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-wise Adaptive Scaling for Position (LASP), a simple layer-dependent adjustment of positional strength. LASP preserves stronger positional signals in shallow layers to stabilize sequence-to-semantics mapping, while attenuating positional influence in deeper layers through a layer-dependent schedule, allowing higher layers to operate with fewer positional restrictions. Experiments on long-context benchmarks show that LASP improves average performance across the evaluated settings.

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 (Beltagy et al., 2020), long-horizon agentic memory (Chhikara et al., 2025), and Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Chen et al., 2023a; Li et al., 2024), the ability to handle extended context windows has become increasingly important.

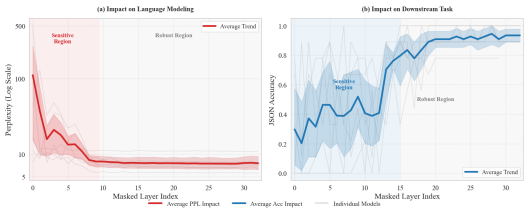


Figure 1: **Layer-wise sensitivity analysis of RoPE across multiple LLMs**. We evaluate the impact of Rotary Positional Embeddings (RoPE) by selectively masking them at individual layers. Using Perplexity (PPL) and Passkey-retrieval as long-context benchmarks, we observe that shallow layers are highly 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 (Vaswani et al., 2017) 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

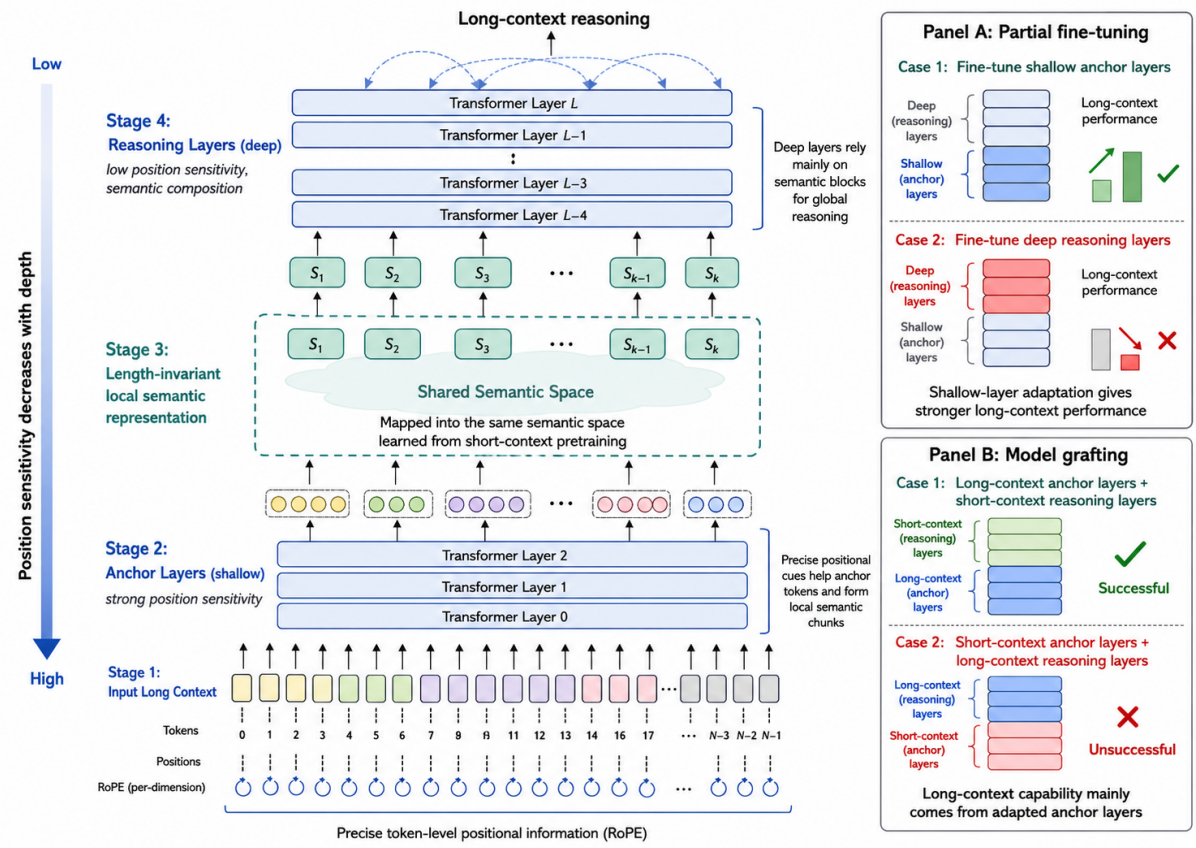


Figure 2: **Anchor-and-reason view of LLM context extension**. Shallow anchor layers use positional cues to organize token-level inputs and form local semantic representations, while deeper layers operate on these representations for broader composition. We evaluate this view through partial fine-tuning and same-lineage grafting probes in Section 4.2.

