Adaptive Zooming via Relevance-Informed Positional Resource Allocation for Training-free LLM Context Extension

Hongbo Zhao[1,*], Huibin Wang[1,*], Bin Tang[1,*], Xianming Hu[1], Yihong Huang[1], Yijun Shen[1], Nuoyi Chen[2], Ping Li[3], Kai Zhang[1,†]

[1]School of Computer Science and Technology, East China Normal University,
[2]Institute of Science and Technology for Brain-inspired intelligence, Fudan University,
[3]School of Computer Science and Software Engineering, Southwest Petroleum University

Correspondence: 51275901107@stu.ecnu.edu.cn, kzhang@cs.ecnu.edu.cn

Abstract

Large Language Models exhibit degraded performance when extrapolating beyond training context lengths. Existing training-free methods like positional reuse or interpolation can alleviate this issue in an efficient manner. However, these strategies are semantics-agnostic by only considering relative token distances, which could indiscriminately blur semantically relevant and irrelevant tokens alike. To address this, we introduce an adaptive positional zooming method called **Relevance-Informed Positional Resource Allocation (RiPRA)**. RiPRA formulates positional encoding as a constrained resource allocation, in which a fixed positional budget is distributed across tokens in a longer context based on their semantic relevance to the query: relevant tokens get higher positional resolution, while irrelevant tokens (positions) are compressed. By doing this, RiPRA enables a dynamic and nonparametric positional zooming where the positional resolution is adaptively modulated across queries and network layers, effectively improving long-range context modeling and retrieval capacity. Besides, an isotonic smoothing is used to further enforce a global linear ordering relationship to preserve stability and generalization, together with a chunk-based hierarchical approximation to further reduce inference overhead. Extensive experiments demonstrate that RiPRA consistently outperforms existing training-free extrapolation methods, showing the value of relevance-conditioned positional encoding for long-context generalization.

1 Introduction

Large language models (LLMs) have become indispensable for a diverse array of natural language processing tasks (Yang et al., 2025; Tay et al., 2021; Kryscinski et al., 2022; Zhong et al., 2022; Zheng et al., 2023). As the demand for richer contextual understanding and extended interactions grows, it has become imperative to increase the lengths of input and output sequences. Although improvements in model architectures and positional encoding methods (Ruoss et al., 2023a; Liu et al., 2024) have partially advanced the ability to handle longer sequences, the performance of LLMs remains limited by the context length established during pre-training (Xiao et al., 2023; Han et al., 2024). When inputs exceed this range, models encounter out-of-distribution positional patterns, resulting in significant performance degradation. While fine-tuning on longer sequences (Chen et al., 2023b; Ruoss et al., 2023b; Li et al., 2024; Ding et al., 2024) offers a potential solution, it is computationally expensive, relies on scarce high-quality long-context data, and may degrade performance on short-context tasks (Gao et al., 2025).

To circumvent these limitations, training-free methods have emerged as a promising alternative. These approaches extend the effective context length by remapping inference-time positions back into the pretraining positional regime, without any additional learning process to adjust the model parameters. They typically fall into two categories. *Positional reuse strategies*, such as SelfExtend (Jin et al., 2024a) and ChunkLlama (An et al., 2024b), expand the context window reusing pre-trained positional slots for tokens beyond the original range. AdaGroPE (Xu et al., 2025) further increases the frequency of positional reuse as token distance grows, adapting to long-term attention decay phenomena in the rotary position embeddings (RoPE) (Su et al., 2024). *Interpolation-based methods*, including NTK-aware scaling (bloc97, 2023; emozilla, 2023) and YaRN (Peng et al., 2023), compress long-range positions into the pre-training window by non-uniformly rescaling the frequency spectrum of RoPE, while GALI (Li et al., 2025) performs interpolation at both the positional and attention-logit levels.

[*]These authors contributed equally.
[†]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 21067–21083
July 2-7, 2026

Despite strong empirical performance, existing training-free strategies remain content-agnostic. They typically use a predefined function to map token positions based on relative distances, ignoring token semantics or inter-token relationships. Such strategies are suboptimal, as they can uniformly compress positions by blurring semantically salient and irrelevant tokens alike. This constrains LLMs' ability to identify key evidence from long contexts for accurate reasoning and language modeling.

In this paper, we investigate how to *effectively coordinate positional encoding with semantic information in a training-free manner*. This problem is inherently challenging and remains largely underexplored, as the absence of additional learning requires the relationship between positional and semantic components to be specified *a priori* as an explicit inductive bias, rather than learned from data. Consequently, although several studies have successfully integrated positional encoding with semantic embeddings through supervised learning (Golovneva et al., 2024; Wang et al., 2025), these approaches typically rely on data-driven optimization to mitigate potential discrepancies and refine the alignment between positional and semantic components. As a result, their designs are not directly transferable to training-free settings.

To solve this, we propose **Relevance-Informed Positional Resource Allocation (RiPRA)**, which is a training-free approach for content-aware context extension. Rather than relying on a fixed, content-agnostic positional mapping, RiPRA introduces adaptive positional zooming that dynamically modulates positional resolution according to the semantic landscape of the input sequence.

Specifically, RiPRA formulates (relative) positional encoding as a *constrained resource allocation* problem, where a finite positional budget is distributed across sequences substantially longer than the pretraining context. Semantically salient regions relevant to the current query are assigned higher positional resolution (i.e., reduced index reuse), while less informative segments are compressed to conserve resources. As a result, positional encoding is determined in a flexible, nonparametric manner, adapting to both the query and the layer-wise processing dynamics. To the best of our knowledge, this is among the first attempts to exploit semantic information in training-free methods for LLM context extension.

Overall, RiPRA provides a useful mechanism for improving long-context modeling and representation capacities of LLMs, and our main contributions are summarized as follows:

• We propose **RiPRA**, a new positional encoding scheme that achieves adaptive zooming by modulating positional resolutions based on the token relevance landscape while simultaneously respecting the global linear ordering relationship.

• We provide an efficient, end-to-end implementation that combines inner embedding representations of Transformers with chunk-based hierarchical approximation to minimize inference overhead without auxiliary models.

• We evaluate RiPRA on several long-context benchmarks (LongBench, L-Eval, and Passkey Retrieval), and report consistent improvements over SOTA training-free models for context extension.

2 Related Work

2.1 Position Encoding

Positional information is essential for modeling token relationships (Vaswani et al., 2017; Dufter et al., 2022; Kazemnejad et al., 2023) and is typically incorporated as either absolute position embeddings or relative positional biases. *Absolute position encoding* assigns position-specific vectors to each token (e.g., sinusoidal or learned embeddings), which are commonly added to token embeddings. In contrast, *rotary position encoding* (RoPE) (Su et al., 2024) rotates the query and key representations by position-dependent phases, so that attention scores depend primarily on relative distance between token pairs. This rotary scheme has been widely used in modern LLMs like Llama (Touvron et al., 2023).

As discussed, some supervised settings allow token embeddings and positional encodings to be combined in a learnable manner, and they have shown promising results in long-context modeling based on the RoPE framework, such as CoPE (Golovneva et al., 2024), and TAPA (Wang et al., 2025). These methods learn new parametric couplings between content and position during training, whereas our goal is to recover and exploit content–position interactions already implicit in a frozen LLM's inference dynamics, without introducing new parameters or supervision.

2.2 Training-free Extrapolation of RoPE

Extrapolating RoPE beyond the pre-training window induces positional O.O.D. effects and degrades long-context performance (Chen et al., 2023a;

Chowdhury and Caragea, 2023; Chen et al., 2024). Training-free methods address this by modifying relative positions in the rotary transform, mainly via (i) *position reuse*, which remaps out-of-window offsets to in-window values using segment-wise constant zooming (Jin et al., 2024b; Xu et al., 2025) or periodic token grouping (An et al., 2024b; Li et al., 2025), and (ii) *interpolation-based compression*, which applies continuous frequency-based zooming to compress long-range rotation angles back into the pretraining range (Chen et al., 2023a; bloc97, 2023; emozilla, 2023; Peng et al., 2023; Li et al., 2025). Our method integrates continuous interpolation with rounding-based discrete remapping and leverages semantic relevance to adaptively modulate positions, bridging distance-based extrapolation and token-level semantics.

3 Method

3.1 Problem Definition

Let $\tilde{L}$ denote the maximum context length supported during pre-training, and let L be the target context length at inference time, where $L > \tilde{L}$. Our objective is to learn a positional mapping function $\mathcal{P}$ that maps the index of each token i—defined as its relative distance to the query placed at the origin—onto a position $\tilde{i}$ within the available positional budget:

$$\tilde{i} = \mathcal{P}(i), \quad \mathcal{P} : [0, L] \to [0, \tilde{L}]. \tag{1}$$

Constructing $\mathcal{P}$ can be interpreted as allocating a finite positional budget $\tilde{L}$ across the L token indices. To formalize this, we define the discrete derivative of $\mathcal{P}$ at token index i as

$$g(i) = \frac{\Delta\mathcal{P}(i)}{\Delta i} = \mathcal{P}(i) - \mathcal{P}(i-1). \tag{2}$$

By definition, $g(i)$ represents local positional resolution around token i: larger values ($g(i) \to 1$) correspond to finer resolution, effectively separating the token from its neighbors, whereas smaller values ($g(i) \to 0$) compress the token with neighbors, leading to minimal resolution.

