Anchors First, Reason Later: A Layer-Wise View of Long-Context Processing

Anonymous ACL submission

Abstract

Extending the context length of large language models (LLMs) remains challenging, especially when models are expected to preserve reasoning performance as sequence length increases. Many existing methods extend context by modifying rotary positional embeddings (RoPE). However, these approaches typically impose the same positional treatment across all layers and do not account for the hierarchical nature of representation formation inside the model. We present an Anchor-and-Reason view of long-context processing that emphasizes layer-wise functional differences. Specifically, we posit two regimes. In earlier layers, the model primarily performs an anchoring operation: accurate and sufficiently strong positional signals help organize long input sequences and support the formation of local semantic representations. In later layers, the model increasingly shifts to reasoning: it integrates intermediate representations to support global composition and deduction, where overly rigid positional constraints can become limiting. Based on this perspective, we propose Layer-Scaling for Position (LASP), a simple layer-dependent adjustment of positional strength. LASP maintains higher-frequency positional components in shallow layers to stabilize sequence-to-semantics mapping, while progressively reducing positional intensity in deeper layers via exponential decay, allowing higher layers to operate with fewer positional restrictions. Experiments on a range of long-context benchmarks show that LASP yields consistent improvements over strong baselines.

1 Introduction

Large Language Models (LLMs) have substantially advanced natural language processing, achieving strong performance in both language understanding and generation (Google DeepMind, 2025; Yang et al., 2025a; AI, 2024; OpenAI, 2023; Reid et al., 2024). With growing interest in long-document summarization, long-horizon agentic memory, and Retrieval-Augmented Generation (RAG) (Chen et al., 2023a; Li et al., 2024), the ability to handle extended context windows has become increasingly important.

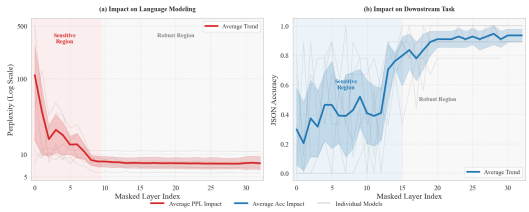

Figure 1: **Layer-wise sensitivity analysis of RoPE across multiple LLMs**. We evaluate the impact of Rotary Positional Embeddings (RoPE) by selectively masking them at individual layers. Using Perplexity (PPL) and Passkey-retrieval as long-context benchmarks, we observe that shallow layers are highly sensitive to RoPE removal, resulting in significant performance degradation. Conversely, deeper layers exhibit remarkable robustness, with both recall and PPL remaining nearly stable. This phenomenon is consistent across various base models, including Llama-1/2/3, Qwen, Phi, and DeepSeek. The curve illustrates the averaged performance across these architectures.

The quadratic cost of Transformer self-attention makes training directly on very long sequences expensive. Consequently, a common practice is to pre-train with moderate context lengths (e.g., 4k–8k tokens), then extend the context window by adjusting Rotary Positional Embeddings (RoPE) and fine-tuning on a relatively small amount of long-context data. Prior work has explored a range of RoPE-based extensions, including Position Interpolation (Chen et al., 2023b), NTK-aware scaling (bloc97, 2023), YaRN (Peng et al., 2023), and LongRoPE (Ding et al., 2024) / LongRoPE2 (Shang et al., 2025). These methods expand the usable window, often by combining high-frequency extrapolation with low-frequency interpolation. However, most approaches apply the same positional

scaling throughout the network, implicitly assuming that all layers rely on positional signals in a similar manner. This design choice does not reflect the layer-wise differences that emerge during representation formation.

Findings from interpretability studies indicate that Transformer layers tend to exhibit functional differentiation: earlier layers are more strongly associated with local feature extraction (e.g., lexical or short-range patterns), whereas deeper layers increasingly support long-range integration and compositional reasoning (Haviv et al., 2022; Liu et al., 2024).

To combine with long context extension, we perform a set of RoPE ablation experiments by selectively masking positional information at different layers. As shown in Figure 1, we observe a clear layer-wise asymmetry: masking RoPE in shallow layers causes a sharp increase in perplexity (PPL) and leads to a collapse in retrieval performance, whereas masking RoPE in deeper layers has little effect on PPL. This suggests that early layers rely heavily on positional cues to build local structure and integrate nearby evidence, while later layers are comparatively more content-oriented and can carry out inference largely from semantic representations; in fact, strong positional signals in deep layers may even be detrimental by injecting noise.

Based on these findings, we hypothesize an *anchor-and-reasoning* mechanism in LLMs. Specifically, we posit that shallow layers function as **anchor layers**: they use positional information to aggregate token-level features, compose local evidence, and project it into a representation that is less sensitive to absolute positions. In contrast, deeper layers serve as **reasoning layers**, operating on these aggregated representations to perform higher-level inference.

Motivated by the observation that many semantic relations in natural language are not tied to absolute token indices, we argue that **extending a short-context base model to longer contexts hinges on correctly composing token information and mapping sequences beyond the pretraining length into an appropriate semantic representation space**. To test this hypothesis, we fine-tune two variants under the same parameter budget: one updates only the shallow (anchor) layers, and the other updates only the deep (reasoning) layers. Across long-context evaluations, updating the anchor layers yields substantially larger gains than updating the reasoning layers.

To further probe the role of anchor layers, we conduct model grafting experiments that splice a long-context model with a short-context model. Replacing the anchor layers of the short-context model with those from the long-context model produces a hybrid that largely inherits long-context capability. Conversely, swapping in the anchor layers from the short-context base model into the long-context model results in pronounced degradation on long-context benchmarks. These results support the view that anchor layers are critical for mapping long sequences into a representation space amenable to downstream reasoning.