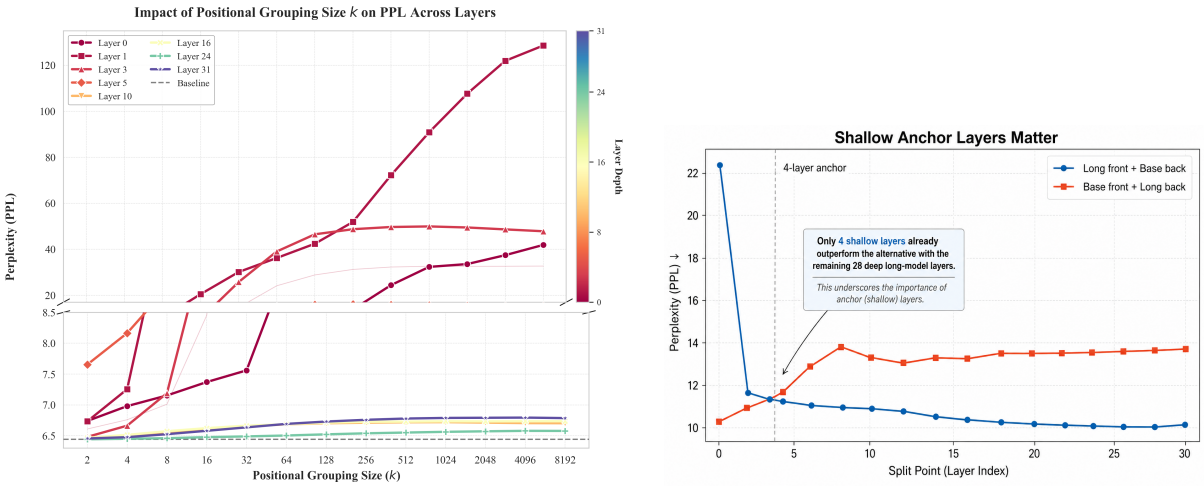
high-frequency extrapolation with low-frequency interpolation. However, most approaches apply the same positional scaling throughout the network, implicitly assuming that all layers rely on positional signals in a similar manner. This design choice does not reflect the layer-wise differences that emerge 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 (Clark et al., 2019; Tenney et al., 2019; Jawahar et al., 2019; Haviv et al., 2022; Liu et al., 2024a; Zhang et al., 2024b; Ju 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



(a) Positional grouping sensitivity

(b) Same-lineage grafting diagnostic

Figure 3: **Additional layer-wise positional sensitivity and grafting diagnostics.** (a) Increasing the positional grouping size sharply degrades PPL in shallow layers, while deeper layers remain close to the baseline. (b) Same-lineage grafting shows that long-context shallow layers reduce PPL when inserted into the short-context model, whereas replacing shallow layers in the long-context model with short-context layers harms PPL. These diagnostics support the layer-wise positional-sensitivity hierarchy used to motivate LASP, but we treat them as complementary evidence rather than complete causal proof.

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 pre-training 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 Adaptive 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 improves average performance across the evaluated settings.

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 pre-training 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. Mechanistic and scaling analyses further study how RoPE frequency choices and content-position coupling affect extrapolation behavior (Liu et al., 2024b; Barbero et al., 2024; Gu et al., 2025). 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, consistent with broader layer-wise analyses of attention, linguistic structure, layer importance, and context knowledge (Clark et al., 2019; Jawahar et al., 2019; Zhang et al., 2024b; Ju et al., 2024). 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., 2024c). 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

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

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

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

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 show that the impact of Rotary Positional Embeddings (RoPE) on long-context performance varies substantially across different layers in a base model. We use these observations to motivate a two-stage mechanism termed "Anchor and Reasoning" for describing how models process long sequences.