Since the total positional budget is fixed, the derivatives across all L positions must satisfy

$$\texttt{constraints:} \quad \sum_{i=1}^{L} g(i) = \tilde{L}, \quad g(i) \in [0, 1]. \tag{3}$$

For convenience, we denote the sequence of positional derivatives by the vector

$$\mathbf{g} = [g(1), \ldots, g(L)] \in \mathbb{R}^{L\times 1}.$$

Under this formulation, the core idea of RiPRA is to allocate positional resolutions $\mathbf{g}_i$ under the global budget and numerical constraints, while prioritizing tokens based on their importance. In particular, task-relevant and semantically significant tokens are assigned higher resolutions, whereas less relevant tokens are compressed. Additionally, global structural priors and normalization are incorporated to enhance stability and generalization. Overall, as illustrated by Figure 1, our method proceeds in three main steps:

(1) Constructing relevance landscape. We construct a relevance landscape that reflects the distribution of contextual importance by measuring the semantic similarity between the token at each position i and the current Query.

(2) Positional resource allocation. We assign positional derivative $\mathbf{g}_i$'s proportional to the semantic relevance scores $\mathcal{R}_i$'s., so that salient contexts are granted high resolution, whereas low-relevance regions are compressed.

(3) Post-processing. To further refine the positional encoding, we apply isotonic regression to guarantee a monotonic derivative decay, thus preserving global linear order relationships. Then we perform a rescaling to make sure that the sum of all positional derivatives equals the budget $\tilde{L}$.

3.2 Relevance Landscape Construction

In order to quantify the importance of each token in the context window to the current focus of the generative process, we resort to the semantic relevance between each token and the query. Specifically, for each Transformer layer, we take the pre-position projections of the query $\tilde{\mathbf{q}}$ and the tokens $\tilde{\mathbf{k}}$'s and compute their dot product. Since positional information has not yet been injected, this score is purely content-driven and captures semantic compatibility rather than proximity effects. We use these semantic relevance scores as the basis for estimating which regions of the context are more relevant to deserve a higher positional resolution.

To improve efficiency, we compute relevance scores at the chunk level, and assume that all tokens within the same chunk share the same positional derivative. Assume we have a sequence of key vectors $\mathbf{k}_i$ for tokens $i = 1, 2, ..., L$. By using a chunk size of S tokens, we segment the sequence into a number of chunks C_j's for $j = 1, 2, ..., N$, with $N = \lceil L/S \rceil$. The fingerprint for C_j is computed

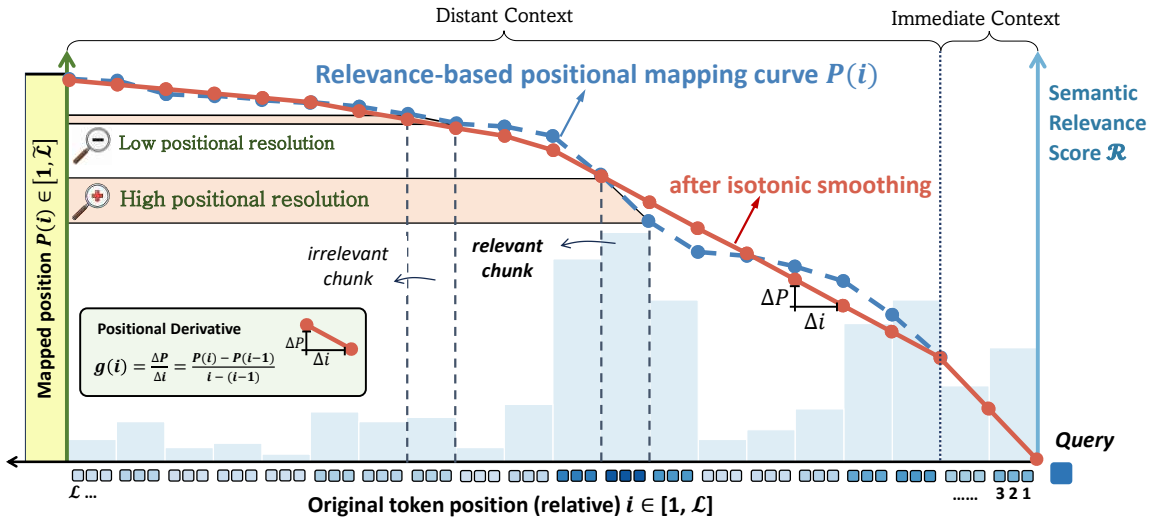

Figure 1: Illustration of the Relevance-Informed Positional Resource Allocation (RiPRA).

by pooling the key vectors from that chunk,

$$\mathbf{c}_j = \frac{1}{|C_j|} \sum_{i \in C_j} \mathbf{k}_i, \tag{4}$$

with $|C_j|$ the size of the i-th chunk. The resulting chunk fingerprints $\mathbf{c} = \{\mathbf{c}_1, \ldots, \mathbf{c}_N\} \in \mathbb{R}^{N \times D}$ will be cached for reuse in the decoding steps. Throughout this paper, we use i to denote the token index and j to denote the chunk index.

We compute the semantic relevance between each chunk and the query $\tilde{\mathbf{q}}$ using inner product. For efficiency, the inner products computed from each individual attention head are aggregated together as a unified, comprehensive relevance score, so as to reflect the relevance of chunk j across different semantic subspaces, as

$$s_j = \frac{1}{H} \sum_{h=1}^{H} \left(\tilde{\mathbf{q}}^{(h)} \cdot (\mathbf{c}_j^{(h)})^\top \right), \tag{5}$$

where $\tilde{\mathbf{q}}^{(h)}$ is query vector, and $\mathbf{c}_j^{(h)}$ is the fingerprint of chunk C_j, both from the h-th head.

To transform s_j's into a standardized signals for adjusting the zooming factors in $[0, 1]$, we apply min-max normalization:

$$\mathcal{R}_j = \frac{s_j - \min(\{s_t\}_{t=1}^N)}{\max(\{s_t\}_{t=1}^N) - \min(\{s_t\}_{t=1}^N) + \epsilon}, \tag{6}$$

where ϵ is a small constant (e.g., $1e^{-6}$) ensuring numerical stability. The resulting set $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_N\}$ constitutes the **Relevance Landscape**, which maps the distribution of semantic relevance across the entire context.

3.3 Positional Derivative Assignment and Post-processing

The context relevance landscape $\mathcal{R}$ serves as pivotal guidance for allocating positional resolution, enabling non-uniform "zooming" across context positions. Intuitively, regions with high relevance scores, representing tokens critical for the current focus, warrant higher local positional resolution to preserve information fidelity. Conversely, low-relevance regions should be compressed to conserve the positional budget.

Based on this intuition, we adopt the semantic relevance score $\mathcal{R}_j$ as an initial (unnormalized) estimate of the **chunk-level positional derivative** $\tilde{G}(j)$ for the j-th chunk, defined as:

$$\tilde{G}(j) = \mathcal{R}_j + \epsilon, \tag{7}$$

where ϵ is a small constant introduced to ensure numerical stability. For convenience, $\tilde{G}(j)$ can be concatenated as a vector $\tilde{\mathbf{G}} = [\tilde{G}(1), \ldots, \tilde{G}(N)]$.

3.3.1 Smoothing via Isotonic Regression

While adapting positional resolution based on relevance scores enhances semantic flexibility, the primitive global linear ordering relation among tokens remains critical for stable positional encoding. To preserve this structural inductive bias, we impose an additional constraint requiring contextual importance to decay monotonically as tokens become increasingly distant from the query. This simple constraint injects a global positional signal as an essential complement to semantic relevance information.

We employ the following isotonic smoothing formulation to fulfill the above concept, as follows

$$\min_{\mathbf{G}\in\mathbb{R}^N} \sum_{j=1}^{N} (\tilde{\mathbf{G}}_j - \mathbf{G}_j)^2, \quad \text{s.t.} \quad \mathbf{G}_j \geq \mathbf{G}_{j+1}. \tag{8}$$

which is aimed to find a sequence of positional derivatives $\mathbf{G}$ that minimizes the squared error w.r.t. the initial estimates $\tilde{\mathbf{G}}$, subject to a non-increasing constraint. This regression can be solved by the Pool Adjacent Violators Algorithm (PAVA) (De Leeuw et al., 2010). It maintains a partition of indices into disjoint contiguous pools $\mathcal{I}_1, \ldots, \mathcal{I}_K$, each assigned an initial mean value. Whenever a monotonicity violation is encountered for two neighboring pools (e.g., $\mathbf{G}_j < \mathbf{G}_{j+1}$), PAVA merges them and replaces their values with the average over the merged pool. This merge-and-average procedure repeats until all violations are eliminated, with each entry computed as

$$\hat{\mathbf{G}}_j = \frac{1}{|\mathcal{I}_k|} \sum_{t\in\mathcal{I}_k} \tilde{\mathbf{G}}_t, \quad \forall j \in \mathcal{I}_k. \tag{9}$$

Overall, PAVA has a linear time complexity $O(N)$ and is quite efficient. It ensures that the resulting derivatives decay gradually with relative distance, while simultaneously taking into account the difference in content relevance. Empirically, isotonic smoothing proves to be as important as relevance-guided resource allocation in improving the generalization capability of positional encoding (see ablation studies in Table 5).