Building on the proposed mechanism, we introduce **Layer-wise Scaling for Position (LASP)**, which applies layer-dependent scaling to positional signals so that each layer receives positional information commensurate with its functional role. LASP strengthens (and refines) positional contributions in anchor layers to improve the projection of long sequences into the correct semantic space, while attenuating positional influence in deep reasoning layers to reduce interference and better allocate capacity to long-context processing.

Our contributions are summarized as follows:

1. We show that base models are highly sensitive to the degradation or removal of RoPE in shallow layers, while deeper layers are substantially more robust to missing positional information in long-context settings.

2. We propose an anchor-and-reasoning mechanism for long-context extension, and provide evidence via targeted fine-tuning as well as model-grafting experiments that splice long- and short-context models.

3. Guided by this mechanism, we introduce LASP, a layer-wise positional scaling strategy that adapts to heterogeneous layer roles. Experiments on long-context benchmarks demonstrate that LASP consistently improves long-context performance.

2 Related Work

2.1 RoPE-based Context Extension

Rotary Positional Embedding (RoPE) (Su et al., 2021) is widely adopted in modern LLMs. Early long-context extensions mainly relied on Position Interpolation (PI) (Chen et al., 2023b), which rescales position indices to fit the pretraining range. Later methods noticed that

different frequency components respond differently: NTK-aware strategies (bloc97, 2023) better preserve high-frequency components for extrapolation, while YaRN (Peng et al., 2023) introduces attention re-scaling to reduce distribution shifts on long sequences. More recently, LongRoPE/LongRoPE2 (Ding et al., 2024; Shang et al., 2025) employ evolutionary search to find non-uniform, per-dimension scaling factors, moving beyond hand-crafted rules. However, these approaches largely treat the model as layer-homogeneous, typically applying uniform or broadly monotonic scaling across layers, which may miss layer-dependent positional sensitivity.

2.2 Hierarchical Positional Information

Recent studies suggest positional information is not equally necessary across layers. RoPENoPE-style designs (Yang et al., 2025b) interleave RoPE and NoPE layers to mitigate potential RoPE biases in deeper layers, and SWAN-GPT (Puvvada et al., 2025) combines NoPE global attention with RoPE-based sliding-window attention (with logit scaling) for better long-context generalization. Closest to our setting, Wang et al. (Wang et al., 2025) propose layer-dependent scaling (e.g., via Bézier curves) to alleviate lost-in-the-middle, but their study focuses on short-context evaluation and does not examine layer-wise positional influence under long-context continued pre-training. Related ideas also appear in other modalities, such as LaPE for ViTs (Yu et al., 2023), HiRoPE for code (Zhang et al., 2024a), and head-wise scaling (e.g., MS-PoE) (Zhang et al., 2024b). In contrast, we emphasize a stronger layer-wise hierarchy of positional sensitivity, aligning with the Transformer's local-to-global information flow.

3 Preliminaries

RoPE (Su et al., 2021) injects positional information by applying a rotation to query/key vectors on each pair of hidden dimensions. Let the hidden size be D (assume D is even). For the d-th 2D subspace index $d \in [0, D/2)$, RoPE defines a base rotation frequency

$$\theta_d = \mathcal{B}^{-2d/D}, \quad \mathcal{B} = 10{,}000. \tag{1}$$

Given token position p, the rotation angle is $p\theta_d$. Denote the 2D rotation matrix as

$$\mathbf{R}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}. \tag{2}$$

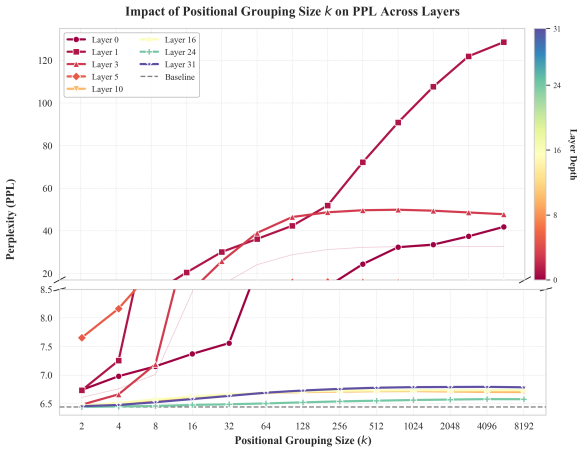

Figure 2: **Layer-wise sensitivity to positional grouping size k.** Perplexity (PPL) degradation is evaluated across different layers of Llama-3-8B as the positional grouping window k increases (log scale). The impact of positional granularity k on language modeling performance across layers. Shallow layers (red/orange) suffer catastrophic degradation with minimal grouping ($k \geq 8$), indicating a reliance on precise, token-level relative distances for low-level syntactic processing. Conversely, deep layers (blue/purple) remain robust to extreme coarsening ($k \to 8192$), suggesting that upper layers abstract away from exact coordinates to focus on the global ordering of semantic blocks. The gray dashed line denotes the original baseline PPL.

For layer l, RoPE transforms the query/key by applying $\mathbf{R}(p\theta_d)$ to each 2D block:

$$\begin{aligned} \tilde{\mathbf{q}}^{(l)}_{p,2d:2d+1} &= \mathbf{R}(p\theta_d)\mathbf{q}^{(l)}_{p,2d:2d+1} \\ \tilde{\mathbf{k}}^{(l)}_{p,2d:2d+1} &= \mathbf{R}(p\theta_d)\mathbf{k}^{(l)}_{p,2d:2d+1}. \end{aligned} \tag{3}$$

In standard Transformers, the same frequency schedule $\{\theta_d\}$ is shared across all layers.