4.1 Layer-wise Sensitivity to Positional Information

While prior studies (Tenney et al., 2019; Clark et al., 2019; Jawahar et al., 2019; Zhang et al., 2024b; Ju et al., 2024) 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. We therefore perturb positional information at individual layers and measure the resulting impact on perplexity (PPL) and passkey-recall. For passkey-recall, we insert the “needle” uniformly across 10 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:

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

Here $\mathcal{R}(\cdot)$ denotes the rotation operator. When $\gamma_l = 1$, the layer functions normally; when $\gamma_l = 0$, the position indices are zeroed out, removing the relative distance signal $(m - n)$ for that layer. We summarize the perturbation effect with two layer-wise sensitivity measures:

$$\begin{aligned} \Delta_l^{\text{PPL}} &= \log \text{PPL}_l(\gamma_l = 0) - \log \text{PPL}_l(\gamma_l = 1), \\ \Delta_l^{\text{Rec}} &= \text{Rec}_l(\gamma_l = 1) - \text{Rec}_l(\gamma_l = 0). \end{aligned} \tag{7}$$

Large positive values indicate that masking RoPE at layer l increases uncertainty and reduces retrieval accuracy.

Figure 1 shows that this sensitivity is concentrated in shallow layers: masking their RoPE signals sharply increases PPL and collapses passkey-recall, while the same perturbation in deeper layers has a much smaller effect. The control experiment in Appendix A.2 helps rule out cumulative error as the sole explanation. Across Phi, Llama, Qwen, Mistral, and DeepSeek models from 1B to 24B parameters, we observe the same qualitative pattern.

Figure 3 further stress-tests this hierarchy with positional coarsening and reverse same-lineage grafting. Both diagnostics are consistent with shallow-layer positional sensitivity, and we use them as supporting evidence rather than standalone proof of the mechanism.

4.2 The Anchor-and-Reason Hypothesis

Based on the above observations, we formulate the Anchor-and-Reason hypothesis as a testable layer-wise asymmetry rather than a complete mechanistic proof. For a model with L Transformer layers and split depth K, let the shallow anchor region and upper-layer region be

$$\begin{aligned} \mathcal{A}_K &= \{0, \ldots, K-1\}, \\ \mathcal{U}_K &= \{K, \ldots, L-1\}. \end{aligned} \tag{8}$$

The hypothesis predicts that positional perturbations should have a larger average effect in the anchor region:

$$\begin{aligned} &\mathbb{E}_{l \in \mathcal{A}_K}\left[\Delta_l^{\text{PPL}} + \Delta_l^{\text{Rec}}\right] \\ &\quad > \mathbb{E}_{l \in \mathcal{U}_K}\left[\Delta_l^{\text{PPL}} + \Delta_l^{\text{Rec}}\right]. \end{aligned} \tag{9}$$

Under this view, shallow layers use high-resolution RoPE signals to support position-sensitive local composition. Once useful intermediate representations have been formed, upper layers are expected to rely less on fine-grained positional cues and to operate more robustly under partial positional degradation. This does not prove that upper layers perform all reasoning or that shallow layers alone determine semantic-space alignment; rather, it provides a compact hypothesis connecting layer-wise positional sensitivity to long-context adaptation.

For long-context extension, the practical implication is that a shifted positional distribution can disrupt local composition before higher-level processing receives stable inputs. Effective adaptation should therefore preserve or improve positional handling in the anchor region, while avoiding unnecessary positional constraint in upper layers.

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-3-8B (short body, long-context embeddings)	12.12	46.96	19.20	1.90	62.04	30.89	2.25	32.71	22.35	1.15	63.24	14.59
Fine-tuning												
Anchor-8Layers (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
Reasoning-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. The short-body baseline uses the short-context Llama-3-8B body with embedding/vocabulary-related parameters from the long-context Llama-3.1-8B model. Fine-tuning the first eight shallow/anchor layers improves long-context performance more than tuning the last eight deep/reasoning layers. In model grafting, the hybrid reaches–and at $K = 16$ can exceed–the long-context model, supporting the view that, under same-lineage conditions, long-context anchor layers produce representations compatible with the short-context upper layers.

4.3 Experimental Validation

To probe this hypothesis, we design two targeted experiments. Additional setup details and experimental boundaries are provided in Appendix A.4.