3.3.2 Normalization of $\hat{\mathbf{G}}_j$'s under the Budget Constraints

The smoothed positional derivatives $\hat{\mathbf{G}}_j$ should be further normalized so that their total sum matches the predefined positional budget $\tilde{L}$. Prior to normalization, the nearest M chunks to the query will be first assigned the maximal positional derivative of 1, pre-consuming a fixed positional budget $\tilde{L}_0 = M \cdot S$. This reservation strategy is standard in the literature (Jin et al., 2024a; Xu et al., 2025) and is typically applied to the nearest 1024 tokens, which corresponds to $M = \lceil 1024/S \rceil$ chunks in our setting. This design reflects the common assumption that the immediate context should be preserved at the highest positional resolution*.

*Note that the M chucks nearest to the query will not be subject to the isotonic smoothing in Eq. (8), namely, the derivatives $\mathbf{g}_i$'s should start to decay for $i \geq \tilde{L}_0$.

The remaining positional budget, $\tilde{L} - \tilde{L}_0$, is allocated to the distant-context chunks C_j with $j > M$. This is accomplished via a global scaling operation that enforces the sum of positional derivatives over these chunks to match the remaining budget:

$$\bar{\mathbf{G}}_j \leftarrow \begin{cases} 1, & \text{if } j \leq M \\ \frac{\hat{\mathbf{G}}_j}{\lambda}, & \text{if } j > M, \end{cases} \quad \left(\lambda = \frac{S \cdot \sum_{j=M+1}^{N} \hat{\mathbf{G}}_j}{\tilde{L} - \tilde{L}_0}\right).$$

The normalization term λ respects the priority of immediate-context while enforcing the global budget under semantic relevance-based modulations. Furthermore, since $\tilde{L}$ is much smaller than L, this operation naturally guarantees that derivatives in the distant chunks satisfy $\mathbf{g}_i \in [0, 1]$.

3.4 From Derivative to Positions

Having determined the positional derivative (resolution) for each chunk, we need to transform these normalized values into actual relative positions $\mathcal{P}(i)$ for each historical token from the query position. To achieve this, we first recover the *chunk-level* positional derivatives $\bar{\mathbf{G}}$ into *token-level* derivatives $\mathbf{g}$ by assigning the same derivatives to all tokens within the same chunk:

$$\mathbf{g}_i = \bar{\mathbf{G}}_j, \quad j = \lceil i/S \rceil. \tag{10}$$

Based on the definition of the positional derivative in Eq. (2), the relative position $\mathcal{P}(i)$ can be computed recursively as

$$\mathcal{P}(i) = \begin{cases} 0, & \text{if } i = 0, \\ \mathcal{P}(i-1) + \mathbf{g}_i, & \text{if } i \geq 1. \end{cases} \tag{11}$$

This process is continuously updated throughout the auto-regressive generation, providing dynamic guidance for positional encoding.

3.5 Efficient Inference Implementation

Due to the high semantic similarity and functional redundancy among adjacent Transformer layers, we compute relevance maps only for a few selected **anchor layers** $\mathcal{L}_{\text{anchor}}$ to reduce the cumulative overhead of per-layer relevance estimation. Specifically, if $l \in \mathcal{L}_{\text{anchor}}$, we run the full relevance computation procedure (from Eq. (4) to Eq. (6)) to obtain $\mathcal{R}^{(l)}$. Otherwise, we reuse the map from the nearest preceding anchor layer: $\mathcal{R}^{(l)} \leftarrow \mathcal{R}^{(l')}$, where $l' = \max\{k \in \mathcal{L}_{\text{anchor}} \mid k < l\}$. This amortizes relevance computation across layers and keeps the added latency negligible. Empirically,

Methods	Single document QA			Multi document QA			Summarization			Few-shot Learning			Synthetic		Code		Avg.
	NarrativeQA	Qasper	MultiField-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PassageCount	PassageRe	Lcc	RepoBench-P	
Llama-2-7B-chat-4k																	
Original	18.70	19.20	36.80	25.40	32.80	9.40	27.30	20.80	25.80	61.50	77.80	40.70	2.10	**9.80**	52.40	43.80	31.52
SelfExtend-16k	21.69	25.02	35.21	34.34	30.24	14.13	27.32	21.35	25.78	69.50	81.99	40.96	5.66	5.83	**60.60**	**54.33**	34.62
ChunkLlama-16k	8.48	13.97	20.40	13.62	16.77	5.46	25.17	12.47	24.78	67.50	74.24	40.28	2.30	3.25	56.39	50.36	27.22
AdaGroPE-16k	18.90	**27.19**	**39.30**	29.07	30.41	12.02	**29.26**	**21.92**	26.08	**70.00**	**83.90**	41.49	5.35	3.25	56.37	51.38	34.12
GALI-16k	6.29	16.73	22.26	12.82	13.65	6.31	23.58	15.96	23.37	62.00	72.80	25.12	1.83	2.83	58.71	48.51	25.80
(Ours) RiPRA-16k	**21.96**	23.99	36.73	**40.35**	**33.74**	**16.00**	28.02	**21.92**	**26.63**	69.50	83.30	**41.83**	**6.06**	6.25	58.96	52.03	**35.45**
Llama-3-8B-ins-8k																	
Original	21.71	42.24	44.54	46.82	36.42	21.49	30.03	22.67	27.79	74.50	90.23	42.53	0.00	67.00	57.00	51.22	42.39
SelfExtend-16k	21.50	43.96	50.26	48.18	28.18	25.58	**34.88**	23.83	26.96	75.50	88.26	42.01	1.42	88.00	36.58	37.73	42.22
ChunkLlama-16k	23.87	43.86	46.97	49.37	35.34	26.52	31.06	21.99	24.45	76.00	90.73	42.29	**7.00**	72.00	**59.93**	**56.98**	44.27
AdaGroPE-16k	23.31	44.15	**52.40**	51.59	36.93	29.93	32.70	23.50	**28.02**	76.00	90.59	42.56	4.95	93.91	49.84	46.14	45.40
GALI-16k	25.88	**45.65**	47.09	51.07	37.42	28.75	30.09	22.70	24.58	**77.00**	90.91	42.43	6.00	83.00	57.04	53.06	45.17
(Ours) RiPRA-16k	**26.11**	43.54	49.62	**52.45**	**40.88**	**32.05**	32.87	**24.80**	27.77	75.00	**91.34**	**43.41**	4.62	**99.00**	54.22	46.49	**46.51**
Llama-3-8B-ins-8k																	
Original	21.71	44.24	44.54	46.82	36.42	21.49	30.03	22.67	27.79	74.50	90.23	42.53	0.00	67.00	57.00	51.22	42.39
SelfExtend-32k	26.27	44.23	50.19	48.28	38.29	29.19	29.24	22.68	24.59	76.00	90.16	42.45	**8.00**	88.00	57.47	49.51	45.28
ChunkLlama-32k	24.48	42.37	47.05	48.79	34.53	26.94	32.08	23.40	24.36	76.00	90.46	42.08	6.50	72.00	**59.52**	**60.54**	44.44
AdaGroPE-32k	23.88	44.47	52.40	**51.59**	36.93	29.93	**32.73**	23.68	**28.02**	76.00	90.15	42.58	7.03	**99.00**	50.01	46.72	45.95
GALI-32k	28.63	**45.66**	47.23	51.07	38.35	29.00	29.98	22.79	24.59	**77.00**	91.13	42.38	5.50	83.00	57.07	52.63	45.38
(Ours) RiPRA-32k	**28.74**	44.26	**52.81**	50.59	**40.17**	**32.07**	32.52	**24.86**	27.99	75.00	**91.68**	**43.32**	5.62	98.00	59.04	51.96	**47.41**
Mistralv0.1-7B-ins-8k																	
Original	19.40	34.53	37.06	42.29	32.49	14.87	27.38	22.75	26.82	65.00	87.77	42.34	1.41	28.50	57.28	**53.44**	37.08
SelfExtend-16k	23.56	39.33	49.50	45.28	34.92	23.14	30.71	**24.87**	26.83	69.50	86.47	**44.28**	1.18	29.50	55.32	**53.44**	39.86
ChunkMistral-16k	20.86	36.56	42.40	35.89	31.25	12.47	28.08	22.87	**27.09**	69.50	86.52	42.94	2.14	21.50	54.92	52.70	36.73
AdaGroPE-16k	25.02	39.00	**53.38**	47.88	35.26	25.47	31.26	23.84	26.67	**70.50**	86.66	43.86	3.41	**33.50**	55.05	51.50	40.77
(Ours) RiPRA-16k	**26.11**	**41.68**	51.03	**49.80**	**36.93**	**27.32**	**31.42**	22.27	26.40	69.50	**89.65**	43.39	**3.44**	32.00	**57.89**	52.72	**41.35**
Phi2-2k																	
Original	4.46	7.01	19.98	9.43	8.55	4.62	25.64	14.32	24.03	50.50	74.55	**1.71**	2.83	4.17	**58.96**	54.14	22.81
SelfExtend-8k	12.04	**12.10**	20.15	8.22	9.68	3.89	27.90	14.58	22.13	**61.00**	82.82	1.40	2.37	2.83	57.87	56.42	24.71
AdaGroPE-8k	14.14	11.90	26.80	9.96	**11.37**	5.09	29.68	20.04	25.19	60.00	82.69	1.29	2.37	**4.73**	58.10	55.07	26.15
(Ours) RiPRA-8k	**15.33**	11.45	**28.32**	**10.59**	9.99	**5.95**	**31.06**	**20.26**	**26.97**	57.00	**83.30**	1.50	**2.85**	4.04	**58.96**	**57.65**	**26.50**