Suppose a model is pre-trained with context length L and we aim to extend it to $L' = \kappa L$. A convenient unifying view is to modify RoPE by scaling the rotation frequencies:

$$\tilde{\theta}_d = \lambda_d \cdot \theta_d, \tag{4}$$

where λ_d is a (dimension-dependent) scaling factor. Many existing long-context extensions differ mainly in how they choose λ_d over the frequency spectrum: e.g., global compression $\lambda_d = \kappa^{-1}$ (Position Interpolation, PI) (Chen et al., 2023a), non-linear schedules that preserve high-frequency components more aggressively (NTK-aware scaling) (bloc97, 2023), or smooth transition functions (YaRN) (Peng et al., 2023). Conceptually, these methods all operate under the same "one scaling rule for the whole model" paradigm.

The above line of work typically enforces that the frequency scaling is *shared across layers*. Formally, let $\lambda_{l,d}$ denote the scaling applied at layer l and dimension index d. Existing approaches impose

$$\lambda_{1,d} = \lambda_{2,d} = \cdots = \lambda_{L_{\text{model}},d} = \lambda_d, \quad \forall d, \quad (5)$$

which implies a fixed global trade-off between interpolation and extrapolation for every layer.

4 The "Anchor-and-Reason" Mechanism

In this section, we first demonstrate that the impact of Rotary Positional Embeddings (RoPE) on long-context performance exhibits significant variance across different layers in a base model. We propose a two-stage mechanism termed "Anchor and Reasoning" to describe how models process long sequences.

4.1 Layer-wise Sensitivity to Positional Information

While prior studies(Tenney et al., 2019) have explored how different layers process distinct types of information, the specific contribution of layer-wise positional information to general long-context benchmarks remains under-explored. To bridge this gap, we systematically perturb positional information at specific layers and measure the resulting impact on model performance. We employ two widely recognized metrics: Perplexity (PPL), which quantifies the model's uncertainty via next-token prediction loss, and passkey-recall, a standard retrieval-based benchmark for evaluating long-context grounding. For passkey-recall, we insert the "needle" (target key) uniformly across 10 distinct depth intervals and report the mean accuracy.To quantify a layer's dependence on positional information, we introduce a masking mechanism. Let $\mathbf{x}_m^{(l)}$ denote the input vector at position m for layer l. To probe layer l, we apply a binary mask $\gamma_l \in \{0, 1\}$. The relative position encoding within the attention mechanism for this layer is formulated as:

$$\mathbf{h}_m^{(l)} = \text{Attention}\left(\tilde{\mathbf{q}}_m^{(l)}, \tilde{\mathbf{k}}_n^{(l)}\right)$$
$$\text{where} \quad \begin{cases} \tilde{\mathbf{q}}_m^{(l)} = \mathcal{R}\left(\mathbf{q}_m^{(l)}, m \cdot \gamma_l\right) \\ \tilde{\mathbf{k}}_n^{(l)} = \mathcal{R}\left(\mathbf{k}_n^{(l)}, n \cdot \gamma_l\right) \end{cases} \quad (6)$$

where $\mathcal{R}(\cdot)$ denotes the rotation operator. When $\gamma_l = 1$, the layer functions normally; when $\gamma_l = 0$, the position indices are zeroed out, effectively eliminating the relative distance information $(m - n)$ for that specific layer.Our results, as illustrated in Figure 1, reveal a striking trend. Masking RoPE in shallow layers triggers a catastrophic surge in PPL (e.g., from < 10 to over 100) and a near-total loss of passkey-recall. This underscores that shallow layers are heavily reliant on positional signals to establish basic semantic structures. However, as the mask moves to deeper layers, PPL rapidly converges to the vanilla baseline, and Recall recovers fully. This suggests a high degree of robustness to positional perturbations in deeper layers. To ensure these observations are not artifacts of cumulative error, we conducted a control experiment (detailed in Appendix A.2) which confirms that the observed sensitivity is an intrinsic preference of the model architecture rather than error propagation.Our experiments span a diverse range of model architectures (Phi, Llama, Qwen, Mistral, DeepSeek) and sizes (1B–24B). Despite differences in the number of layers and parameter configurations, these models exhibit similar characteristics.

To further investigate how layers utilize positional information, we introduce a Positional Coarsening function $\phi_k(\cdot)$, parameterized by a grouping factor $k \in Z^+$. The transformed position index $\tilde{m}$ is defined as:

$$\tilde{m} = \phi_k(m) = \left\lfloor \frac{m}{k} \right\rfloor$$

This transformation reduces the positional resolution to $1/k$, turning the relative distance into a step function. We group k adjacent tokens together and assign them identical positional information within RoPE. Consequently, within a single group, RoPE no longer provides local positional distinctions, although relative positional relationships between different groups remain intact. We posit that this approach primarily affects the local positional information encoded by RoPE. As shown in Figure 2, changing k from 2 to 4096, we observe the tolerance of different layers to the degradation of the resolution.In shallow layers, even minor coarsening (e.g., $k = 2$) leads to a significant degradation in PPL, indicating a requirement for precise, token-level relative distances to parse syntax. Conversely, in deep layers, PPL remains remarkably stable even at $k = 4096$. This demonstrates that deep layers shift their focus from precise coordinates to the relative order of macro-level Semantic Blocks.

