Scale-Aware Positional Reparameterization for Long-Context LLMs

1st Yihong Huang
School of Computer Science and Technology
East China Normal University
Shanghai, China
51275901062@stu.ecnu.edu.cn

2nd Hongbo Zhao
School of Computer Science and Technology
East China Normal University
Shanghai, China
51275901107@stu.ecnu.edu.cn

***Abstract*—Large language models (LLMs) suffer from rapidly deteriorating performance once the input sequence exceeds the pre-training context window. Existing methods mitigate this issue through interpolation or extrapolation. Despite being effective, these strategies either incur additional computational overhead or lack flexibility across varying context lengths. In this work, we propose Scale-Aware Reparameterization for long-context eXtrapolation (SAReX), a training-free scale-aware method to extend LLMs' extrapolation capability. SAReX formulates positional reuse as an optimization problem, minimizing a distance-based objective to adaptively determine group sizes, thereby enabling more effective utilization of pre-trained positional indices. In addition, we introduce an anchor-guided group scheme selection mechanism to adapt to inputs of varying lengths, avoiding per-length optimization and incurring no additional inference-time computational overhead. By integrating adaptive positional reuse with anchor-guided scheme selection, SAReX provides a flexible framework for long-context extrapolation that adapts seamlessly to varying lengths. Extensive experiments demonstrate that SAReX consistently achieves up to 4.35% absolute gains over existing training-free extrapolation methods across diverse base models, further validating its robustness and effectiveness.**

***Index Terms*—long context, length extrapolation, position embedding, large language models.**

I. Introduction

Large Language Models (LLMs) have gained widespread traction, demonstrating remarkable proficiency in various tasks [1], [2]. As a result, LLMs are increasingly expected to process and reason over long-context inputs, such as multi-document understanding, code repositories, and long-from question answering [3]. Accordingly, the ability to effectively extrapolate to longer context lengths has become a crucial requirement for modern LLM systems.

However, LLMs remain fundamentally constrained by their predefined context windows, as performance degrades sharply once the input length exceeds these limits [4]–[9]. A straightforward solution is to fine-tune models on long-context corpora [10]; despite its effectiveness, high-quality long-context data are scarce [11], and such fine-tuning incurs quadratic computational and memory costs [12].

Therefore, recent research [4]–[6], [13] has shifted toward more resource-efficient alternatives. Based on the observation that position embeddings are a major source of out-of-distribution (OOD) behavior [9], these methods primarily remap positional indices within the pre-training window when long-context inputs exceed it.

Accordingly, existing methods can be broadly categorized into two classes. *Interpolation approaches*, such as PI [4], NTK-aware [14], YaRN [5] and GALI [15], extend context length by rescaling positional indices into the original range, mitigating distributional shift. Therefore, they reduce reliance on long-context data and typically require only lightweight fine-tuning on shorter corpora. *Extrapolate approaches*, such as Self-Extend [6] and DCA [13], remap positional indices to reuse pre-trained embeddings, avoiding unseen positions and enabling training-free long-context extrapolation.

While effective, existing training-free long-context extrapolation methods predominantly suffer from two fundamental limitations: **abrupt positional granularity changes** and **rigid extrapolation schemes**.

First, existing methods often rely on coarse and discontinuous positional modeling, causing substantial loss of positional resolution. For example, Self-Extend [6] partitions inputs into two groups: the first increases positions normally, while the second repeatedly reuses the same indices 16 times. This binary design introduces an abrupt granularity shift, which may hinder smooth long-range dependency modeling.

Second, extrapolation strategies and hyperparameters in existing methods are typically set empirically and remain fixed at inference. Consequently, they cannot adapt to varying input lengths, resulting in suboptimal reuse of positional indices.