Table 1: Performance comparison of competing methods on LongBench. The number following each method denotes the target context window size. "Original" refers to the backbone model without any extension.

for a 32-layer architecture, selecting a few anchor layers, such as $\mathcal{L}_{anchor} = \{0, 16\}$, already yields promising results with minute overhead compared to standard attention (see Table 7).

4 Experiments

4.1 Backbone Models and Baseline Methods

We build our method on four representative LLMs: Llama-2-7B-chat (Touvron et al., 2023), Llama-3-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.1 (Jiang et al., 2023), and Phi-2 (Javaheripi et al., 2023). Details of the implementation are provided in Appendix A.

We include the following training-free RoPE extrapolation methods: positional reuse methods (Self-Extend (Jin et al., 2024a), ChunkLlama (An et al., 2024b), and AdaGroPE (Xu et al., 2025)) and interpolation methods (NTK-Aware (bloc97, 2023), Dyn-NTK (emozilla, 2023), YaRN (Peng et al., 2023), and GALI (Li et al., 2025)). All results were obtained by running the officially released codes.

Datasets. Following Li et al. (2025); Xu et al. (2025), we adopt three categories of long-context benchmarks that have been widely adopted in the literature. (1) **Real-world tasks:** we use LongBench (Bai et al., 2024), covering 16 English datasets spanning QA, summarization, and code completion. To further probe long-range reasoning, we also report results on 4 challenging closed-ended L-Eval datasets (An et al., 2024a): TOFEL, QuALITY, Coursera, and SFiction. For these tasks, we strictly adhere to the official prompt templates and truncation protocols. (2) **Synthetic tasks:** following Li et al. (2025), we adopt standard Passkey Retrieval and the harder Multi-Passkey Retrieval benchmark. (3) **Long-context language modeling task:** we evaluat the language modeling performance of LLMs by examining the perplexity on their generated texts on the PG19 dataset (Rae et al., 2019).

4.2 Main Results

4.2.1 Real-World Long-Context Task Results

Table 1 summarizes the results on LongBench. RiPRA consistently outperforms the baselines

across different LLM backbones. On Llama-3-8B-Instruct, it achieves notable improvements in average performance at both 16k (+1.34%) and 32k (+2.03%) context lengths, demonstrating strong robustness under varying extension scales. RiPRA also performs strongly on representative tasks. It consistently ranks first on NarrativeQA and MuSiQue across all evaluated models and context lengths, indicating a strong ability to preserve long-range dependencies and integrate dispersed evidence for narrative comprehension and multi-hop question answering. It also ranks first or second on GovReport and TriviaQA across all settings, suggesting an advantage in maintaining coherent global context representations over long inputs. Its gains on synthetic and code-related tasks, including PassageRetrieval-en, further suggest that adaptive positional resolution helps separate key information from substantial irrelevant content and positional noise. On the smaller Phi-2 (2.7B) model, RiPRA achieves the highest average score and attains the best performance on 11 out of 16 datasets, highlighting its effectiveness even under limited model capacity.

To assess the reasoning performance over closed-ended tasks, we report results on L-Eval in Table 2 across different models. On Llama-3-8B-Instruct, RiPRA surpasses AdaGroPE by 5.3% and GALI by 6.79%. The advantage is most prominent on the longest dataset, SFiction (+6.25%), where RiPRA's "Zoom In" mechanism effectively preserves key narrative details often blurred by static positional mapping strategies. Its superior performance on mid-length tasks (QuALITY, Coursera) further confirms the versatility across varying context lengths. Note that TOFEL is excluded for Llama-3-8B-Instruct as it can fit within the native 8k window, sparing the need for context extrapolation.

	Method	TOFEL (3k∼5k)	QuALITY (4k∼9k)	Coursera (5k∼17k)	SFiction (6k∼27k)	Avg.
Llama-2-7B-chat-4k	Original	51.67	37.62	29.21	60.15	44.66
	Self-Extend	55.39	41.09	35.76	57.81	47.51
	ChunkLlama	57.62	35.14	32.12	61.72	46.65
	AdaGroPE	**61.34**	38.12	35.47	64.06	49.28
	NTK	52.78	33.16	32.71	41.41	40.02
	Dyn-NTK	52.27	30.69	13.95	57.02	38.48
	YaRN	57.62	42.08	**36.49**	42.97	44.79
	GALI	54.65	39.11	35.32	51.43	45.18
	RiPRA (Ours)	58.36	**45.05**	36.05	**67.19**	**51.66**
Llama-3-8B-ins-8k	Original	-	61.88	53.05	60.16	58.36
	Self-Extend	-	63.37	53.92	65.63	60.97
	ChunkLlama	-	60.89	54.36	64.06	59.77
	AdaGroPE	-	64.80	52.33	70.31	62.48
	NTK	-	65.35	52.03	42.97	53.45
	Dyn-NTK	-	61.88	52.03	52.34	55.42
	YaRN	-	63.37	55.96	62.50	60.61
	GALI	-	62.38	54.17	66.41	60.99
	RiPRA (Ours)	-	**69.80**	**56.98**	**76.56**	**67.78**

Table 2: Performance comparison of all competing methods on L-Eval with context lengths up to 27k tokens using Llama-2-7B-chat and Llama-3-8B-ins backbones. Baselines are grouped by their characters: positional reuse (middle three) and interpolation (last four).

4.2.2 Long Language Modeling Task Results

Table 3 reports the perplexity (PPL) of different methods on PG19 with context lengths ranging from 1k to 32k. Overall, RiPRA shows strong and stable performance across the entire context range. For short context lengths (1k–8k), its perplexity is close to the best-performing method (with 0.1%–0.3% relative difference). For longer contexts (12k–32k), RiPRA consistently achieves the lowest perplexity (0.2%–2.0% lower than the best baseline). Moreover, its perplexity decreases steadily as the context window expands. These results indicate that dynamically allocating positional resolution based on semantic relevance enables preserving and exploiting critical historical information, leading to accurate next-token prediction.

4.2.3 Synthetic Long-Context Task Results

Table 4 reports the results on the Passkey Retrieval benchmark (Mohtashami and Jaggi, 2023). All competing methods achieve 100% accuracy on the 16k and 32k settings, where the task requires retrieving a single passkey. Since this setting is already saturated, we further consider the more challenging 64k Multi-Passkey Retrieval task introduced by Li et al. (2025). In this variant, the 64k-token input is divided into four segments, each containing a random 5-digit passkey, and a prediction is counted as correct only if all four passkeys are retrieved correctly. Under this more demanding setting, RiPRA achieves 40.00% accuracy, outperforming the strongest baselines. This result suggests that relevance-guided positional zooming helps preserve positional distinctiveness for query-relevant chunks even under extreme compression, making RiPRA particularly effective for difficult retrieval-oriented long-context tasks.

Beyond its gains on contexts longer than the pretrained window, we also observe that RiPRA improves the base model even within its native context length, as shown in Appendix B.2. This finding suggests that RiPRA may also improve attention dynamics of LLMs.

	Method	1k	4k	8k	12k	16k	20k	24k	28k	32k
Llama-3-8B-ins-8k	Self-Extend	11.52	11.54	11.32	11.18	11.07	10.97	11.01	11.04	10.91
	ChunkLlama	11.72	11.77	11.54	11.39	11.27	11.22	11.19	11.15	11.12
	AdaGroPE	11.52	11.55	11.48	11.40	11.32	11.28	11.25	11.22	11.18
	NTK	11.93	11.94	11.67	11.50	11.39	13.03	23.00	42.95	77.41
	Dyn-NTK	11.51	11.53	12.75	66.88	166.86	269.93	334.83	360.57	365.36
	YaRN	11.93	11.81	11.48	11.30	11.18	11.06	11.10	11.13	11.18
	GALI	11.52	11.54	11.35	11.25	11.17	11.09	11.14	11.18	11.05
	RiPRA (Ours)	11.52	11.54	11.36	11.16	10.99	10.87	10.85	10.82	10.81

Table 3: Perplexity (PPL) of Llama-3-8B-Instruct equipped with various training-free methods for long-language modeling on the PG19 dataset, in which token lengths range from 1k to 32k.

