SCOPE: Boosting LLM Efficiency with Scoped Position Encoding

Qingguo Qi[1,2], Hongyang Chen * [2], Zhao Li[2]

[1]Zhejiang University, [2]Zhejiang Lab

qiqingguo@zju.edu.cn, hongyang@zhejianglab.org, lzjoey@gmail.com

Abstract

Positional encodings are fundamental to Transformers, yet explicit methods like RoPE can degrade under length extrapolation and may incur extra arithmetic and memory-access overhead. In this paper, we propose **Sco**ped **P**osition **E**ncoding (**SCOPE**), a novel framework that reimagines structured sparsity as an intrinsic position encoding mechanism. Instead of relying on explicit arithmetic signals, SCOPE assigns exponentially scaled look-back scopes to attention heads. We theoretically demonstrate that this simple topological constraint transforms multi-head attention into a hierarchical processor, yielding an order awareness horizon that grows exponentially with depth up to the sequence length. Consequently, SCOPE is parameter-free and avoids relying on fragile positional arithmetic. Empirically, it significantly enhances efficiency by masking the majority of attention computations, offering a theoretical $8\times$ reduction in attention FLOPs at long contexts. Extensive evaluations on LLaMA-3-8B architectures reveal that SCOPE achieves superior native length extrapolation and robust retrieval fidelity compared to RoPE, all while substantially reducing training and inference latency. The code is available at https://github.com/oncemoe/ScoPE.

1 Introduction

Large Language Models (LLMs) have fundamentally reshaped the landscape of artificial intelligence, demonstrating remarkable capabilities across tasks ranging from code synthesis to complex logical reasoning (Achiam et al., 2023; Anthropic, 2024; Team et al., 2023; Dubey et al., 2024; Bai et al., 2023; Liu et al., 2024; Zhao et al., 2025). However, despite this success, scaling these models to process long contexts remains a formidable architectural challenge.

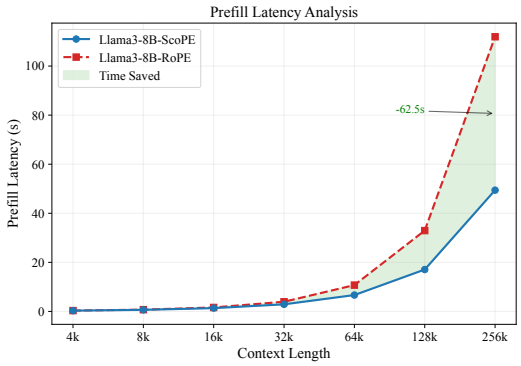


Figure 1: Latency reduction analysis. Comparison of end-to-end prefill latency between RoPE and SCOPE. The shaded green area highlights the substantial time savings achieved by our approach. At the maximum length of 256k, SCOPE halves the prefill latency compared to RoPE ($\approx 2\times$ speedup).

The backbone of modern LLMs, the Transformer architecture (Vaswani et al., 2017), relies on Multi-Head Self-Attention (MHA) to capture dependencies. This power comes at a steep cost: standard MHA exhibits quadratic computational complexity ($O(T^2)$) with respect to sequence length T (Tay et al., 2022), causing inference latency and memory usage to explode as sequence length increases (see Figure 1).

To mitigate these efficiency hurdles, sparse attention mechanisms (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020) have been widely explored. While effective at reducing FLOPs, heuristic or fixed sparsity patterns can compromise the model's ability to capture dense, long-range dependencies, leading to performance degradation in tasks requiring precise retrieval or complex reasoning. Consequently, achieving high computational efficiency without sacrificing the modeling fidelity of full attention remains a central trade-off in long-context modeling.

In this work, we propose a paradigm shift by revisiting this challenge through the lens of Posi-

*Corresponding author: hongyang@zhejianglab.org

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 35653–35673

(a) Input: (x_1, x_2) (b) Input: (x_2, x_1)

Figure 2: Illustration of Lemma 2.2. Even if the scope-2 head (bottom) acts as a "bag-of-words" and cannot distinguish the set $\{x_1, x_2\}$ from $\{x_2, x_1\}$, the scope-1 head (upper) breaks the symmetry by observing only the current token. The aggregation of these two views creates a unique signature for the sequence order.

tion Encoding (PE). Traditionally, PE is treated as an explicit, arithmetic signal—injected via embeddings (Vaswani et al., 2017) or rotational modulations (RoPE) (Su et al., 2024) or attention biases (ALiBi) (Press et al., 2022). However, these explicit methods face two practical limitations in long-context scenarios: (1) they incur additional arithmetic and implementation overhead; and (2) they may be precision-sensitive when scaling to ultra-long sequences under low-precision formats (e.g., BFloat16), as highlighted by Wang et al. (2024). Conversely, recent studies on NoPE (No Positional Encoding) (Haviv et al., 2022; Kazemnejad et al., 2023; Irie, 2025) suggest that causal masks alone can implicitly encode position via "predecessor counting." However, standard NoPE typically suffers from weak performance, failing to match the accuracy of explicit PEs (Haviv et al., 2022).