Motivated by these limitations, we propose **S**cale-**A**ware **R**eparameterization for long-context **eX**trapolation (SAReX), a training-free, scale-aware framework that extends the long-context capabilities of LLMs. SAReX employs a multi-group design with varying positional resolutions to enable intermediate-granularity positional reuse and smooth transitions, while adaptively selecting extrapolation schemes based on input sequence length.

Specifically, SAReX operates in two stages: offline and online. *In the offline stage*, we formulate multi-group size selection as an optimization problem under a carefully designed objective. This objective encodes the intuition that tokens near the query require finer positional resolution, while distant tokens can tolerate coarser representations, naturally balancing local precision with long-context extrapolation. Since the

grouping depends only on input length, it can be fully precomputed. *In the online stage*, SAReX selects the appropriate grouping based on input length and applies it to determine positional reuse for each group, enabling smooth and adaptive long-context extrapolation without additional optimization.

Overall, SAReX provides a training-free, plug-and-play framework for extending existing LLMs to longer contexts while naturally adapting to varying input lengths. Extensive experiments on LLaMA and Mistral show that, across challenging long-context QA, synthetic tasks, and code completion, SAReX achieves improvements of up to 4.35% over existing extrapolation methods.

Our contributions are summarized as follows.

• We propose **SAReX**, a training-free method that introduces gradual resolution control for position embeddings, avoiding abrupt changes in positional granularity and enabling smoother long-context modeling.

• We introduce a anchor-guided scheme selection mechanism that adaptively chooses grouping schemes based on input length, enabling LLMs to efficiently extrapolate to sequences of arbitrary length with minimal computational overhead.

• Extensive experiments on long-context benchmarks further validate the effectiveness of our method, with SAReX consistently outperforms existing s.o.t.a. methods.

II. Related Work

A. Position Embedding

Given the permutation-invariant nature of the Transformer architecture [12], [16], position embedding is required to inject order information into token representations. Existing approaches can be broadly categorized into absolute position embeddings and relative position embeddings. *Absolute position embedding*, including sinusoidal encodings and learnable embeddings [12], assigns each token a unique representation based on its absolute position in the sequence. *Relative position embedding*, on the other hand, encodes positional information based on the relative distance between tokens, enabling better generalization to variable-length and long-context inputs. Representative approaches include RoPE [17] and ALiBi [18], and RoPE have become the dominant choice in most existing LLMs [1], [2], [19].

B. Context Window Extension

Building on RoPE, prior studies have shown that most existing LLMs struggle to extrapolate to longer sequences, primarily due to the exposure to unseen positional indices beyond the pre-training context window. Previous works [20] have explored directly modifying positional indices to out-of-distribution indices and fine-tuning on short sequences to simulate fine-tuning on longer contexts. While effective, their performance remains limited and may degrade performance on shorter sequences. To address this limitation, recent work has mainly focused on interpolation- and extrapolation-based methods to enhance long-context generation in LLMs.

Interpolation-based methods like PI [4] extend the context window by rescaling positional indices within the pre-training range, while YaRN [5] further refines this strategy by applying non-uniform interpolation across different frequency bands to better balance short- and long-range positional information. GALI [15] performs interpolation at both the positional and attention-logit levels, thereby enabling training-free long-context extrapolation. Despite constraining positional indices to the pre-training range, most interpolation-based methods yield non-integer positions, which may require an additional lightweight fine-tuning step on short sequences.

Extrapolation-based methods typically reuse positional indices within the pre-training range. For example, Self-Extend [6] maintains two groups of positional indices, assigning original positional indices to nearby tokens and reusing the remaining indices a fixed number of times, while DCA [13] decomposes long-sequence attention into chunk-based modules to capture both intra- and inter-chunk relative positional information. By avoiding the introduction of unseen positional indices, these methods are inherently compatible with training-free, plug-and-play deployment.