	Method	16k Standard	32k Standard	64k Multi-Passkey
Llama-3-8b-ins	SelfExtend	100.00	100.00	15.00
	ChunkLlama	100.00	100.00	5.00
	AdaGroPE	100.00	100.00	10.00
	NTK	100.00	100.00	0.00
	Dyn-NTK	100.00	100.00	10.00
	YaRN	100.00	100.00	0.00
	GALI	100.00	100.00	30.00
	RiPRA (Ours)	100.00	100.00	**40.00**

Table 4: Accuracy on Passkey Retrieval. For the 64k multi-passkey setting, results are reported in increments of 5 as evaluation was conducted on 20 test samples.

4.3 Ablation Studies

We conduct a series of ablation studies to better understand the behavior of RiPRA and to quantify the contribution of its key design choices. Unless otherwise specified, all ablation experiments are performed using the Llama-3-8B-Instruct backbone with a 16k target context window, and are evaluated on two representative LongBench QA datasets, NarrativeQA and Musique.

4.3.1 Impact of Key Algorithm Modules

Table 5 shows that replacing the dynamic relevance scores with a uniform scheme (*RiPRA w/o Semantic Relevance Info.*) decreases the average performance from 29.08% to 26.43%, while removing isotonic regression (*RiPRA w/o Isotonic Regression*) reduces it to 26.74%. The similar magnitude of these drops indicates that both components are critical and complementary for the final performance, with semantic relevance estimation having a slightly larger effect in this setting. Taken together, the results suggest that RiPRA benefits from combining content-aware local positional resolution allocation with global structural regularization.

Method	NarrativeQA	Musique	Avg.
Full RiPRA	**26.11**	**32.05**	**29.08**
w/o Semantic Relevance Info.	22.72	30.14	26.43
w/o Isotonic Regression	22.63	30.84	26.74

Table 5: Ablation study on key algorithm components.

4.3.2 Granularity Analysis

We further conduct a sensitivity analysis on three key hyper-parameters in RiPRA: the chunk size, the number of anchor layers, and the immediate context window size.

(1) *Chunk Size.* Larger chunks greatly reduce the number of inner-product computations, but may compromise fine-grained semantic resolution. Figure 2 (left) shows performance is fairly stable from $S = 64$ to $S = 256$ on both datasets, suggesting that the method is robust to chunk granularity in this range. Interestingly, on the Musique dataset, $S = 256$ yields a higher F1 score than a smaller chunk size of $S = 64$. We hypothesize that overly small chunks may fragment semantically coherent information, whereas a moderate chunk size better preserves contextual integrity and supports effective retrieval. Overall, these results suggest that $S = 256$ strikes the best balance between efficiency and semantic precision.

(2) *Number of Anchor Layers*. The number of anchor layers reflects a trade-off between computational efficiency and representational granularity. Figure 2 (right) shows that reducing the number of anchor layers from 32 to 2 incurs only a marginal performance drop while substantially reducing redundant layer-wise relevance computation. By contrast, using only a single anchor layer (layer 0) leads to a much larger degradation. This trend is consistent with the analysis in Appendix C, where we show that the relevance landscape of the last token across the 32 layers of Llama-3-8B-Instruct

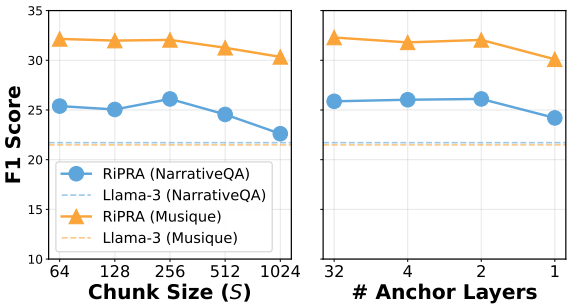

Figure 2: Performance of RiPRA under different hyperparameter settings, including the Semantic Granularity (chunk size S) and the number of anchor layers.

$\tilde{L}_0$	NarrativeQA	Musique
Base Model	21.71	21.49
256	25.87	31.98
512	26.01	**32.11**
1024	**26.11**	32.05
2048	25.85	31.80

Table 6: Performance of RiPRA under different immediate context window size choices ($\tilde{L}_0$).

naturally forms two clusters. The result suggests that neighboring Transformer layers often share similar relevance patterns, making it effective to reuse relevance information across adjacent layers. However, overly sparse anchoring, such as using only one layer, fails to capture sufficient layer-wise variation and therefore harms retrieval quality.

(3) *Immediate Context Window Size*. The immediate context window size $\tilde{L}_0$ determines the number of nearby tokens to the query that are assigned a fixed high-resolution budget, introducing a trade-off between capturing sufficient local context and the distribution of the remaining positional resources. As shown in Table 6, increasing $\tilde{L}_0$ from 256 to 1024 leads to consistent performance improvements on both datasets, while further increasing it to 2048 results in slight degradation. This pattern suggests that the immediate window is important for modeling short-range dependencies, which support the logical consistency and topical coherence of generated responses.

At the same time, $\tilde{L}_0$ directly influences the allocation of the remaining positional budget. An excessively large local window consumes too much of this budget, reducing the positional resolution available for distinguishing key tokens from irrelevant content in the rest of the prefix. This weakens the relevance signal over longer ranges and ultimately hurts performance. Taken together, the results indicate that RiPRA benefits from a balanced allocation strategy: enough local capacity to preserve immediate coherence, but enough remaining resolution to support relevance discrimination over the broader context.

Context Length	Latency (ms/token)		Memory (GB)	
	SelfExtend	RiPRA (Ours)	SelfExtend	RiPRA (Ours)
12k	172.75	178.27 (+3.2%)	25.1265	25.1272 (+0.0008)
16k	212.70	221.84 (+4.3%)	28.8335	28.8348 (+0.0012)
24k	292.85	309.84 (+5.8%)	35.2751	35.2769 (+0.0018)
32k	382.43	408.06(+6.7%)	42.0530	42.0556 (+0.0026)

Table 7: Latency and memory consumption across context lengths for SelfExtend and our method.

4.3.3 Computational and Memory Cost

We evaluate the computational efficiency of RiPRA against SelfExtend using the Llama-3-8B-Instruct backbone on an NVIDIA A6000 GPU (48GB). As shown in Table 7, RiPRA preserves strong long-context performance while introducing only modest additional cost. In particular, the overhead in per token generation latency ranges from 3.2% to 6.7% and grows approximately linearly with the input context length, which remains practical for deployment. This overhead is mainly due to chunk-based relevance computation, while the use of sparse anchoring layers helps keep it limited. In contrast, the memory overhead is almost negligible. RiPRA's memory footprint is virtually identical to that of the baseline, with the additional storage for chunk-level embeddings reaching only about 2.6 MB even at the 32k context length. These results show that RiPRA achieves its performance gains with minimal extra resource consumption.

5 Conclusion

This paper proposes RiPRA, a semantically aware, training-free long context extrapolation method. By modeling positional encoding as a constrained resource allocation, it adaptively allocates positional resolution based on the semantic relevance of the token to the query, highlighting key information and compressing irrelevant content. Experimental results show that RiPRA significantly outperforms existing training-free methods and effectively improves long context generalization ability. In the future, we shall further integrate RiPRA with other forms of positional encoding strategies and expand the scope to pretraining settings as well.

Limitations

Our empirical validation is restricted to English-language benchmarks. Although the underlying mechanism of content-aware zooming is theoretically language-agnostic, we have not rigorously verified its generalizability to multilingual settings.

This work focuses exclusively on the textual modality. With the rapid evolution of Multimodal Large Language Models (MLLMs), efficient long-context processing is increasingly vital for tasks involving high-resolution images or video streams. Whether the proposed method can be directly extended to visual labels or cross-modal sequences remains an open question.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (with Grants No. 62276099).

References

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024a. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14388–14411.

Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. 2024b. Training-free long-context scaling of large language models. *arXiv preprint arXiv:2402.17463*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.

bloc97. 2023. NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation.

Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. 2024. CLEX: continuous length extrapolation for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation. *ArXiv*, abs/2306.15595.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Jishnu Ray Chowdhury and Cornelia Caragea. 2023. Monotonic location attention for length generalization. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28792–28808. PMLR.

Jan De Leeuw, Kurt Hornik, and Patrick Mair. 2010. Isotone optimization in r: pool-adjacent-violators algorithm (pava) and active set methods. *Journal of statistical software*, 32:1–24.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2022. Position information in transformers: An overview. *Comput. Linguistics*, 48(3):733–763.

emozilla. 2023. Dynamically Scaled RoPE further increases performance of long context LLaMA with zero fine-tuning.