Instead of treating sparsity and position encoding as separate problems, in this paper, we propose SCOPE, a novel framework that bridges the dichotomy between efficient sparse attention and robust position modeling. We demonstrate that *structured sparsity itself can serve as an implicit mechanism for inducing order awareness*, as shown in Figure 2. Our core innovation lies in configuring attention heads with *exponentially scaled scopes*. By assigning varying look-back windows to different heads, we enable the model to capture dependencies at diverse granularities—from local patterns to global contexts. Crucially, as layers are stacked, the order-awareness horizon grows exponentially with depth, transforming a cascade of sparse layers into a precise sequence processor capable of global order awareness without explicit arithmetic encodings. Consequently, our approach significantly reduces computational overhead while preserving—and in many cases enhancing—the model's ability to capture complex dependencies compared to standard Transformers.

This design yields a "best-of-both-worlds" outcome: it significantly reduces attention FLOPs while achieving superior length generalization and retrieval precision. To validate SCOPE, we conducted comprehensive evaluations spanning language modeling, long-context retrieval, and standard NLU benchmarks. Our contributions are summarized as follows:

- Theoretical Framework for Implicit OA: We propose SCOPE, a mechanism that induces Order Awareness (OA) through exponentially scaled scopes. We provide a theoretical analysis that this hierarchical structure achieves exponential resolution growth with network depth, without injecting explicit positional signals.
- Elastic Structure & Efficiency: We demonstrate that SCOPE exhibits a hierarchical allocation of positional capacity, adapting naturally to varying lengths. This design reduces attention computation by up to $8\times$ (theoretically) and achieves a $2\times$ measured speedup at 256k context end-to-end prefill.
- Superior Extrapolation & Performance: Empirical results on LLaMA-3-8B show that SCOPE achieves remarkable native extrapolation (up to $4\times$ training length) and maintains $> 90\%$ accuracy on "Needle-in-a-Haystack" retrieval at 128k context. Crucially, this efficiency comes at almost no cost to generic capabilities, as SCOPE maintains competitive performance on standard NLU benchmarks.

2 Preliminaries

In this section, we revisit the decoder-only Transformer formulation and formally define *Order Awareness* (OA), a critical property indicating that the model can distinguish sequences based on token order.

2.1 Decoder-Only Transformer

A standard decoder-only Transformer layer (Brown et al., 2020) transforms input $\mathbf{x} \in \mathbb{R}^{T\times d}$ via Causal Multi-Head Attention (MHA) and a Feed-Forward Network (FFN). The core mechanism in MHA for a head h at step t is:

$$\mathbf{o}_t^{(h)} = \sum_{i=1}^{t} \text{Softmax}\left(\frac{\mathbf{q}_t^{(h)\top}\mathbf{k}_i^{(h)}}{\sqrt{d_h}} + m_{t,i}\right)\mathbf{v}_i, \tag{1}$$

where $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v} \in \mathbb{R}^{T\times d_h}$ are queries, keys, and values, projected from the input, and $m_{t,i}$ represents the causal mask. Without explicit positional encodings, self-attention depends on content ($\mathbf{q}^\top\mathbf{k}$) and is largely permutation-equivariant. As a result, the representation at a given position can be insensitive to permutations of its visible context (one-layer attention), leading to limited order awareness (Yun et al., 2020).

2.2 The Goal: Order Awareness.

To overcome this limitation, PE mechanisms are employed to imbue models with *Order Awareness* (OA)—the ability to distinguish sequences based on token order. While prior literature has largely bifurcated into *absolute* or *relative* positioning schemes, we instead focus on the essence of PE: the fundamental capability to distinguish sequences based on token order. We formally define this critical property:

Definition 2.1 (Order Awareness)**.** For a causal self-attention model f, let $f_T : \mathbb{R}^{T\times d} \to \mathbb{R}^d$ denote the representation at last position T when running f on the length-T prefix. For $1 \le t \le T$ and any permutation π on token indices, let $\pi_t(\mathbf{x}_{1:T})$ denote the sequence obtained by applying π to the last t tokens $(x_{T-t+1}, \ldots, x_T)$ while keeping the first $T-t$ tokens unchanged. [1] We say f has OA horizon at least t, denoted $\text{OA}(f) \ge t$, if for almost all input $\mathbf{x} \in \mathbb{R}^{T\times d}$ and any non-identity permutation π_t, the inequality $f_T(\mathbf{x}) \ne f_T(\pi_t(\mathbf{x}))$ holds.

This implies that distinct token orderings yield distinct representations. Then, for sliding window attention, we can derive a straightforward corollary.