III. Methodology

A. Problem Definition

Let L_{train} denote the context window length used during pre-training. At inference time, we consider an out-of-distribution input sequence of length $\tilde{L} > L_{\text{train}}$. Since position embeddings are only learned for positions within the range $[1, \ldots, L_{\text{train}}]$, directly assigning positional indices beyond this range is known to cause severe performance degradation. Consequently, the core challenge of long-context extrapolation lies in how to assign valid positional indices to an over-length input while remaining compatible with the positional distribution observed during pre-training.

Formally, our objective is to construct a remapping function

$$\phi : \{1, \ldots, \tilde{L}\} \to \{1, \ldots, L_{\text{train}}\}, \tag{1}$$

such that the relative order of positions is preserved, i.e., $\phi(i) \leq \phi(j)$ for any $i < j$, while ensuring that all mapped indices lie within the pre-training range.

To this end, we adopt a multi-group positional index reuse strategy. Specifically, we first partition the input sequence into N contiguous and non-overlapping groups. For notational convenience, we introduce the group-length vector

$$\mathbf{g} = [g_1, \ldots, g_N], \tag{2}$$

where g_n denotes the length of the n-th group. The group lengths satisfy the constraint

$$\sum_{n=1}^{N} g_n = \tilde{L}, \tag{3}$$

which guarantees that every token in the input sequence belongs to exactly one group.

Within this formulation, positional indices are reused in a group-dependent manner to enable extrapolation beyond the pre-training context window. Concretely, within the k-th

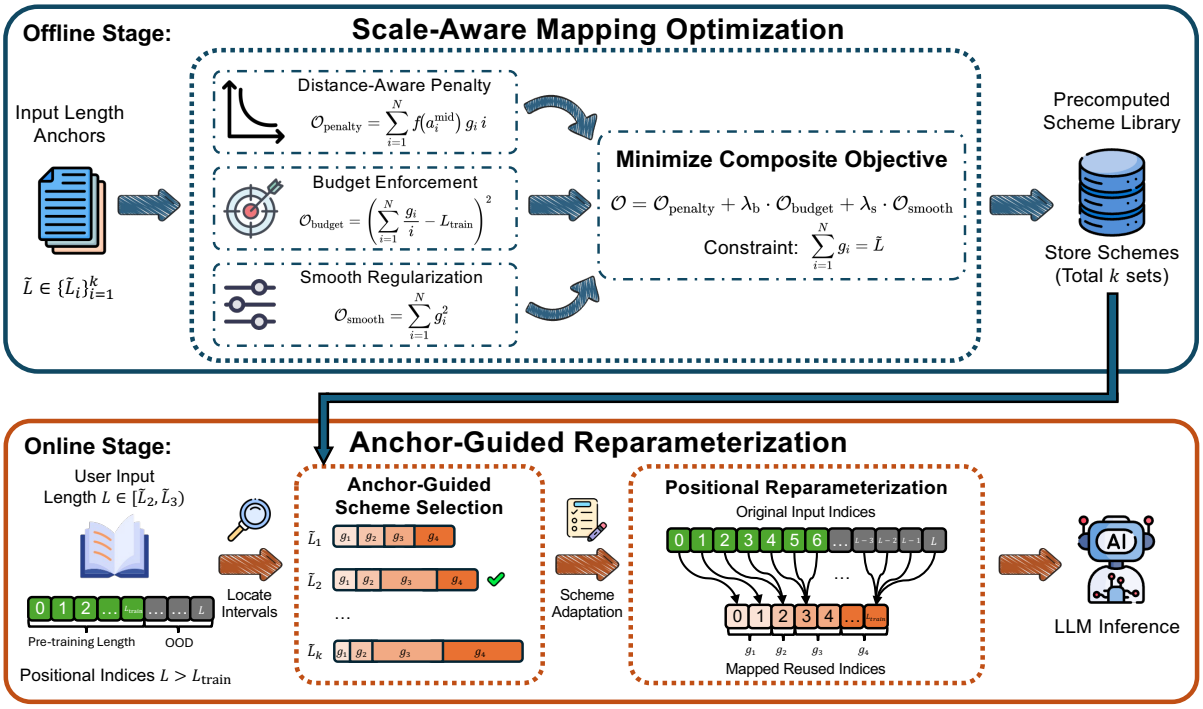

Fig. 1. Overview of the SAReX architecture.