Chaochen Gao, Xing Wu, Zijia Lin, Debing Zhang, and Songlin Hu. 2025. Nextlong: Toward effective long-context training without long documents. *arXiv preprint arXiv:2501.12766*.

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. 2024. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719*.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. Lm-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, and 1 others. 2023. Phi-2:

The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv e-prints*, arXiv:2310.06825.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024a. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325*.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024b. LLM maybe longlm: Selfextend LLM context window without tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BOOKSUM: A collection of datasets for long-form narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6536–6558. Association for Computational Linguistics.

Rongsheng Li, Jin Xu, Zhixiong Cao, Hai-Tao Zheng, and Hong-Gee Kim. 2024. Extending context window in large language models with segmented base adjustment for rotary position embeddings. *Applied Sciences*, 14(7):3076.

Yan Li, Tianyi Zhang, Zechuan Li, and Caren Han. 2025. A training-free length extrapolation approach for llms: Greedy attention logit interpolation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8784–8804.

Xiaoran Liu, Hang Yan, Chenxin An, Xipeng Qiu, and Dahua Lin. 2024. Scaling laws of rope-based extrapolation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Amirkeivan Mohtashami and Martin Jaggi. 2023. Random-access infinite context length for transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. 2023a. Randomized positional encodings boost length generalization of transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1889–1903. Association for Computational Linguistics.

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. 2023b. Randomized positional encodings boost length generalization of transformers. *arXiv preprint arXiv:2305.16843*.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Yu Wang, Sheng Shen, Rémi Munos, Hongyuan Zhan, and Yuandong Tian. 2025. Positional encoding via token-aware phase attention. *arXiv preprint arXiv:2509.12635*.

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

Xinhao Xu, Jiaxin Li, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2025. Extending llm context window with adaptive grouped positional encoding: A training-free method. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 573–587.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Lei Shen, Zihan Wang, Andi Wang, Yang Li, and 1 others. 2023. Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5673–5684.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11765–11773. AAAI Press.

A Implementation Details

To provide a holistic view of the proposed framework, we summarize the complete execution flow of RiPRA in **Algorithm 1**.

Regarding the configuration of RiPRA, we set the chunk size to $S = 256$ and the immediate context window to $\tilde{L}_0 = 1024$ tokens. Since all evaluated backbone models (Llama-2, Llama-3, Mistral, and Phi-2) possess a 32-layer architecture, we consistently designate Layer 0 and Layer 16 as the anchor layers when realizing the sparse anchor strategy. Following GALI (Li et al., 2025), we fix the random seed to 42 and report results from a single run. All experiments are conducted on NVIDIA A6000 GPUs.

The target maximum position limit $\tilde{L}$ setting follows AdaGroPE (Xu et al., 2025), which is restricted to half of the pre-training context window size (e.g., $\tilde{L} = 2048$ for Llama-2-4k and $\tilde{L} = 4096$ for Llama-3-8k). This decision is based on an empirical observation that positional embeddings corresponding to smaller relative distances are thoroughly optimized during pre-training. By mapping extended sequences into this "well-trained" lower half of the positional spectrum, we may maximize representation quality and stability, as evidenced by the ablation results in Table 8.

Setting	Ratio	NarrativeQA	Musique	Avg.
Original	-	21.71	21.49	21.60
RiPRA ($\tilde{L} = 2048$)	$0.25 \times L_{train}$	24.52	30.15	27.34
RiPRA ($\tilde{L} = 4096$)	$0.50 \times L_{train}$	**26.11**	**32.05**	**29.08**
RiPRA ($\tilde{L} = 8192$)	$1.00 \times L_{train}$	25.20	31.45	28.33

Table 8: Impact of the maximum position limit $\tilde{L}$ using Llama-3-8B-Instruct with a pre-trained window length of $L_{train} = 8192$.

Algorithm 1 RiPRA's Inference Procedure

Input: Query $\tilde{\mathbf{q}}$, Key Sequence $\tilde{\mathbf{k}}$, Value $\mathbf{v}$, Layer index l, Chunk size S, Immediate context window $\tilde{L}_0$, Target budget $\tilde{L}$
Globals: Shared Cache $\mathcal{C}$, Anchor index l_{Anchor}
Output: Attention output $\mathbf{o}$

// Phase 1: Constructing relevance landscape
if $l \in \mathcal{L}_{anchor}$ **then**
  $\mathbf{C} \leftarrow \text{Segment}(\tilde{\mathbf{k}}, S)$
  $\mathbf{c} \leftarrow \text{MeanPool}(\mathbf{C})$
  $\mathcal{R} \leftarrow \text{Min-max Normalize}(\tilde{\mathbf{q}} \cdot \mathbf{c}^\top)$
  $l_{Anchor} \leftarrow l$ ▷ Cache anchor layer
  $\mathcal{C}[l_{Anchor}].\mathcal{R} \leftarrow \mathcal{R}$
else
  $\mathcal{R} \leftarrow \mathcal{C}[l_{Anchor}].\mathcal{R}$
end if
// Phase 2: Positional Derivative Assignment and Post-processing
$N \leftarrow \text{Length}(\mathcal{R})$, $M \leftarrow \lceil \tilde{L}_0/S \rceil$
$\tilde{\mathbf{G}} \leftarrow \mathcal{R} + \epsilon$
$\hat{\mathbf{G}} \leftarrow \text{IsotonicRegression}(\tilde{\mathbf{G}}, \text{decreasing})$
$L_{dist} \leftarrow S \cdot \sum_{j=M+1}^{N} \hat{\mathbf{G}}_j$
$\lambda \leftarrow L_{dist}/(\tilde{L} - \tilde{L}_0)$
for $j \leftarrow 1$ **to** N **do**
  if $j \leq M$ **then**
    $\mathbf{G}_j \leftarrow 1$ ▷ Neighbor preservation
  else
    $\mathbf{G}_j \leftarrow \hat{\mathbf{G}}_j/\lambda$
  end if
end for
// Phase 3: Positional Remapping & Attention
$\mathbf{g} \leftarrow \text{Repeat}(\mathbf{G}, S)$
$\mathcal{P} \leftarrow \text{CumulativeSum}(\mathbf{g})$
$\mathbf{k}_{rot} \leftarrow \text{ApplyRoPE}(\tilde{\mathbf{k}}, \text{relative\_pos} = \mathcal{P})$
return $\mathbf{o} = \text{Softmax}(\tilde{\mathbf{q}} \cdot \mathbf{k}_{rot}^\top/\sqrt{d}) \cdot \mathbf{v}$

B Additional Experimental Results

B.1 Detailed Comparison on LongBench

Due to space constraints, we defer additional comparisons with interpolation-based RoPE extrapolation methods to Table 9. This table reports the full LongBench breakdown for Llama-2-7B-Chat (4k) and Llama-3-8B-Instruct across extended context windows from 16k to 32k. Results for the baselines are taken directly from GALI (Li et al., 2025), which follows the same evaluation protocol.

Performance Across Backbone Models. The experimental results in Table 9 reveal distinct behaviors across different backbone capacities. On Llama-2-7B-Chat (16k), interpolation methods face significant challenges, with average scores largely below the original 4k baseline. This phenomenon suggests that for models with limited pre-trained contexts (4k), simple frequency scaling might aggressively alter the rotation angles, disrupting the model's original attention patterns that are not robust enough to handle such shifts. In contrast, RiPRA achieves a robust average score of 35.45%, which suggests that our method offers a more stable alternative by preserving the original positional features for critical tokens.

As the extension ratio increases to 4× on Llama-3-8B-Instruct (32k), advanced interpolation meth-