Based on the above experiments, we conclude that the local positional information provided by

Model	32k						64k					
	Recall	RAG	Long-QA	Cite	ICL	Rerank	Recall	RAG	Long-QA	Cite	ICL	Rerank
Baselines												
Llama-3.1-8B	90.19	63.00	36.20	**3.52**	71.76	**35.87**	86.31	60.67	**45.05**	1.95	**75.48**	22.05
llama-8B(with long embedding)	12.12	46.96	19.20	1.90	62.04	30.89	2.25	32.71	22.35	1.15	63.24	14.59
fintune												
Anchor-8Layer(shallow)	83.60	60.90	36.60	3.29	77.68	31.00	80.18	57.75	43.35	1.91	82.72	19.41
Resean-8Layers(Deep)	74.25	59.58	34.20	3.17	64.64	26.89	67.56	55.91	40.55	1.8	66.1	17.33
Model Grafting												
Anchor-8Layers	35.81	52.04	32.33	2.00	67.44	30.17	26.75	50.92	35.44	1.61	72.52	**22.46**
Anchor-16Layers	**98.25**	63.04	33.67	1.74	72.16	29.17	**92.12**	59.84	42.66	2.26	75.12	20.69
Anchor-24Layers	93.62	**63.29**	**37.05**	2.75	**72.16**	32.09	87.50	**60.67**	43.28	**2.33**	75.04	19.43

Table 1: HELMET benchmark Performance aggregation at 32k and 64k context.Fine-tuning the first 8 layers improves long-context performance far more than tuning the last 8. In model grafting, the hybrid reaches—and at K=16 can exceed the long-context model, indicating that long-context ability is largely set by shallow anchor layers and that the base and long-context models share a compatible semantic space once early representations are aligned.

RoPE in the anchor (shallow) layers plays a critical role in semantic composition and long-sequence modeling, thereby substantially affecting long-context capability. When such local positional signals are degraded or removed in the shallow layers, the model's overall performance deteriorates markedly. In contrast, similar perturbations applied to deeper layers have a much smaller impact on final model quality.

4.2 The Anchor-and-Reason Hypothesis

Based on the above empirical observations, we formulate the Anchor-and-Reason conjecture: when acquiring the ability to model long sequences, a model may undergo a two-stage process. In the early (shallow) layers, the model leverages the high-frequency and precise positional signals provided by RoPE to anchor tokens to their locations in the sequence, thereby constructing local semantic representations. Once such local semantics are established, the model becomes less reliant on RoPE's positional cues; it can instead use semantic information to support long-range modeling.

More specifically, we partition the network layers into two stages:

Anchor Layers. In this stage, the model receives token inputs and combines them with RoPE-based positional information and nearby tokens to form local semantic representations, effectively mapping long-sequence information into a local semantic space.

Reasoning Layers. In this stage, the model primarily operates on the local semantic representations formed in the shallow layers, performing more complex semantic composition and retrieval. After local semantics have been established, the model no longer depends strongly on RoPE-provided positional information; accordingly, partial loss or degradation of positional signals does not substantially affect the resulting semantic representations.

This perspective is further motivated by the observation that the semantics of long texts can be largely regarded as being covered by collections of short-text semantic subspaces. When extending context length, semantic information is therefore less likely than positional information to suffer from out-of-distribution (OOD) issues. Consequently, the central challenge of long-context extension is to ensure that, under a shifted positional distribution, the model can still correctly aggregate and compose neighboring tokens, mapping the semantics of long sequences into the pretrained local semantic space. Under this view, effective long-context extension hinges on providing appropriate positional information in the lower (anchor) layers, so that long-sequence inputs are mapped correctly into their corresponding local semantic regions.

4.3 Experimental Validation

To validate this hypothesis, we designed two sets of experiments:

Partial Fine-tuning.To demonstrate that the location-sensitive anchor layer is the key factor determining performance on long-context tasks, We freeze most parameters and fine-tune only the first k layers. As a baseline, we compare this against fine-tuning only the last k layers. We set $k = 8$ for Llama3-8B, corresponding to the layers identified as most position-sensitive in Figure 1.

Model Grafting.To show that Anchor-Layer maps long-sequence information into the same lo-