Partial Fine-tuning. We freeze most parameters and update only selected transformer modules to test whether position-sensitive shallow layers are more effective adaptation targets for long-context training. For a selected trainable set $\mathcal{T}$, partial fine-tuning optimizes

$$\Theta^{\star}_{\mathcal{T}} = \arg\min_{\Theta_{\mathcal{T}}} \mathcal{L}_{\mathrm{LC}}(\Theta), \qquad \Theta_{\overline{\mathcal{T}}} \text{ frozen}, \quad (10)$$

where $\mathcal{L}_{\mathrm{LC}}$ is the long-context continued-training loss. We compare two trainable sets in Llama-3-8B:

$$\begin{aligned} \mathcal{T}_{\text{anchor}} &= \{\theta^{(l)} : 0 \le l < 8\}, \\ \mathcal{T}_{\text{deep}} &= \{\theta^{(l)} : L-8 \le l < L\}. \end{aligned} \quad (11)$$

The first eight layers are selected from the most position-sensitive region in Figure 1, while the last eight layers serve as a less position-sensitive deep-layer control.

Model Grafting. As a complementary structural probe, we construct zero-shot hybrids between a short-context model Θ_{Short} and a same-lineage long-context model Θ_{Long}. For a split depth K, the first K transformer blocks and early input-facing components come from Θ_{Long}, while the remaining transformer blocks and output-side components come from Θ_{Short}. All grafted models use the long-context RoPE configuration globally rather than mixing positional configurations by layer:

$$\begin{aligned} \Theta^{(K)}_{\text{Hybrid}} = \big(&\mathcal{C}^{\text{in}}_{\text{Long}}, \{\theta^{(l)}_{\text{Long}}\}_{l=0}^{K-1}, \\ &\{\theta^{(l)}_{\text{Short}}\}_{l=K}^{L-1}, \mathcal{C}^{\text{out}}_{\text{Short}}; \mathcal{R}_{\text{Long}}\big), \end{aligned} \quad (12)$$

where $\mathcal{C}^{\text{in}}$ denotes input-facing non-block components, $\mathcal{C}^{\text{out}}$ denotes output-side non-block components, and $\mathcal{R}_{\text{Long}}$ denotes the global long-context RoPE configuration.

We report the validation results on HELMET (Yen et al., 2025), a comprehensive benchmark for long-context performance. As shown in Table 1, fine-tuning the first eight shallow/anchor layers yields stronger long-context performance than fine-tuning the last eight deep/upper layers, providing evidence that the anchor layers' handling of positional information is important for long-context extension.

In the model-grafting experiments, using the first 16 long-context anchor layers matches–and in several metrics surpasses–the Llama-3.1-8B baseline. Under the same-lineage condition, this suggests that representations produced by long-context anchor layers remain compatible with the short-context upper layers. The reverse direction is examined as a PPL diagnostic in Figure 3(b); we treat it as supporting evidence rather than a complete proof of the proposed mechanism.

5 LASP: Layer-wise Adaptive Scaling for Position

LASP operationalizes the anchor-and-reason view by assigning RoPE positional strength on a per-layer basis. The goal is to preserve high-resolution positional cues in shallow anchor layers while re-

Panel A: HELMET synthetic long-context benchmark										
Extension	**Base Config**	**Layer Base**	**Layer Scale**	**Avg Score**	**Recall**	**RAG**	**Long-QA**	**Cite**	**ICL**	**Rerank**
YaRN	500k	–	–	50.84	97.54	62.06	**35.18**	4.55	69.35	36.35
	2M	–	–	42.59	76.60	53.87	26.83	2.86	64.60	30.79
	500k–2M	✓	–	50.04	82.50	**62.58**	33.28	5.05	**75.47**	**41.35**
	500k	–	✓	51.73	98.95	61.73	33.18	3.94	73.47	39.11
	500k–2M	✓	✓	51.99	99.22	61.05	33.67	**6.10**	72.43	39.44
LongRoPE	Search	–	–	51.02	98.80	62.30	35.00	4.40	69.00	36.60
LongRoPE	**Search**	✓	✓	**52.28**	**99.50**	61.80	35.10	5.60	72.00	39.70
Panel B: LOFT real-world retrieval benchmark										
Extension	**Base Config**	**Layer Base**	**Layer Scale**	**Avg Score**	**ArguAna**	**FEVER**	**HotpotQA**	**NQ**	**Quora**	**SciFact**
YaRN	500k	–	–	62.0	10.0	78.0	44.50	85.0	90.0	66.0
	2M	–	–	64.0	10.0	89.0	43.50	93.0	89.0	57.0
	500k–2M	✓	–	66.0	17.0	89.0	46.0	96.0	90.0	60.0
	500k	–	✓	64.0	**18.0**	82.0	44.0	91.0	87.0	64.0
	500k–2M	✓	✓	69.0	14.0	92.0	47.5	97.0	**96.0**	66.0
LongRoPE	Search	–	–	65.3	12.0	87.0	45.0	94.0	90.0	64.0
LongRoPE	**Search**	✓	✓	**69.5**	15.0	**93.0**	**48.0**	**98.0**	**96.0**	**67.0**