group, each positional index is reused k times. The corresponding remapping function is defined as

$$\phi(i) = \phi(e_{k-1}) + \left\lfloor \frac{i - e_{k-1}}{k} \right\rfloor, \quad i \in [e_{k-1}, e_k], \tag{4}$$

where $e_k = \sum_{n=1}^{k} g_n$ denotes the ending index of the k-th group, and we define $e_0 = 0$.

Under this formulation, the remapping function $\phi(\cdot)$ is fully determined by the group-length vector $\mathbf{g}$. Consequently, optimizing the remapping function can be equivalently reduced to solving for an appropriate group-length configuration $\mathbf{g}$.

B. Scale-Aware Mapping Optimization

Building upon the above reparameterization, we now consider how to determine a scale-aware allocation of $\mathbf{g}$. Different choices of $\mathbf{g}$ induce different patterns of positional index reuse, which in turn directly affect the projected extension length. Therefore, selecting an appropriate $\mathbf{g}$ is crucial for robust long-context extrapolation.

We therefore formulate scale-aware mapping as an optimization problem over the group-length vector $\mathbf{g}$. The objective balances preserving fine-grained positional information and extending the usable context length, while remaining compatible with positional indices observed during pre-training. Designing such an objective is challenging, as it must satisfy multiple, potentially competing, desiderata.

First, the objective should discourage excessive position reuse near the query, where precise positions are critical for attention fidelity. Aggressive reuse in early positions can cause information loss and harm tasks dependent on local ordering.

Second, the objective should encourage full utilization of the pre-trained positional index range. Underuse reduces effective positional resolution and wastes pre-trained capacity, limiting extrapolation benefits.

Third, the objective should prevent degenerate group allocations that either concentrate reuse in a few groups or distribute it too uniformly. Such configurations can lead to unstable extrapolation behavior.

Motivated by these considerations, we construct a composite objective comprising three complementary terms:

$$\mathcal{O}(\mathbf{g}) = \mathcal{O}_{\text{penalty}} + \lambda_{\text{budget}} \cdot \mathcal{O}_{\text{budget}} + \lambda_{\text{smooth}} \cdot \mathcal{O}_{\text{smooth}}, \tag{5}$$

where each term targets a specific aspect of scale-aware position reuse. The resulting optimization problem is then formulated as:

$$\min_{\mathbf{g}} \ \mathcal{O}(\mathbf{g}) \quad \text{s.t.} \quad \sum_{n=1}^{N} g_n = \tilde{L}. \tag{6}$$

We next detail the design of each objective term and explain how it contributes to scale-aware position reuse.

1) Distance-Aware Reuse Penalty: We introduce a distance-aware reuse penalty, $\mathcal{O}_{\text{penalty}}$, to regulate positional index reuse across the sequence. This design is motivated by the observation that positional precision is more critical for nearby tokens, while reuse is less harmful for distant ones and enables longer extrapolation.

Accordingly, $\mathcal{O}_{\text{penalty}}$ accounts for both token distance and reuse multiplicity, where a distance-dependent weighting function $f(\cdot)$ models the importance of positional precision and assigns higher penalties to aggressive reuse at shorter distances.

TABLE I

PERFORMANCE COMPARISON OF DIFFERENT LLMS ON LONGBENCH. BEST AND SECOND-BEST RESULTS IN EACH GROUP ARE HIGHLIGHTED WITH BOLD AND UNDERLINE, RESPECTIVELY.