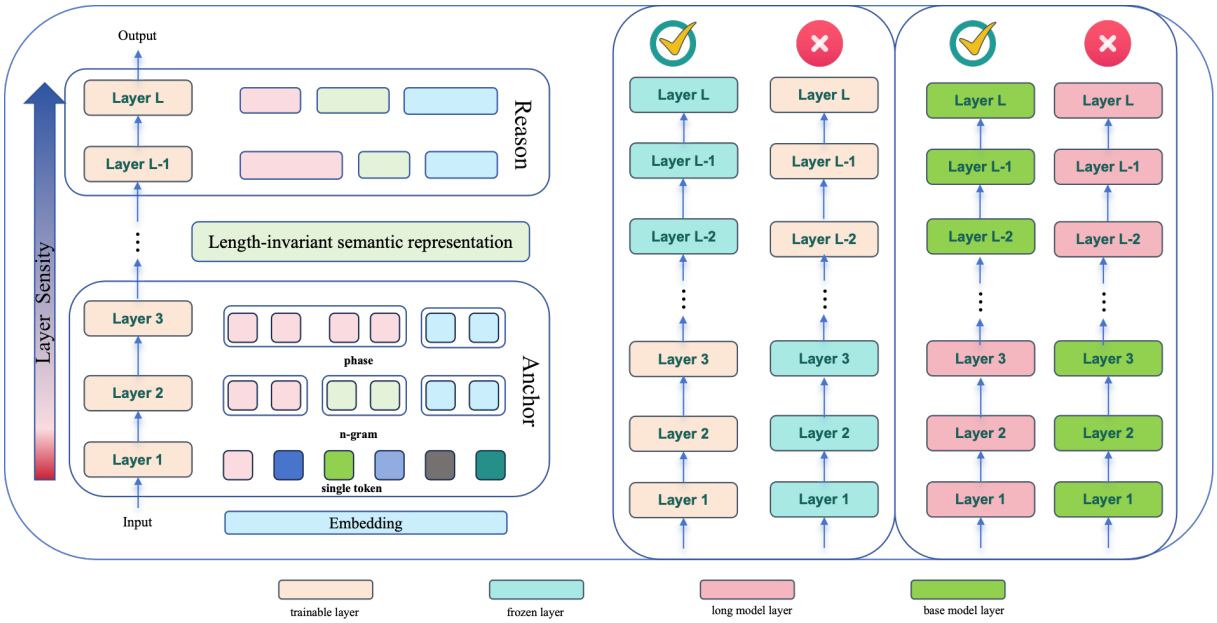

Figure 3: **Mechanism of "Anchor and Reasoning" in LLM Context Extension**. During the extension process, shallow anchor layers aggregate token-level information using new positional cues, synthesizing local context into a position-agnostic semantic space. In contrast, deeper reasoning layers leverage these integrated representations for complex cognitive tasks. Empirical evidence supports this division: 1) Fine-tuning shallow anchor layers yields significantly higher long-context performance compared to fine-tuning deep reasoning layers of the same parameter scale. 2) To validate the existence of this semantic space, we "graft" models by replacing the anchor layers of a short-context LLM with those from a long-context variant. This hybrid model successfully achieves robust long-context capabilities, whereas the reverse configuration fails.

cal semantic space as short sequences, we conduct model-grafting experiments. We construct a hybrid model $\Theta^{(K)}_{\text{Hybrid}}$ by stitching the first K layers of a long-context model Θ_{Long} with the remaining $L-K$ layers of a short-context model Θ_{Short}. The embedding and head layers are similarly partitioned. Simultaneously, we conducted a comparative analysis of long-context capabilities by replacing an equivalent number of deep layers across different hybrid models with identical parameter counts.

$$\Theta^{(K)}_{\text{Hybrid}} = \left\{ \theta^{(l)} \mid l = 0, 1, \ldots, L, \right\}$$
$$\text{where} \quad \theta^{(l)} = \begin{cases} \theta^{(l)}_{\text{Long}}, & 0 \le l < K \\ \theta^{(l)}_{\text{Short}}, & K \le l \le L \end{cases} \tag{7}$$

To ensure a robust evaluation, we employed HELMET (Yen et al., 2025), a comprehensive benchmark for long-context performance. As shown in Table 1, fine-tuning the first eight layers yields substantially stronger long-context performance than fine-tuning the last eight layers, indicating that the Anchor Layer's handling of positional information is central to long-context extension.

In the model-grafting experiments, the long-context Anchor Layer from the first 16 layers matches—and in several metrics surpasses—the Llama-3.1-8B baseline, suggesting that the semantic representations produced by the long-context Anchor Layer can be effectively consumed by the base model. Taken together, these results support the view that long-context capability primarily depends on whether the Anchor Layer can robustly process positional information. For a further comparison, we perform the same substitution on the long-context model itself; the results are shown in Figure 4.

5 LASP: Layer-wise Adaptive Scaling for Position

We have established that, in long-context extrapolation, shallow and deep layers play distinct roles, with the most critical component being the anchor layers that handle positional information. Existing methods differ in how they regulate the effective strength of RoPE positional signals, typically through method-specific control parameters. In this section, we propose the LASP framework, which adjusts the RoPE positional strength on a per-layer

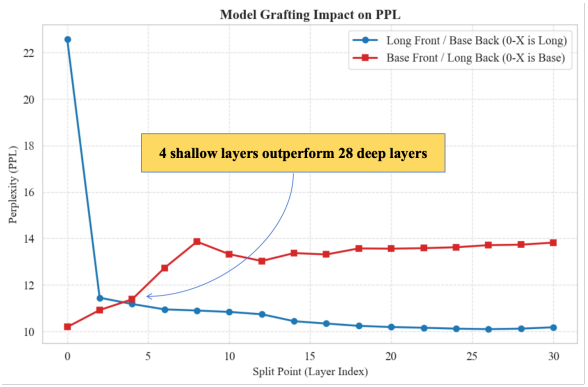

Figure 4: **Perplexity (PPL) analysis of layer-wise grafting between Llama 3.1-8B (128k) and Llama 3-8B (8k) on the PG-19 dataset**. Replacing the initial 8 layers of the long-context model with those from the short-context model triggers a significant PPL spike followed by stabilization. Conversely, substituting the initial layers of the short-context model with long-context parameters yields a steady PPL reduction. Notably, replacing only the first 4 layers of the short-context model with long-context weights outperforms a configuration where the final 28 layers are replaced.

basis so that the positional information better aligns with the functional role of each layer.

Model	Avg Score	Recall	RAG	Long-QA	Cite	ICL	Rerank
Yarn500k	50.84	97.54	62.06	**35.18**	4.55	69.35	36.35
Yarn2M	42.59	76.60	53.87	26.83	2.86	64.60	30.79
LASP(base)	50.04	82.50	**62.58**	33.28	5.05	**75.47**	**41.35**
LASP(scale)	51.73	98.95	61.73	33.18	3.94	73.47	39.11
LASP (Both)	**51.99**	**99.22**	61.05	33.67	**6.10**	72.43	39.44

Table 2: Evaluation results on the HELMET benchmark across various long-context tasks. We report the average score and performance across six key categories: Recall, RAG, Long document question-answering (Long-QA), Citation (Cite), In-Context Learning (ICL), and Reranking.

5.1 LASP: Layer-Aware Scaling and RoPE Base Scheduling

Conventional context extension applies a single global scaling factor to all layers, uniformly reducing RoPE rotation frequencies. LASP generalizes this idea by introducing a *layer-dependent* schedule, so that shallow "anchoring" layers preserve fine-grained positional resolution while deeper "reasoning" layers prioritize long-range capacity.