Table 2: **Main results on HELMET and LOFT.** Extension denotes the underlying context-extension method, Base Config denotes its global or searched base/frequency setting, and Layer Base / Layer Scale indicate whether LASP applies layer-wise base or scale scheduling. Panel A reports HELMET synthetic long-context results across Recall, RAG, Long-QA, Cite, ICL, and Rerank. Panel B reports LOFT real-world retrieval results across ArguAna, FEVER, HotpotQA, NQ, Quora, and SciFact. Avg denotes the mean score across the corresponding tasks.

ducing unnecessary positional constraint in deeper reasoning layers.

5.1 LASP: Layer-Aware Scaling and RoPE Base Scheduling

Conventional context extension applies one global scaling rule to all layers. LASP instead uses layer-dependent scale and base schedules:

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

$$\theta_d^{(l)} = \mathcal{B}(l)^{-2d/D}, \tag{13}$$

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 $\mathcal{S}(l)$:

$$\tilde{\theta}_d^{(l)} = \frac{\theta_d^{(l)}}{\mathcal{S}(l)} = \frac{\mathcal{B}(l)^{-2d/D}}{\mathcal{S}(l)}. \tag{14}$$

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

We designate the first K Transformer layers as anchor layers and supply them with the finest positional granularity. Scale increases progressively within this region. For the RoPE base, we keep $b_{\min}$ in shallow layers and increase the base only after the anchor region, preserving early-layer positional resolution while allowing deeper layers to use smoother long-range signals.

Piecewise linear scaling schedule. With anchoring depth L_{anchor}, we ramp from $s_{\min}$ to $s_{\max}$ and then keep the scale fixed:

$$\lambda_s = \frac{s_{\max} - s_{\min}}{L_{\text{anchor}}},$$
$$\mathcal{S}(l) = \begin{cases} s_{\min} + l \cdot \lambda_s, & l < L_{\text{anchor}}, \\ s_{\max}, & l \geq L_{\text{anchor}}. \end{cases} \tag{15}$$

Layer-wise RoPE base schedule. For the RoPE base, shallow layers keep $b_{\min}$, while deeper layers linearly move toward $b_{\max}$ across L_{tot} layers:

$$\lambda_b = \frac{b_{\max} - b_{\min}}{L_{\text{tot}} - L_{\text{anchor}}},$$
$$\mathcal{B}(l) = \begin{cases} b_{\min}, & l < L_{\text{anchor}}, \\ b_{\min} + \lambda_b(l - L_{\text{anchor}}), & l \geq L_{\text{anchor}}. \end{cases} \tag{16}$$

5.2 Experiments and Results

We evaluate whether the layer-wise positional schedule motivated by Section 4 improves long-context adaptation beyond uniform RoPE scaling. All main experiments use `meta-llama/Meta-Llama-3-8B` as the base model and continue training on 5B Common Crawl tokens. YaRN is the primary context-extension method because it provides a controlled setting for varying the RoPE base, and LongRoPE is included

to test whether LASP can complement a stronger dimension-wise extension baseline. Training-data preprocessing and evaluation-protocol details are reported in Appendix B.

The experiments are organized around three questions. First, does simply increasing the global RoPE base improve all long-context behaviors? Second, do the two LASP components–layer-wise base scheduling and layer-wise scale scheduling–contribute different benefits? Third, can the same layer-wise schedule be combined with LongRoPE rather than only improving a YaRN configuration? We evaluate these questions on HELMET (Yen et al., 2025), which covers synthetic long-context skills, and LOFT (Lee et al., 2024), which emphasizes real-world retrieval scenarios. We set $s_{\max} = 16$, $b_{\min} = 500{,}000$, and $b_{\max} = 2{,}000{,}000$, and compare YaRN-500k, YaRN-2M, LASP(base), LASP(scale), LASP(both), LongRoPE, and LASP+LongRoPE.