Model	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	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
+Self-Extend-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	20.37	24.60	34.78	26.44	28.42	9.87	27.03	20.85	26.17	68.50	73.84	39.88	3.90	3.50	56.69	53.83	32.42
+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
+SAReX (Ours)	**21.80**	24.95	**39.38**	31.61	29.12	**14.23**	**29.64**	**21.92**	**26.28**	**69.50**	**84.50**	**42.20**	5.50	6.50	58.86	53.81	**34.98**
Llama-3-8b-Instruct	21.71	44.24	44.54	46.82	36.42	21.49	30.03	22.67	**27.79**	74.50	90.23	**45.23**	0.00	67.00	57.00	51.22	42.39
+Self-Extend-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
+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
+SAReX (Ours)	23.17	44.17	49.58	51.06	**39.75**	**31.44**	**34.66**	**23.56**	27.31	75.50	**91.46**	43.67	**8.53**	**98.50**	57.21	52.88	**47.03**
Mistral-7b-ins-0.1 w/o SWA	20.46	35.36	39.30	34.81	29.91	11.21	24.70	21.67	26.67	68.00	86.66	41.28	0.18	24.00	56.94	**55.85**	36.06
+Self-Extend-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
+GALI-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
+SAReX (Ours)	**25.52**	**41.26**	50.80	**48.15**	**36.12**	**26.77**	31.05	23.42	26.40	69.50	**88.25**	43.50	**3.50**	33.00	**57.38**	52.70	**41.08**

Ideally, penalties would be computed at the token level. However, under group-wise reparameterization, tokens in the same group share identical reuse multiplicity and exhibit limited weighting variation, making group-level aggregation sufficient while avoiding unnecessary computational overhead.

To balance fidelity and efficiency, we approximate the token-level formulation by computing penalties at the group level using group midpoints. Formally, for the i-th group, its midpoint position is defined as

$$a_i^{\text{mid}} = \sum_{j=1}^{i-1} g_j + \frac{1}{2} g_i. \tag{7}$$

This midpoint serves as a proxy for the average distance of tokens within the group. Using this approximation, the distance-aware reuse penalty is formulated as

$$\mathcal{O}_{\text{penalty}} = \sum_{i=1}^{N} f\big(a_i^{\text{mid}}\big)\; g_i\, i, \tag{8}$$

where g_i is the group length, i indexes the reuse multiplicity of the i-th group, and $f(\cdot)$ is a monotonically decreasing function modulating the penalty by distance. Here, we adopt an exponential decay function:

$$f(x) = e^{-\frac{c \cdot x}{\tilde{L}}}, \tag{9}$$

where c controls decay relative to input length $\tilde{L}$, assigning stronger penalties to nearby reuse and gradually relaxing for distant tokens.

2) Position Budget Enforcement: While $\mathcal{O}_{\text{penalty}}$ shapes positional resolution across scales, it alone does not ensure full use of the pre-trained positional index range. Unconstrained optimization over $\mathbf{g}$ may either concentrate reuse in a narrow index range, underutilizing many pre-trained positions, or be overly conservative, causing remapped positions to exceed the pre-trained maximum. Both hinder robust long-context extrapolation, motivating explicit position budget enforcement.

To address this, we introduce a constraint that encourages the mapping to utilize the full range of available indices. Under group-wise reparameterization, the effective number of distinct positional indices consumed by the i-th group with reuse multiplicity i is $\frac{g_i}{i}$, and summing over all N groups yields the total indices used. Based on this, we define the position budget utilization objective as

$$\mathcal{O}_{\text{budget}} = \left(\sum_{i=1}^{N} \frac{g_i}{i} - L_{\text{train}} \right)^2, \tag{10}$$

which penalizes deviations from the full pre-trained index range. Minimizing $\mathcal{O}_{\text{budget}}$ encourages the mapping to fully exploit the available positional embeddings, preventing both under-utilization and excessive sparsity in index assignments.