Let the RoPE frequency for dimension d (hidden size D) be

$$\theta_d^{(l)} = \mathcal{B}(l)^{-2d/D}, \tag{8}$$

where we allow the RoPE base $\mathcal{B}(l)$ to vary by layer. LASP further modulates the effective frequency in layer l using a layer-wise scaling factor $\mathcal{S}(l)$:

$$\tilde{\theta}_d^{(l)} = \frac{\theta_d^{(l)}}{\mathcal{S}(l)} = \frac{\mathcal{B}(l)^{-2d/D}}{\mathcal{S}(l)}. \tag{9}$$

This decouples local structural anchoring (early layers) from global semantic reasoning over extended context (late layers).

In its simplest form, the granularity of positional information can be modulated by adjusting the RoPE scale and rope base. We propose a straightforward layer-wise strategy that designates the first K Transformer layers as anchor layers, which are most sensitive to positional cues; these layers are therefore supplied with the most fine-grained positional signals. Concretely, we apply a progressively increasing schedule for the scale within the first K layers. For the rope base, we observe that decreasing its value substantially degrades positional information, leading to a sharp increase in perplexity (PPL). In line with common practice in recent large-scale models, increasing the rope base is generally a more favorable direction. Accordingly, we keep the rope base fixed for the first K layers and then increase it gradually in subsequent layers, enabling a smooth transition while preserving high-resolution positional information in the early layers.

Piecewise linear scaling schedule. Sensitivity probing (Figure 1) suggests an anchoring depth L_{anchor} where the model transitions from local structure extraction to global reasoning. We therefore use a piecewise linear schedule that ramps from a conservative scale $s_{\min}$ to an aggressive scale $s_{\max}$ by L_{anchor}:

$$\begin{aligned} \lambda_s &= \frac{s_{\max} - s_{\min}}{L_{\text{anchor}}}, \\ \mathcal{S}(l) &= \begin{cases} s_{\min} + l \cdot \lambda_s, & l < L_{\text{anchor}}, \\ s_{\max}, & l \geq L_{\text{anchor}}. \end{cases} \end{aligned} \tag{10}$$

Layer-wise RoPE base schedule. Rather than decreasing the RoPE base to obtain finer positional granularity, we follow the principle used in recent architectures (e.g., Qwen-style designs) and *increase* the base in deeper layers, which are empirically less sensitive to precise positions. Concretely, we keep a standard base $b_{\min}$ in shallow layers and linearly increase it toward $b_{\max}$ after L_{anchor} across L_{tot} layers:

$$\lambda_b = \frac{b_{\max} - b_{\min}}{L_{\text{tot}} - L_{\text{anchor}}},$$
$$\mathcal{B}(l) = \begin{cases} b_{\min}, & l < L_{\text{anchor}}, \\ b_{\min} + \lambda_b (l - L_{\text{anchor}}), & l \geq L_{\text{anchor}}. \end{cases} \tag{11}$$

5.2 Experiments and Results

We adopt YaRN as our primary scaling strategy. Importantly, our framework is compatible with any dimension-based context-extension method, including NTK-aware scaling, YaRN, LongRoPE, and LongRoPE2. We use LLaMA-3-8B as the base model, as it is a *vanilla* pretrained model without any prior long-context modification. Our training corpus contains 5B tokens, filtered to retain sequences longer than 128k tokens.

We evaluate performance on two complementary benchmarks: LOFT (Lee et al., 2024), which targets real-world long-context scenarios, and HELMET (Yen et al., 2025), a comprehensive synthetic benchmark designed to provide holistic coverage of long-context capabilities.

We set $s_{\max} = 16$, $b_{\min} = 500{,}000$, and $b_{\max} = 2{,}000{,}000$. To enable controlled comparisons against both scaling and base-configuration factors, we consider two baselines YaRN-500k and YaRN-2M and three experimental variants: LASP(base), LASP(scale), and LASP(both), corresponding respectively to applying the proposed *base* adjustment, the proposed *scale* adjustment, and their combination.

Model	Avg Score	ArguAna	FEVER	HotpotQA	NQ	Quora	SciFact
Yarn500k	62.0	10.0	78.0	44.50	85.0	90.0	**66.0**
Yarn2M	64.0	10.0	89.0	43.50	93.0	89.0	57.0
LASP(base)	66.0	17.0	89.0	46.0	96.0	90.0	60.0
LASP(scale)	64.0	**18.0**	82.0	44.0	91.0	87.0	64.0
LASP(Both)	**69.0**	14.0	**92.0**	**47.5**	**97.0**	**96.0**	**66.0**

Table 3: Evaluation results on the LOFT real-world retrieval benchmark. We report the performance across six diverse retrieval datasets. The **Avg** column represents the mean score across all tasks. Our LASP variant consistently outperforms the baselines, demonstrating superior long-context retrieval capabilities.

Table 2 reports results on the HELMET synthetic benchmarks. We first observe that the 2M baseline underperforms the 500k baseline (YaRN-500k) across all metrics. Notably, LASP(base) achieves strong overall performance despite adopting a larger RoPE base in later layers, because the anchor layers retain $b = 500{,}000$. This design preserves effective positional grounding and yields competitive results, with the main degradation occurring on recall tasks, which depend more on precise positional information than semantic cues. In contrast, LASP(scale), which explicitly provides more accurate positional information, attains substantially better performance on recall. Finally, LASP(both) combines the strengths of both approaches, achieving the best average performance. It mitigates the recall weakness observed in LASP(base) while maintaining advantages on the remaining metrics.