Panel A of Table 2 shows that a uniformly larger RoPE base is not sufficient on HELMET: YaRN-2M is worse than YaRN-500k on every reported category. This supports the concern raised by the anchor-and-reason analysis that uniformly smoothing positional signals can harm tasks requiring precise positional structure. LASP mitigates this trade-off by keeping stronger positional resolution in the anchor region while allowing later layers to use smoother long-range signals. The component results are complementary rather than identical: LASP(scale) gives the strongest YaRN-based Recall score, LASP(base) is strongest on RAG, ICL, and Rerank, and LASP(both) obtains the best YaRN-based average and Cite score. When the same layer-wise schedule is applied on top of LongRoPE, LASP+LongRoPE achieves the best HELMET average in the table. The gains remain task-dependent: YaRN-500k is still strongest on Long-QA, and no single LASP variant dominates every subtask.

Panel B gives a different but compatible picture on LOFT. Unlike HELMET, YaRN-2M does not uniformly degrade, suggesting that these real-world retrieval tasks may rely more on semantic matching than on fine-grained positional resolution. Even in this setting, the layer-wise schedules provide useful adjustments. LASP(base) improves the YaRN-based average more than LASP(scale), while LASP(both) further improves the average and reaches the strongest YaRN-based scores on FEVER, HotpotQA, NQ, and Quora. Combining LASP with LongRoPE yields the best LOFT average and the best FEVER, HotpotQA, NQ, and SciFact scores, indicating that layer-wise positional scheduling can act as a complementary adjustment rather than a replacement for dimension-wise context extension.

Overall, the results support the design principle behind LASP: preserving precise positional information in shallow anchor layers while relaxing positional constraints in upper layers can improve average long-context performance. At the same time, the improvements are not uniform across tasks, so we interpret LASP as a mechanism-guided adaptation strategy rather than a claim of universal dominance.

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 important for effective length generalization. We study this hypothesis through finetuning studies and model-grafting experiments. Guided by this view, we further introduce **LASP**, a dynamic strategy that assigns layer-specific positional configurations. Long-context benchmarks show average improvements across the evaluated settings.

Limitations

Our experiments are designed to isolate the effect of layer-wise positional schedules under a controlled training budget rather than to claim state-of-the-art long-context performance. The training corpus is limited to 5B Common Crawl tokens, and we do not perform explicit document-level deduplication or benchmark decontamination. These choices make the setup transparent but leave stricter contamination control and larger-scale training for future work. In addition, although LASP is compatible with other scaling-based approaches, YaRN is the only primary scaling strategy used in our ablation studies. We also do not exhaustively tune layer-wise schedules; instead, we adopt a simple progressive schedule shared across layers.

References

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

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

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Preprint*, arXiv:2004.05150.

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.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

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.

Zihan Gu, Ruoyu Chen, Han Zhang, Hua Zhang, and Yue Hu. 2025. Deconstructing positional information: From attention logits to training biases. *Preprint*, arXiv:2505.13027. Accepted by ICLR 2026.

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.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. 2024. How large language models encode context knowledge? a layer-wise probing study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8235–8246, Torino, Italia. ELRA and ICCL.

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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33.

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. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xiaoran Liu, Hang Yan, Chenxin An, Xipeng Qiu, and Dahua Lin. 2024b. Scaling laws of rope-based extrapolation. In *The Twelfth International Conference on Learning Representations*.

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

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.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint*.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

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*.

Yang Zhang, Yanfei Dong, and Kenji Kawaguchi. 2024b. Investigating layer importance in large language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 469–479, Miami, Florida, US. Association for Computational Linguistics.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, Zhangyang Wang, and 1 others. 2024c. 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 Sensitivity

A.1 Mask RoPE

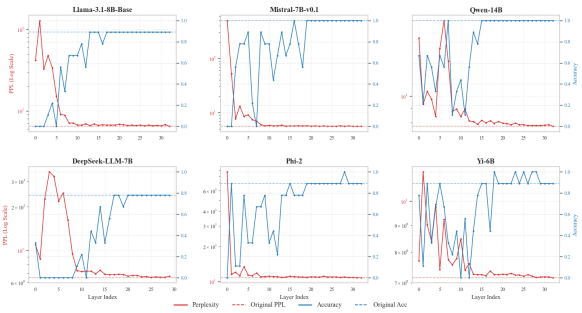


Figure 4: **Performance visualization of masking RoPE in specific layers across different models**. While evident PPL deviations occur in shallow layers, PPL and recall in deeper layers align closely with the unmasked configuration. For models with fewer parameters, specifically Phi-2, targeted RoPE 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 (Rae et al., 2019).