3) Group Smoothness Regularization: In addition to positional penalties and budget constraints, scale-aware mapping should vary smoothly across groups. Without regularization, optimizing $\mathbf{g}$ may produce degenerate solutions with abrupt fluctuations between adjacent groups, leading to uneven positional resolution.

To address this, we introduce a smoothness regularization term over group lengths:

$$\mathcal{O}_{\text{smooth}} = \sum_{i=1}^{N} g_i^2, \tag{11}$$

which penalizes extreme values in $\mathbf{g}$ via a quadratic term.

Above all, Under the constraint in Eq.3, the objective in Eq.5 is minimized to obtain the group-length vector $\mathbf{g}$ for optimal allocation of N groups. Each element of $\mathbf{g}$ is discretized via rounding to ensure a valid grouping scheme. By updating the input length $\tilde{L}$ in both the constraint and objective, SAReX generates scale-aware grouping schemes that can be computed offline without inference-time overhead.

C. Anchor-Guided Reparameterization

Although the optimal group allocation $\mathbf{g}$ can be precomputed for every input length $\tilde{L}$, this approach has limitations.

TABLE II
PERFORMANCE COMPARISON OF VARIOUS LLMS ON THE RULER BENCHMARK WITH A 16K CONTEXT LENGTH. AVERAGE SCORES ARE CALCULATED ACROSS ALL 13 SUB-TASKS.

Models	NIAH	VT	Aggregation	QA	Avg. (13 tasks)
Llama-3-8b-Instruct	88.94	**78.72**	63.51	55.70	79.13
+Self-Extend	93.38	72.56	57.29	**59.50**	81.01
+ChunkLlama	92.21	71.74	65.99	57.90	81.32
+GALI	94.08	74.78	69.19	57.90	83.20
+SAReX (Ours)	**94.09**	74.52	**71.50**	58.20	**83.59**

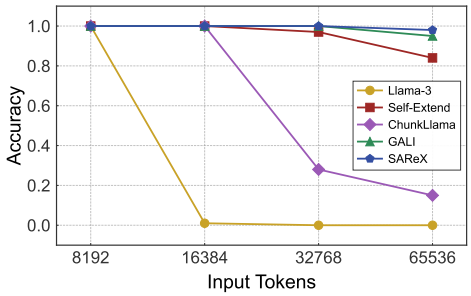

Fig. 2. Passkey retrieval accuracy of LLaMA-3-8B-Instruct using SAReX and baseline methods under varying input tokens.

Precomputing and storing grouping schemes for all lengths incurs significant computational and storage overhead, while the grouping schemes are often highly similar across ranges of $\tilde{L}$, leading to redundant computations.

Therefore, we propose an anchor-guided reparameterization strategy based on input length anchors to balance optimality and efficiency. Specifically, we discretize the input length space with a fixed step size Δ_L and precompute optimal group allocations for anchor lengths $\{L_k\}_{k=1}^{K}$:

$$L_k = L_{\text{train}} + (k-1)\Delta_L, \quad k = 1, \ldots, K, \tag{12}$$

where L_{train} is the pre-training context window and K is the number of anchors. For each anchor, we optimize $\mathbf{g}$ offline using Eq. 5 and store the resulting allocations for efficient inference lookup.

At inference time, an input of arbitrary length is mapped to the nearest anchor in this predefined set. Specifically, for an input sequence of length $\tilde{L}$, if $\tilde{L} \in [L_k, L_{k+1})$, we simply adopt the grouping scheme corresponding to L_k. In this way, SAReX provides a simple and efficient mechanism for anchor-guided adaptive grouping across sequences of varying lengths.

IV. EXPERIMENTS

Datasets. Following prior work [6], [15], we evaluate on both real-world and synthetic long-context benchmarks. For real-world evaluation, we use LongBench [7] and RULER [8]. LongBench includes 16 English tasks across QA, summarization, code completion, and reasoning, with diverse context lengths. RULER contains 13 tasks targeting long-context reasoning, such as Needle-in-a-Haystack, variable tracking, and aggregation. In our experiments, all models are evaluated on RULER with a fixed length of 16k tokens.