Methods	Single document QA			Multi document QA			Summarization			Few-shot Learning			Synthetic		Code		Avg.
	NarrativeQA	Qasper	MultiField-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PassageCount	PassageRe	Lcc	RepoBench-P	
Llama-2-7b-chat-4k																	
Original	18.70	19.20	**36.80**	25.40	32.80	9.40	27.30	20.80	25.80	61.50	77.80	40.70	2.10	**9.80**	52.40	43.80	31.52
NTK-16k	0.73	10.33	19.44	2.38	7.91	0.42	19.47	6.26	26.13	59.50	17.89	23.17	0.52	0.51	50.70	27.91	17.08
Dyn-NTK-16k	3.79	10.37	22.38	7.47	10.26	3.81	**29.52**	20.13	22.84	63.50	45.35	31.79	2.29	4.33	57.13	42.16	23.57
YaRN-16k	3.22	10.86	22.14	5.52	13.36	1.32	24.78	10.90	25.92	64.50	40.60	32.36	2.20	2.15	51.74	43.91	22.22
(Ours) RiPRA-16k	**21.96**	**23.99**	36.73	**40.35**	**33.74**	**16.00**	28.02	**21.92**	**26.63**	**69.50**	**83.30**	**41.83**	**6.06**	6.25	**58.96**	**52.03**	**35.45**
Llama-3-8b-ins-8k																	
Original	21.71	42.24	44.54	46.82	36.42	21.49	30.03	22.67	**27.79**	74.50	90.23	42.53	0.00	67.00	**57.00**	51.22	42.39
NTK-16k	8.04	**43.85**	47.94	20.44	34.32	1.57	24.31	13.22	24.12	74.50	52.18	33.12	4.50	45.50	46.84	38.71	32.07
Dyn-NTK-16k	8.19	43.31	47.91	34.63	35.26	7.92	26.83	17.85	24.51	**76.50**	71.72	39.15	5.67	83.50	56.58	46.39	39.12
YaRN-16k	12.39	42.60	**51.70**	40.06	35.03	12.81	30.30	22.56	23.51	75.50	82.99	42.31	**6.50**	89.00	50.51	**51.58**	41.83
(Ours) RiPRA-16k	**26.11**	43.54	49.62	**52.45**	**40.88**	**32.05**	**32.87**	**24.80**	27.77	75.00	**91.34**	**43.41**	4.62	**99.00**	54.22	46.49	**46.51**
Llama-3-8b-ins-8k																	
Original	21.71	44.24	44.54	46.82	36.42	21.49	30.03	22.67	27.79	74.50	90.23	42.53	0.00	67.00	57.00	51.22	42.39
NTK-32k	7.31	**45.11**	**53.18**	52.31	37.70	27.37	29.37	21.45	23.69	73.50	78.25	41.65	**9.00**	69.00	34.25	36.12	39.97
Dyn-NTK-32k	23.06	43.95	48.55	**52.68**	37.46	25.22	31.53	22.19	24.52	**77.00**	90.96	42.42	8.00	71.50	56.77	43.78	43.72
YaRN-32k	17.09	40.90	52.51	46.40	33.92	29.47	29.93	22.69	23.11	75.00	91.29	42.54	5.50	89.50	46.50	51.38	43.61
(Ours) RiPRA-32k	**28.74**	44.26	52.81	50.59	**40.17**	**32.07**	**32.52**	**24.86**	**27.99**	75.00	**91.68**	**43.32**	5.62	**98.00**	**59.04**	**51.96**	**47.41**

Table 9: Performance comparison between RiPRA and interpolation-based methods on LongBench

ods like YaRN show improved resilience compared to NTK. Meanwhile, RiPRA further advances the average score to 47.41%. This suggests that RiPRA can"zoom out" on irrelevant contexts to avoid the overwhelming effect of the increasing noise floor, which pure frequency interpolation lacks in processing extremely long sequences.

In Multi-document QA tasks such as HotpotQA, when evaluated on the Llama-3-8B-Instruct model with a 16k context length, different positional adaptation strategies exhibit varying levels of effectiveness. Methods based on positional interpolation may face challenges in precisely localizing fine-grained evidence within long contexts. For instance, NTK achieves an accuracy of 20.44%. One possible explanation is that the frequency scaling in RoPE is applied in an approximately isotropic manner, which may limit its ability to preserve high-frequency positional distinctions that are important for exact evidence localization.

By contrast, RiPRA explicitly allocates higher positional resolution to content that is estimated to be more relevant, enabling more accurate focus on evidence-bearing chunks. As a result, RiPRA attains an accuracy of 52.45% under the same setting. These results suggest that adapting positional resolution according to content relevance can be particularly effective for downstream tasks that demand high-precision retrieval from long contexts.

B.2 Impact of RiPRA on Native Long-Context Backbone Models

To test whether RiPRA is merely a tool for context window extension or it could also be beneficial to the general attention mechanism, we evaluate the impact of RiPRA on Mistralv0.3-7B-Instruct, an LLM that is already capable of handling pretraining context windows as long as 128k. As a result, the LongBench dataset lies entirely within this window, and so no extrapolation is needed in principle. This case study allows us to examine the impact of different context-extension methods on a backbone model that is already endowed with long context-window processing capacities.

Table 10 reports the results of both RiPRA and two SOTA methods, SelfExtend and AdaGroPE, for comparison. For SelfExtend, we retain its official implementation and hyperparameter settings. For both AdaGroPE and RiPRA, we set the target maximum relative position limit to $\tilde{L} = 8192$ (8k). We have the following two interesting observations.

(1) Pure positional reuse methods do not necessarily improve the performance of the base model *when the data fall within the pretrained context length*. As can be seen, when applying SelfExtend and AdaGroPE on Mistralv0.3-7B-Instruct, the average score drops by 0.67% and 1.85% relative to the original model, respectively. We speculate that this might be due to the static compression of token positions, which leads to a certain level of blurring when the backbone already supports full-resolution attention at this length.

(2) RiPRA slightly improves performance of the base model when the data fall within the pretrained context window. Table 10 shows that applying RiPRA to Mistralv0.3-7B-Instruct increases the average performance from 47.15% (original model) to 47.68%. In fact, among all the 16 sub-tasks, using RiPRA improves the performance over the base model for 12 cases. This suggests that RiPRA is not just a remedy for context window extension; instead, the relevance-informed positional resolution

Methods	Single document QA			Multi document QA			Summarization			Few-shot Learning			Synthetic		Code		Avg.
	NarrativeQA	Qasper	MultiField-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PassageCount	PassageRe	Lcc	RepoBench-P	
Mistralv0.3-7b-ins-128k																	
Original	**27.81**	41.21	52.99	49.76	40.01	28.58	34.64	24.69	**27.86**	76.00	88.89	45.36	5.00	91.54	59.19	**60.93**	47.15
SelfExtend	26.32	40.85	54.25	50.14	43.13	**30.86**	34.74	25.14	27.40	76.50	88.86	42.53	**5.50**	**89.00**	58.92	57.65	46.99
AdaGroPE	26.32	41.04	**54.55**	50.00	42.39	27.03	34.68	**25.26**	27.63	75.50	88.68	46.66	5.00	72.5	58.08	57.97	45.83
(Ours) RiPRA-16k	26.01	**41.50**	54.09	**51.83**	**43.77**	30.34	**34.92**	25.22	27.65	**77.00**	**88.94**	**47.09**	**5.50**	**89.00**	**59.31**	60.74	**47.68**

Table 10: Performance comparison between RiPRA and interpolation-based methods on LongBench, with a 128k pre-training context window of the Mistralv0.3-7b-instruction backbone model.

modulation seems to be beneficial to the underlying attention mechanism in general.

B.3 Impact of Different Components in Different Cases

To provide a clearer ablation study beyond two specific tasks, we extend the results in Table 5 to all six LongBench task groups (covering all 16 tasks), as reported in Table 11, which reveal where each component contributes most:

Removing either component causes a consistent drop in the average score, dropping from 46.51% to 43.89% without semantic relevance and to 44.27% without isotonic regression. More importantly, task-specific ablation results are as follows:

- Semantic relevance matters when inputs contain many distractors, e.g., for Code (50.36%→41.36%) and Synthetic (51.81%→47.67%). In these settings, accurately separating evidence from irrelevant content is crucial, and this can be achieved through content-aware relevance estimation.

- Isotonic regression matters for globally structured tasks, e.g., Summarization (28.48%→22.79%). In these settings, stable long-range position allocation is crucial, which can be achieved through global monotone smoothing that produces a more reliable positional budget distribution.

B.4 Generalization to Alternative Positional Encodings

To test the generalizability of different positional inductive biases beyond RoPE, we employ RiPRA on MPT-7B-Chat, which is pretrained with AliBi (Press et al., 2022) with a context window of 2k tokens. AliBi differs structurally from RoPE by adding a linear distance-dependent term directly to attention logits. The adaptation is straightforward: only replace the rope-based computation of the adjusted relative distance in Eq. (11) with that of AliBi. Following standard long-context evaluation protocols, we report perplexity on PG19 with context lengths ranging from 2k to 8k.

As shown in Table 12, the original AliBi-based model exhibits a sharp increment in perplexity as context length increases beyond its pretraining range (especially when exceeding 4096). In contrast, all positional extrapolation methods, including SelfExtend, AdaGroPE, and GALI, reduce the performance degradation. Among them, RiPRA maintains the lowest perplexity values across all evaluated lengths. These results provide additional evidence that RiPRA remains effective for non-RoPE positional encoding schemes.

B.5 Effectiveness on Newer Models and Longer Context

To assess whether the performance gains of RiPRA generalize across different backbone architectures, we further evaluate it on a newer RoPE-based model, Qwen3-8B, and compare it against the same baselines (SelfExtend, AdaGroPE, and GALI). All evaluations are conducted on LongBench.

As shown in Table 13, RiPRA achieves the best overall average performance and consistently outperforms all baselines across the six task categories. The improvements are particularly notable in both single-document and multi-document QA, indicating the enhanced ability to retrieve and utilize relevant context.