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 4 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

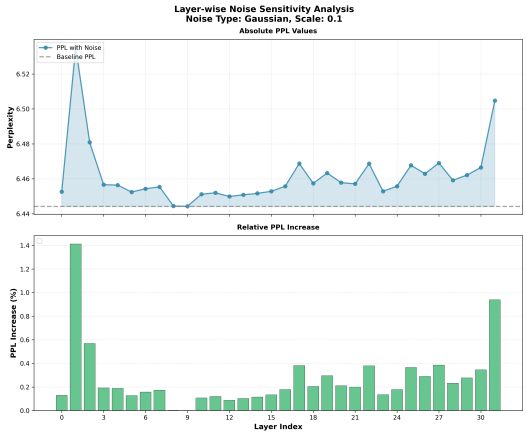


Figure 5: **PPL variations of the LLaMA-3-8B model under layer-wise noise injection**

the hidden states of specific layers while recording 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 Additional Positional Sensitivity and Grafting Diagnostics

We include two diagnostics that complement the main RoPE-masking result and the mechanism-validation table. First, positional coarsening tests whether layers require fine-grained RoPE resolution rather than only the presence or absence of positional encoding. For a grouping factor $k \in \mathbb{Z}^+$, we transform the position index as

$$\tilde{m} = \phi_k(m) = \left\lfloor \frac{m}{k} \right\rfloor . \tag{17}$$

This transformation reduces positional resolution to $1/k$: tokens within the same group receive identical RoPE position indices, while relative ordering between groups remains available.

As shown in Figure 3(a), increasing k sharply degrades PPL in shallow layers, while deeper layers remain close to the baseline even under substantial coarsening. This pattern is consistent with the view that shallow layers depend more strongly on fine-grained local positional resolution.

Second, we run a reverse same-lineage grafting diagnostic by inserting short-context shallow layers

into the long-context model. Figure 3(b) shows that this reverse direction harms PPL, whereas inserting long-context shallow layers into the short-context model reduces PPL. Because corresponding HELMET values are not available for this diagnostic, we use it only as a PPL-based supporting check.

A.4 Additional Details for Mechanism Validation

Partial fine-tuning setup. The partial fine-tuning probe starts from `meta-llama/Meta-Llama-3-8B` with YaRN enabled and performs long-context continued fine-tuning. To reduce compute cost, the target context length for this probe is 64k rather than the 128k setting used in the main long-context training runs. The training data source is the same Common Crawl corpus used in the main experiments, but sequences are truncated to 64k tokens. The token budget remains 5B tokens. Except for this shorter target length, the optimizer, peak learning rate, global batch size, scheduler, and parallelism configuration follow the training setup in Table 4.

Only the selected transformer layers are trainable. Within those layers, we unfreeze the self-attention and MLP parameters and keep all other parameters frozen. This is a dense parameter update of the original modules, not a LoRA or adapter-based setup. We compare exactly two settings: the first eight layers and the last eight layers. The first eight layers are selected because they are the most sensitive region in the PPL sensitivity analysis, while the last eight layers provide a relatively less position-sensitive deep-layer control. We do not train additional variants with 4, 16, or 24 layers. Checkpoint selection uses the final checkpoint, and each setting is run once with a single seed because long-context fine-tuning is compute-intensive.

Model grafting setup. Model grafting uses two same-lineage Llama models: a Llama-3 32k model and a Llama-3.1 128k model. The two models share the same architecture and common lineage, including layer count, hidden size, tokenizer, and vocabulary. We perform no additional training after grafting; all grafted models are evaluated zero-shot.

For a graft depth K, the Anchor-KLayers model copies the first K shallow transformer blocks from the long-context model and the remaining transformer blocks from the short-context model:

$$\theta_H^{(l)} = \begin{cases} \theta_L^{(l)}, & 0 \le l < K, \\ \theta_S^{(l)}, & K \le l < L. \end{cases} \qquad \mathcal{R}_H = \mathcal{R}_L, \tag{18}$$

where $\theta_L^{(l)}$ and $\theta_S^{(l)}$ denote the l-th transformer block from the long- and short-context models, respectively, and $\mathcal{R}_L$ denotes the long-context RoPE and context configuration. RoPE, YaRN, and other long-context configuration choices are applied globally to the hybrid model; we do not mix positional configurations on a per-layer basis. Non-transformer-block parameters are handled according to their computational position: early input-facing components such as token embeddings are copied from the long-context model, while output-side components such as the final normalization and language-model head are copied from the short-context model.