For synthetic long-context evaluation, we use the passkey retrieval task, which tests a model's ability to locate a target token at arbitrary positions within long sequences. We evaluate inputs up to 64k tokens to systematically assess model behavior under extended contexts.

Models and Baselines. We evaluate SAReX against representative baselines on three base models: LLaMA 2 [1], LLaMA 3 [2], and Mistral-7B-Instruct-v0.1 [19]. The pre-training context window of LLaMA 2 is 4k tokens, while the other two models support up to 8k tokens. For baselines, we compare SAReX with several training-free extrapolation methods, including Self-Extend [6], DCA [13], and GALI [15]. For all open-source methods, results are obtained using their official implementations. All experiments are conducted on a workstation equipped with a single NVIDIA A6000 GPU.

A. Evaluation Results

Real-World Long Context Tasks. Table I reports results on LongBench. Overall, SAReX delivers consistently strong performance across different LLMs, demonstrating its effectiveness for long-context extrapolation. On LLaMA-3-8B-Instruct, SAReX surpasses all baselines, achieving up to a 2.59% absolute gain over existing training-free methods and 4.64% over the original model. Similar trends hold for smaller models: with Mistral-7B, SAReX ranks first or second on 12 of 16 tasks, yielding average gains of 2.59%–4.35%, further confirming its effectiveness.

To further evaluate performance on closed-ended long-context tasks, we report results on RULER in Table II using LLaMA 3. Overall, SAReX demonstrates strong performance on synthetic long-context tasks, achieving up to a 2.58% absolute improvement over competing baselines. In challenging settings requiring comprehensive input understanding, such as Needle-in-a-Haystack and aggregation, SAReX ranks first among all compared methods.

Thanks to its anchor-guided reparameterization, SAReX adapts automatically to varying benchmarks and context lengths without manual extrapolation. This enables robust performance on datasets with diverse input lengths, like LongBench. By contrast, RULER uses nearly fixed input lengths, limiting the benefits of anchor-guided adaptation. Even in this setting, SAReX outperforms GALI by 0.39%, showing consistent robustness.

Synthetic Long Context Tasks. Figure 2 illustrates the passkey retrieval accuracy of SAReX and other baseline methods on LLaMA 3. In this experiment, the passkey consists of 36 digits and is randomly positioned within the input context. As shown, for the base model, once the input length exceeds the original context window, the passkey retrieval accuracy drops sharply. A similar trend is observed for ChunkLlama, which partitions the input into multiple chunks and may disrupt long-range token dependencies. In contrast to these methods, positional reuse approaches exhibit more robust performance; among them, SAReX consistently achieves the best results, validating its effectiveness on both synthetic and real-world long-context tasks.

TABLE III
ABLATION STUDIES OF SAREX.

Method	HotpotQA	MuSiQue	Avg
SAReX	**51.06**	**31.44**	**41.25**
w/o position budget	49.67	29.81	39.74
w/o group smoothness	47.24	26.55	36.90
w/o anchor-guidance	50.16	30.44	40.30

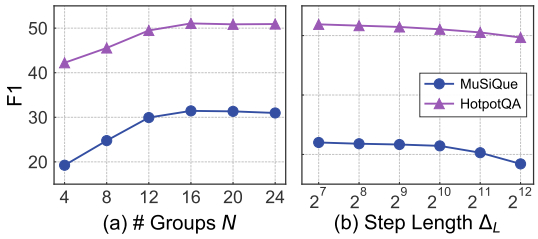

Fig. 3. Ablation study of SAReX with respect to (a) the number of groups N, and (b) the fixed step length Δ_L.

B. Ablation Studies

We further conduct ablation studies to validate the effectiveness of SAReX under different hyperparameter settings. All experiments are performed on LLaMA 3 and evaluated on two LongBench subsets, HotpotQA and Musique.