As shown in Table 3, YaRN-2M does not exhibit the same degradation as on HELMET, which may indicate that real-world tasks rely more heavily on semantic information than exact positional precision. Meanwhile, LASP(scale) and LASP(base) each demonstrate advantages on different subsets of tasks, and LASP(both) further improves overall performance on real-world evaluations. These results suggest that incorporating stronger positional signals at anchor layers while reducing the degree of positional injection in deeper layers can improve long-context capacity and, in turn, enhance downstream performance.

6 Conclusion

We identify a hierarchical layer-wise dependence of large language models (LLMs) on Rotary Positional Embeddings (RoPE). Building on this observation, we propose the *anchor-and-reason* hypothesis for long-context extension: early "anchor" layers map information from long sequences into the semantic space learned during short-context pretraining, making positional signals in shallow layers critical for effective length generalization. We validate this hypothesis through finetuning studies and model-grafting experiments. Guided by this theory, we further introduce **LASP**, a dynamic strategy that assigns layer-specific positional configurations. Extensive experiments and long-context benchmarks demonstrate the effectiveness of LASP.

Limitations

Because of computational constraints, we trained Llama-3-8B on a reduced dataset, which prevented us from reaching state-of-the-art performance. Nevertheless, we observed the same behavior across nearly all open-source base models. In addition, although our ablation studies were conducted solely with YaRN, the proposed method is compatible

with other scaling-based approaches.Moreover, we did not tune layer-wise optimal configurations; instead, we adopted a simple progressive schedule shared across layers.

References

Meta AI. 2024. The llama 3 herd of models. *arXiv preprint*.

bloc97. 2023. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. Reddit.

Shouuan Chen and 1 others. 2023a. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023b. Extending context window of large language models via positional interpolation. *Preprint*, arXiv:2306.15595.

Yiran Ding and 1 others. 2024. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*.

Google DeepMind. 2025. Gemini 2.5: Unlocking thinking and long-context capabilities. Technical report, Google. Technical Blog and Model Card.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer language models without positional encodings still learn positional information. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1382–1390, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu, Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole, Sebastian Riedel, Iftekhar Naim, Ming-Wei Chang, and Kelvin Guu. 2024. Can long-context language models subsume retrieval, rag, sql, and more? *Preprint*, arXiv:2406.13121.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

OpenAI. 2023. New models and developer products announced at devday.

Bowen Peng and 1 others. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Krishna C. Puvvada, Faisal Ladhak, Santiago Akle Serrano, Cheng-Ping Hsieh, Shantanu Acharya, Somshubra Majumdar, Fei Jia, Samuel Kriman, Simeng Sun, Dima Rekesh, and Boris Ginsburg. 2025. SWAN-GPT: an efficient and scalable approach for long-context language modeling. *CoRR*, abs/2504.08719.

Machel Reid and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. 2025. Longrope2: Near-lossless llm context window scaling. *Preprint*, arXiv:2502.20082.

Jianlin Su and 1 others. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Zhenghua Wang, Yiran Ding, Changze Lv, Zhibo Xu, Tianlong Li, Tianyuan Shi, Xiaoqing Zheng, and Xuanjing Huang. 2025. Layer-specific scaling of positional encodings for superior long-context modeling. *Preprint*, arXiv:2503.04355.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Bowen Yang, Bharat Venkitesh, Dwarak Talupuru, Hangyu Lin, David Cairuz, Phil Blunsom, and Acyr Locatelli. 2025b. Rope to nope and back again: A new hybrid attention strategy. *Preprint*, arXiv:2501.18795.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. Helmet: How to evaluate long-context language models effectively and thoroughly. *Preprint*, arXiv:2410.02694.

Runyi Yu, Zhennan Wang, Yinhuai Wang, Kehan Li, Chang Liu, Haoyi Duan, Xiangyang Ji, and Jie Chen. 2023. Lape: Layer-adaptive position embedding for vision transformers with independent layer normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5886–5896.

Kechi Zhang, Ge Li, Huangzhao Zhang, and Zhi Jin. 2024a. Hirope: Length extrapolation for code models using hierarchical position. *arXiv preprint arXiv:2403.19115*.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, Zhangyang Wang, and 1 others. 2024b. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *Advances in Neural Information Processing Systems*, 37:60755–60775.

A Layer Sensity

A.1 Mask Rope

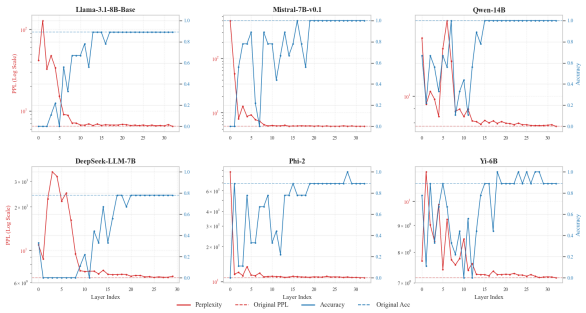

Figure 5: **Performance visualization of masking RoPE in specific layers across different models**. While evident PPL deviations occur in shallow layers, PPL and recall in deeper layers align closely with the unmasked configuration. For models with fewer parameters, specifically Phi-2, targeted RoPE masking results in improved effectiveness."

We investigated the impact of masking Rotary Positional Embeddings (RoPE) in specific layers on model performance, specifically focusing on changes in Perplexity (PPL) and Recall. Our findings reveal that while base models exhibit a high sensitivity to layer-wise RoPE masking, this dependency is significantly mitigated following long-context extension. Specifically, positional information processing becomes concentrated in fewer layers, whereas a larger number of layers become increasingly agnostic to positional signals. These evaluations were conducted at a sequence length of 8192 using the emozilla/pg19 dataset.

