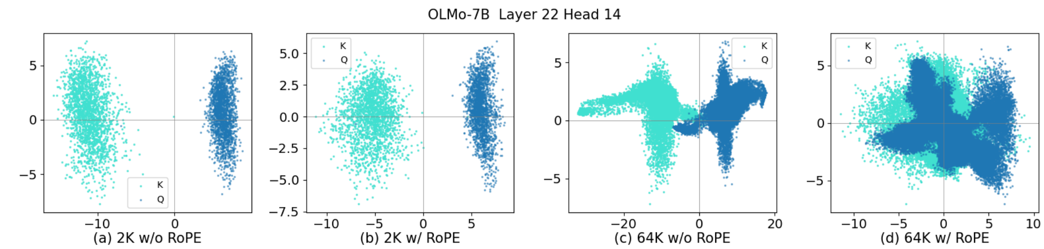

OLMo-7B Layer 5 Head 5
K
Q
(a) 2K w/o RoPE
(b) 2K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE

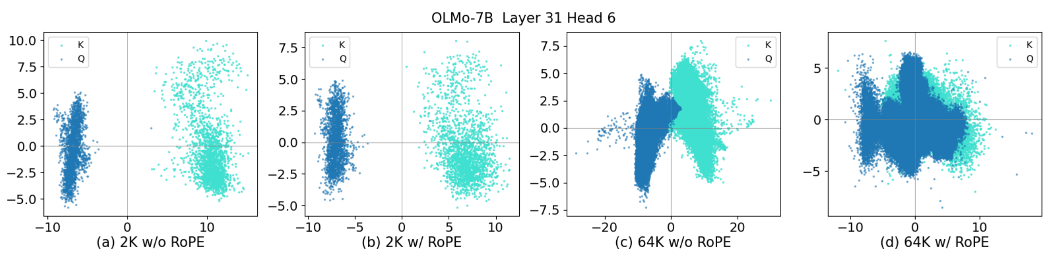

OLMo-7B Layer 5 Head 30
K
Q
(a) 2K w/o RoPE
(b) 2K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE

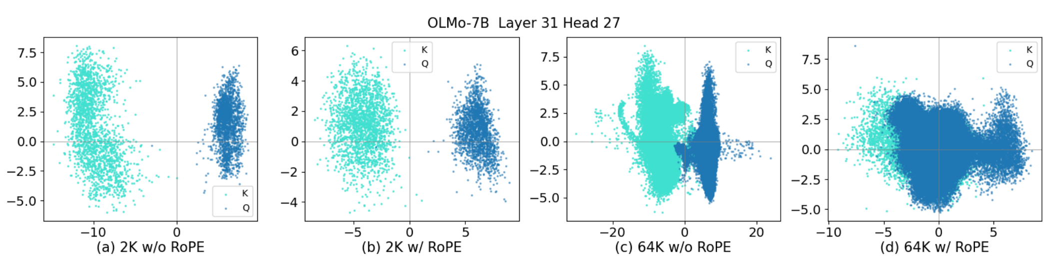

OLMo-7B Layer 14 Head 8
K
Q
(a) 2K w/o RoPE
(b) 2K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE


OLMo-7B Layer 14 Head 23
K
Q
(a) 2K w/o RoPE
(b) 2K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE


OLMo-7B Layer 22 Head 3
K
Q
(a) 2K w/o RoPE
(b) 2K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE


OLMo-7B Layer 22 Head 14
(a) 2K w/o RoPE
(b) 2K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE


OLMo-7B Layer 31 Head 6
(a) 2K w/o RoPE
(b) 2K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE


OLMo-7B Layer 31 Head 27
(a) 2K w/o RoPE
(b) 2K w/ RoPE
(c) 64K w/o RoPE
(d) 64K w/ RoPE