To further evaluate robustness under extended context lengths, we conduct experiments on the RULER benchmark (Hsieh et al., 2024) with context windows of 64k, 128k, and 256k tokens. RULER is a newer standard benchmark for long-context evaluation, comprising 13 tasks that span retrieval, multi-hop reasoning, information aggregation, and long-form question answering. We

Method	Single-Document QA	Multi-Document QA	Summarization	Few-shot Learning	Synthetic	Code	Avg.
Full RiPRA (Ours)	**39.76**	**41.79**	**28.48**	**69.92**	**51.81**	**50.36**	**46.51**
w/o Semantic Relevance Info	38.49	39.38	27.28	69.17	47.67	41.36	43.89
w/o Isotonic Regression	38.36	39.34	22.79	69.39	50.21	49.13	44.27

Table 11: Category-level ablation results on LongBench task groups.

Context length	2048	3172	4096	5120	6144	8192
MPT-7b-chat (AliBi)	9.6	12.0	26.3	52.0	115.5	196.0
+ SelfExtend	9.5	11.8	11.1	11.5	11.6	12.2
+ AdaGroPE	9.4	11.6	11.8	11.2	12.5	13.0
+ GALI	9.4	11.5	11.6	11.7	12.5	12.8
+ RiPRA (Ours)	**9.2**	**10.6**	**10.9**	**10.8**	**11.1**	**11.3**

Table 12: Perplexity comparison under the AliBi positional encoding method on PG19 for varying context lengths. Lower values represent the better performance.

report the average accuracy across all tasks to reflect overall performance. All experiments are performed using LLaMA-3.1-8B-Instruct, which natively supports context lengths up to 128k. The results are summarized in Table 14.

As the backbone model exhibits a noticeable performance degradation at longer context lengths (128k and 256k), RiPRA consistently mitigates this decline, improving average accuracy by +2.92% and +2.56%, respectively. Moreover, RiPRA outperforms all competing methods across these settings, demonstrating stronger robustness and scalability in long-context scenarios.

Combined with the results in Table 1 and 9, these findings indicate that the effectiveness of RiPRA generalizes across backbone architectures and remains robust on newer RoPE-based models, particularly under longer context regimes.

C Anchor Layer Selection

The proposed sparse anchoring strategy (in Sect. 3.5) relies on the hypothesis that the semantic relevance distribution (Relevance Landscape $\mathcal{R}$) evolves gradually across Transformer layers, allowing adjacent layers to share computation results.

To empirically validate this hypothesis, we analyze the inter-layer similarity of the Relevance Landscape. Specifically, we select a single question from the NarrativeQA dataset together with its associated context sequence. We then run Llama-3-8B-Instruct and, right before predicting the last token of the answer, compute the layer-wise Relevance Landscapes (among all previous chunks) from all 32 transformer layers. Finally, we compute the pairwise cosine similarity matrix across these Landscapes to quantify their cross-layer alignment.

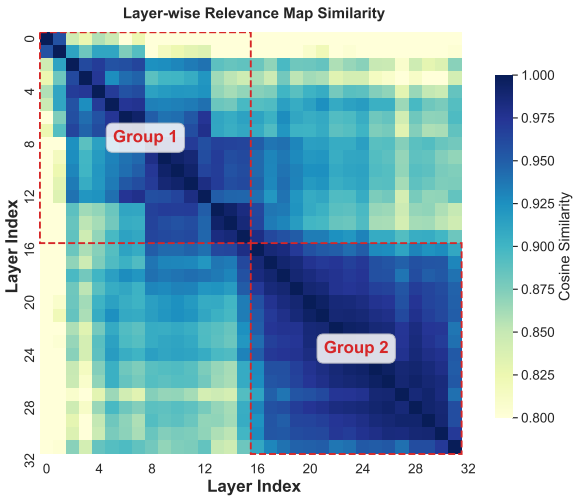

Figure 3: Layer-wise cosine similarity of Relevance Maps ($\mathcal{R}$) in Llama-3-8B-Instruct. The red dashed boxes indicate the regions covered by anchor layers (0-th and 16-th). The high similarity (dark color) within these boxes justifies the reuse of relevance maps, while the low similar region (light color) outside the boxes confirms the need for mid-layer transition.

The heatmap in Figure 3 exhibits a prominent block-diagonal structure: Similarity remains consistently high within local layer blocks (e.g., layers 0–15, enclosed by the upper red dashed boxes), suggesting that the Relevance Map from the anchor (e.g., Layer 0) provides an accurate proxy for nearby layers. In contrast, similarity between far-apart layers (e.g., Layer 0 vs. Layer 30) drops markedly, indicating substantial cross-layer semantic drift. Therefore, sharing a single static map across all layers is likely suboptimal.

Together with the results in Figure 2 and Table 7, these observations further confirm that selecting the 0-th and 16-th layers as anchors achieves an optimal balance between layer-wise semantic consistency and overall computational efficiency.

D Visualization of Relevance Landscape

To evaluate whether the Relevance Landscape R can reliably pinpoint key evidence, we conduct a qualitative study with Llama-3-8B-Instruct on HotpotQA. Here, we randomly sample 10 questions and extract the layer-16 relevance map at the de-

Method	Single-Document QA	Multi-Document QA	Summarization	Few-shot Learning	Synthetic	Code	Avg.
Qwen3-8B	42.67	41.45	27.36	71.69	51.75	61.06	49.33
+ SelfExtend	42.21	42.02	27.58	70.94	52.10	61.38	49.37
+ AdaGroPE	43.48	42.37	27.84	71.12	52.31	61.55	49.78
+ GALI	42.93	41.94	27.43	70.87	52.25	61.44	49.48
+ RiPRA (Ours)	**44.82**	**43.31**	**28.41**	**73.03**	**52.75**	**62.87**	**50.87**

Table 13: LongBench results on the Qwen3-8B backbone.

Model	64k	128k	256k
LLaMA-3.1-8B	84.71	77.04	71.23
+ SelfExtend	84.96	76.82	71.02
+ AdaGroPE	85.35	79.48	71.98
+ GALI	85.90	76.84	72.11
+ RiPRA (Ours)	**86.88**	**79.96**	**73.79**

Table 14: Performance on the RULER benchmark under extended context lengths.

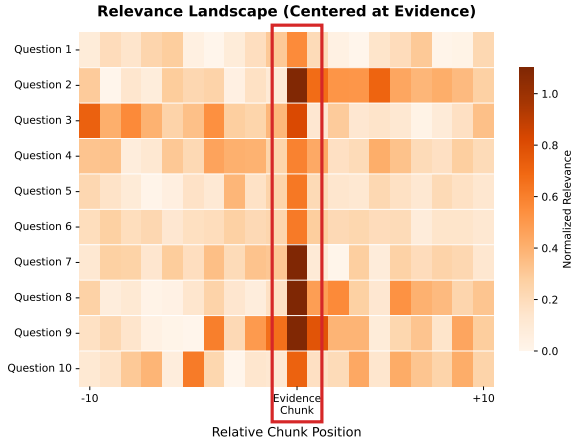

Figure 4: Visualization of the Relevance Landscape from the 16th layer in Llama-3-8B-Instruct. The heatmap displays a neighboring window of relevance scores around the ground truth evidence chunk (highlighted by the red box) across 10 randomly selected question samples from HotpotQA. Darker colors indicate higher normalized relevance.

coding step immediately before generating the final answer token. For each question, we first identify the chunk containing the ground-truth supporting evidence and treat its position as the reference index (0). We then re-index all chunks relative to this position and construct a symmetric window spanning 10 neighboring chunks on each side. This setup allows us to conveniently visualize whether the relevance score at the evidence chunk stands out compared to its surrounding chunks.

Figure 4 visualizes these aligned relevance profiles as a heatmap, with the evidence chunk highlighted by the red box. Across questions, the highest relevance mass is consistently concentrated on the evidence chunk, while scores diminish substantially in surrounding chunks. This sharp contrast between contextual signal and noise indicates that the Relevance Landscape can accurately localize the critical semantic chunks that support the answer, even in long-context settings.

E Data Stastics

We provide detailed information about each dataset in LongBench and L-Eval. Table 15 summarizes the dataset names, task types, and the number of samples for both benchmarks.

Source	Dataset	Task Type	#Samples
LongBench	NarrativeQA	Single-doc QA	200
	Qasper	Single-doc QA	200
	MultiField-en	Single-doc QA	150
	HotpotQA	Multi-doc QA	200
	2WikiMQA	Multi-doc QA	200
	Musique	Multi-doc QA	200
	GovReport	Summarization	200
	QMSum	Summarization	200
	MultiNews	Summarization	200
	TREC	Few shot	200
	TriviaQA	Few shot	200
	SAMSum	Few shot	200
	PassageCount	Synthetic	200
	PassageRe	Synthetic	200
	LCC	Code Completion	500
	RepoBench-P	Code Completion	500
L-Eval	TOFEL	Multiple Choice	269
	QuALITY	Multiple Choice	202
	Coursera	Multiple Choice	172
	SFiction	True or False	64

Table 15: Detailed statistics of the datasets used in LongBench and L-Eval.

License compliance. All scientific artifacts used in this paper comply with the corresponding licenses stated in the original papers or websites and are used solely for research purposes.