Algorithm 1 Anchor-Layer Model Grafting

Require: Short-context model $\mathcal{M}_S$; long-context model $\mathcal{M}_L$; graft depth $K \in \{8, 16, 24\}$
Ensure: Zero-shot hybrid model $\mathcal{M}_H^{(K)}$

```
Assert that M_S and M_L have the same layer count, hidden size, tokenizer, and vocabulary.
Initialize M_H^(K) with the long-context RoPE and context configuration from M_L.
Copy input-facing components, including token embeddings, from M_L.
for l = 0 to L − 1 do
    if l < K then
        M_H^(K).θ^(l) ← M_L.θ^(l)
    else
        M_H^(K).θ^(l) ← M_S.θ^(l)
    end if
end for
Copy output-side components, including final normalization and language-model head, from M_S.
return M_H^(K) without additional training.
```

Controls and interpretation. The short-body baseline in Table 1 is a sanity baseline rather than the main grafting construction: it keeps the short-context transformer body and changes only embedding or vocabulary-related parameters to their long-context counterparts. We also run the reverse grafting direction by inserting short-context shallow layers into the long-context model. This reverse direction is substantially worse in PPL-based

Probe	Model source	Modified components	Context and evaluation	Boundary
Partial fine-tuning	Llama-3-8B with YaRN continued fine-tuning on Common Crawl	Self-attention and MLP parameters in either the first 8 or last 8 layers; all other parameters frozen	64k target length, 5B tokens, HELMET reporting, final checkpoint	Dense updates only; no LoRA/adapters; single run and single seed; no 4/16/24-layer variants
Model grafting	Same-lineage Llama-3 32k and Llama-3.1 128k models with matched architecture and tokenizer	First K shallow blocks and input-facing components from the long-context model; remaining blocks and output-side components from the short-context model	Global long-context RoPE/configuration, zero-shot HELMET reporting for Anchor-8/16/24Layers	No post-grafting training; no per-layer mixed RoPE configuration; interpretation relies on same-lineage compatibility

Table 3: Structured setup summary for the mechanism-validation probes. The table reports reproducibility-relevant choices rather than additional results.

diagnostics, as summarized in Appendix A.3; because corresponding HELMET values are not available for this diagnostic, we do not add a separate numerical result table. Overall, the grafting results indicate that, under same-lineage conditions, long-context anchor layers can produce representations that remain compatible with the short-context upper layers. This supports the semantic-space compatibility view, but it does not by itself fully establish the anchor-and-reason mechanism.

B Training Details

The training details and hyperparameters are summarized in Table 4. Due to resource constraints, particularly regarding computational power, we conducted long-context continued training for 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.

Data preprocessing and evaluation protocol. The training corpus contains 5B tokens from Common Crawl. We retain documents or sequences longer than 128k tokens and use the tokenizer corresponding to the base model, i.e., the Llama-3 tokenizer for Llama-3-8B. We do not use sequence packing; sequences longer than the training length are truncated to 128k tokens. Since all retained training samples satisfy the length requirement, there are no short sequences requiring padding or length-based completion. We do not apply additional explicit deduplication or benchmark decontamination beyond this filtering pipeline; we therefore treat stricter contamination control as a limitation rather than as an experimental claim.

For HELMET and LOFT, we follow the official benchmark protocols and default versions, using their default prompt templates and task-specific official evaluators or scorers. Unless otherwise specified, generation uses greedy decoding. For LOFT, we evaluate the commonly used retrieval subset reported in Panel B of Table 2. We do not introduce custom prompts or additional evaluators.

Figure 6: **The loss curves for training the first and last eight layers of the Llama-3-8B model**. The curves show that the training loss of the initial eight layers is consistently lower than that of the final eight layers. This pattern is consistent with the view that long-context extension primarily modifies position-related information, to which shallow layers exhibit higher sensitivity than deeper layers.

We present the loss curves for the first eight layers (the shallow/anchor layers) and the last eight layers (the deep/reasoning layers). As illustrated in Figure 6, the loss for the first eight layers is consistently lower than that of the last eight layers during training. This pattern is consistent with the hypothesis that long-context extension primarily involves modifying position-related information, to which shallow layers are more sensitive than deeper layers.

Table 4: **Training details and hyperparameters for the Llama-3-8B model.** The model underwent long-context continued training 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	
Base Model	`meta-llama/Meta-Llama-3-8B`
Total Training Tokens	5 Billion
Sequence Length	128k
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