We observed a consistent phenomenon across nearly all model series: masking the RoPE positional information in the initial layers results in a significantly greater impact on PPL (perplexity) and recall compared to masking the deeper layers. Our experiments further reveal that this effect is notably mitigated when a short-context base model undergoes long-context extension. In such cases, the model relies on a sparser subset of layers to process positional information; for instance, in the Qwen3-8B(128k) model, significant anomalies are confined solely to the first two layers.See Figure 5 for details.

A.2 Add Noise

To demonstrate that the rapid decline in Perplexity (PPL) is unrelated to error accumulation, we conducted an experiment involving noise injection into the hidden states of specific layers while recording

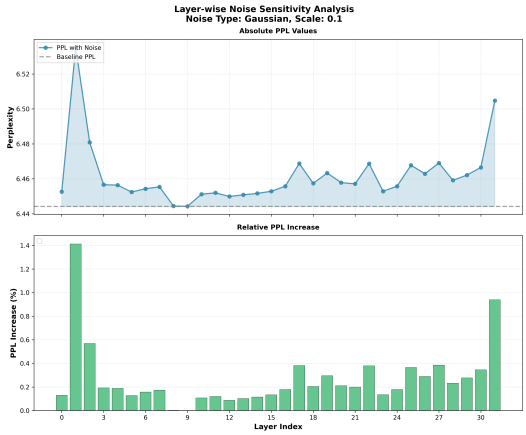

Figure 6: **PPL variations of the LLaMA-3-8B model under layer-wise noise injection**

the resulting changes in PPL. As illustrated in the figure above, the increase in PPL caused by noise injection is generally consistent across most layers. With the exception of the first two layers and the final layer, which show a significant increase, we did not observe phenomena similar to those in our main experiment. This evidence suggests that our observations result from the intrinsic properties of the model rather than layer-wise error accumulation.

A.3 Model Grafting

The algorithmic procedure for our model grafting approach is outlined as follows: In essence, we define a cut-off threshold (or pivot index), denoted as k. For all parameters positioned at or prior to k, we adopt the weights from the base model, whereas for all parameters succeeding k, we integrate the weights from the target model.

It is a prerequisite that these two models share a common lineage; specifically, one model should be derived from the other via Continual Pre-training (CPT) or a similar training regime. Consequently, the models share a substantial portion of their parameter space, with the primary distinction lying in the application of long-context training.

Algorithm 1 Model Grafting

Require: Base model $\mathcal{M}_B$; Long-context model $\mathcal{M}_L$; split layer index $l \in \{0, \ldots, N-1\}$; mode $m \in \{\text{B-Front}, \text{L-Front}\}$
Ensure: Hybrid model $\mathcal{M}_H$

▷ Cache components
Extract $E_B, \{L_B^{(i)}\}_{i=0}^{N-1}, (N_B, H_B)$ from $\mathcal{M}_B$
Extract $E_L, \{L_L^{(i)}\}_{i=0}^{N-1}, (N_L, H_L)$ from $\mathcal{M}_L$
▷ Initialize hybrid model container
Create empty model $\mathcal{M}_H$ with N layers
if $m =$ B-Front **then**
  $\mathcal{M}_H.E \leftarrow E_B$; $\mathcal{M}_H.(N, H) \leftarrow (N_L, H_L)$
  for $i = 0$ to $N-1$ **do**
    $\mathcal{M}_H.L^{(i)} \leftarrow \begin{cases} L_B^{(i)} & i < l \\ L_L^{(i)} & i \geq l \end{cases}$
  end for
else ▷ $m =$ L-Front
  $\mathcal{M}_H.E \leftarrow E_L$; $\mathcal{M}_H.(N, H) \leftarrow (N_B, H_B)$
  for $i = 0$ to $N-1$ **do**
    $\mathcal{M}_H.L^{(i)} \leftarrow \begin{cases} L_L^{(i)} & i < l \\ L_B^{(i)} & i \geq l \end{cases}$
  end for
end if
return $\mathcal{M}_H$

Figure 7: **The loss curves for training the first and last eight layers of the Llama-3-8B model**. It indicate that the training loss of the initial eight layers is consistently lower than that of the final eight layers. This discrepancy arises because long-context extension primarily involves the modification of position-related information, to which shallow layers exhibit a significantly higher sensitivity than deeper layers.

B Training Details

The training details and hyperparameters are summarized in Table 4. Due to resource constraints, particularly regarding computational power, we trained the Llama-3-8B model on 5 billion tokens. This process required approximately 28 hours utilizing 64 NVIDIA A100 (80GB) GPUs. Furthermore, benchmarking for long-context tasks is also significantly resource-intensive.

We present the loss curves for the first 8 layers (referred to here as top-8 layers) and the last 8 layers (bottom-8 layers). As illustrated in the figure 4, the loss for the top-8 layers is consistently lower than that of the bottom-8 layers during training. This disparity arises because long-context extension primarily involves modifying position-related information, to which shallow layers are significantly more sensitive than deeper layers.

Table 4: **Training details and hyperparameters for the Llama-3-8B model.** The model was pre-trained on 5 billion tokens using a distributed setup. The total training process was completed in 28 hours on 64 NVIDIA A100 GPUs.

Hyperparameter	**Value**
Model and Dataset	
Model Architecture	Llama-3-8B
Total Training Tokens	5 Billion
Sequence Length	96384
Optimization	
Peak Learning Rate	3.6×10^{-5}
Learning Rate Scheduler	Cosine Decay
Global Batch Size	32
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.95$)
Infrastructure and Parallelism	
Hardware	64 $\times$ NVIDIA A100 (80GB)
Tensor Parallelism (TP)	8
Pipeline Parallelism (PP)	4
Total Wall-clock Time	28 Hours