Table III analyzes key design choices in SAReX. (1) Removing position budget enforcement (Eq.10) causes an average drop of 1.51%, indicating the importance of fully utilizing pre-trained positional indices. (2) Removing group smoothness regularization (Eq.11) leads to a larger drop of 4.35%, suggesting that extreme grouping harms positional consistency. (3) Removing anchor-guided reparameterization and directly extrapolating to 16k results in a 0.95% drop, demonstrating the robustness of SAReX across input lengths.

Figure 3(a) shows the F1 score versus the number of groups N for position reuse. The F1 score increases with N and peaks at $N = 16$, after which it declines, indicating that overly large N reduces positional resolution and harms performance.

Figure 3(b) presents the F1 score versus the step length Δ_L in anchor-guided reparameterization. Smaller Δ_L improves performance, but gains become marginal when $\Delta_L < 1024$ while increasing computational cost, suggesting a trade-off between accuracy and efficiency.

Based on these observations, we fix $N = 16$ and $\Delta_L = 1024$ in all experiments for evaluation.

V. CONCLUSION

This paper proposes SAReX, a training-free method for extending existing large language models to long-context inputs. Given the input length, SAReX adaptively defines an objective to minimize positional ambiguity and derives a corresponding group allocation to reparameterize positional embeddings. Extensive experiments show that SAReX consistently outperforms existing training-free approaches across different LLMs, demonstrating its robustness and effectiveness.

REFERENCES

[1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[2] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[3] G. Lee, V. Hartmann, J. Park, D. Papailiopoulos, and K. Lee, "Prompted llms as chatbot modules for long open-domain conversation," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 4536–4554.

[4] S. Chen, S. Wong, L. Chen, and Y. Tian, "Extending context window of large language models via positional interpolation," *arXiv preprint arXiv:2306.15595*, 2023.

[5] B. Peng, J. Quesnelle, H. Fan, and E. Shippole, "YaRN: Efficient context window extension of large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[6] H. Jin, X. Han, J. Yang, Z. Jiang, Z. Liu, C.-Y. Chang, H. Chen, and X. Hu, "Llm maybe longlm: Selfextend llm context window without tuning," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 22 099–22 114.

[7] Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou *et al.*, "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)*, 2024, pp. 3119–3137.

[8] C.-P. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, and B. Ginsburg, "RULER: What's the real context size of your long-context language models?" in *First Conference on Language Modeling*, 2024.

[9] X. Liu, H. Yan, C. An, X. Qiu, and D. Lin, "Scaling laws of roPE-based extrapolation," in *The Twelfth International Conference on Learning Representations*, 2024.

[10] Y. Chen, S. Qian, H. Tang, X. Lai, Z. Liu, S. Han, and J. Jia, "LongloRA: Efficient fine-tuning of long-context large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[11] Y. Fu, R. Panda, X. Niu, X. Yue, H. Hajishirzi, Y. Kim, and H. Peng, "Data engineering for scaling language models to 128k context," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 14 125–14 134.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[13] C. An, F. Huang, J. Zhang, S. Gong, X. Qiu, C. Zhou, and L. Kong, "Training-free long-context scaling of large language models," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 1493–1510.

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

[15] Y. Li, T. Zhang, Z. Li, and C. Han, "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*, 2025, pp. 8784–8804.

[16] H. Xu, L. Xiang, H. Ye, D. Yao, P. Chu, and B. Li, "Permutation equivariance of transformers and its applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5987–5996.

[17] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[18] O. Press, N. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," in *International Conference on Learning Representations*, 2022.

[19] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023.

[20] D. Zhu, N. Yang, L. Wang, Y. Song, W. Wu, F. Wei, and S. Li, "PoSE: Efficient context window extension of LLMs via positional skip-wise training," in *The Twelfth International Conference on Learning Representations*, 2024.