Corollary 2.1 (OA of Sliding Window Attention)**.** *Consider a causal self-attention model whose last-position representation $f_T(\mathbf{x})$ depends only on the last w tokens $(x_{T-w+1}, \ldots, x_T)$ (e.g., fixed sliding-window attention of size w). If $\text{OA}(f) \ge w$ holds at length $t = w$, then $\text{OA}(f) \ge w$ holds for all $t \ge w$.*

2.3 Intuition of Sparse-Induced OA

Before detailing methodology, we demonstrate how structured sparsity alone can break the permutation invariance described in Section 2.1.

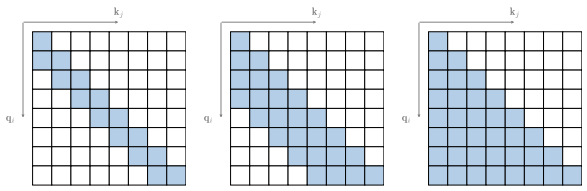
Lemma 2.2 (Length-2 Order Awareness). *Consider an attention layer with two heads h_1 and h_2, with scopes $S_1 = 1$ and $S_2 = 2$, respectively. If the token embeddings are distinct and the relevant projections are non-degenerate, then this layer achieves $\text{OA} \ge 2$.*

As shown in Figure 2, head h_1 can only attend to the current token, while head h_2 can attend to both tokens. Consequently, the fused representation can distinguish (x_1, x_2) from (x_2, x_1). In practice, the residual connection also directly preserves the last-token features, playing a role analogous to a scope-1. We did not observe a significant difference when we clamped the minimal scope to 2 in our experiments.

3 Methodology

In this section, we detail SCOPE, a streamlined mechanism designed to induce OA into multi-head attention by incorporating structured sparsity constraints. By assigning exponentially scaled scopes to attention heads and stacking layers, SCOPE transforms the model into a hierarchical sequence processor.

3.1 Scoped Attention



(a) Head 1 ($S_1 = 2$) (b) Head 2 ($S_2 = 4$) (c) Head 3 ($S_3 = 8$)

Figure 3: Visualization of scoped attention masks ($T = 8, H = 3$). By assigning exponentially growing look-back scopes (e.g., $2^1, 2^2, 2^3$), different heads capture dependencies at varying granularities, from local patterns to global context.

In SCOPE, each attention head h is constrained to a specific look-back distance, or *scope* $S_h \in \mathbb{N}^+$.

[1] When π is applied to the whole sequence, we can omit the subscript of π.

We implement via a modified head-specific mask $\mathcal{M}^{(h)}$ based on Equation 1:

$$\mathcal{M}_{t,i}^{(h)} = \begin{cases} 0 & \text{if } t - S_h < i \leq t, \\ -\infty & \text{otherwise.} \end{cases} \tag{2}$$

In practice, we set:

$$S_h = \lceil \gamma^h \rceil, \quad \text{where } \gamma = T^{\frac{1}{H}}, \tag{3}$$

where T is the maximum sequence length. This distribution ensures that heads collectively cover the context history efficiently.

3.2 Scoped Position Encoding

We now analyze that stacking attention layers with these exponentially scaled scopes leads to OA capability on input sequences, effectively serving as an implicit position embedding.

Theoretical Setup: Consider a stack of L residual blocks:

$$\mathbf{z}^{(\ell)} = \mathbf{z}^{(\ell-1)} + \mathrm{MHA}^{(\ell)}\left(\mathbf{z}^{(\ell-1)}\right), \tag{4}$$

where $\mathbf{z}^{(0)} = \mathbf{x}$. For simplicity, the MHA is defined as the summation of H attention heads: $\sum_{i=1}^{H} \mathrm{softmax}\left((zW_Q^{(i)})(zW_K^{(i)})^\top\right) z$. This abstraction sidesteps rank-related complexities to provide an upper-bound analysis of model capacity. In practice, we find that models are capable of learning these structures through the optimization process.

For clarity, we analyze the case when $\gamma = 2$, which means each layer has heads with scopes $\{2^1, 2^2, \ldots, 2^H\}$ and $2^H \geq T$, though the results generalize to any $\gamma \in (1, 2]$.

Theorem 3.1 (Exponential Order Awareness)**.** *Assume that, within each head's visible window, the attention is almost everywhere sensitive to its inputs. After stacking ℓ scoped masking attention layers, the OA horizon at the last position is at least 2^ℓ (up to T). Consequently, choosing $L \geq \lceil \log_2 T \rceil$ yields $\mathrm{OA}(f) \geq T$.*

The theorem provides a sufficient condition under an almost everywhere sensitivity regime, which is most plausible when attention does not collapse to a few tokens. To complement this, we additionally consider an extreme low-entropy condition where each head behaves as a Top-1 selector. In this regime, we derive a sufficient condition based on the induced input-dependent linear combination and validate it via simulations, showing that scoped attention still leads to near full-sequence OA in extreme settings. Full proofs and experimental details are deferred to Appendix A.

Intuition When $\gamma = 2$, if the previous layer can achieve $\mathrm{OA}(f) = c$, the depth-$(\ell - 1)$ representation of each length-c block (with $c = 2^{\ell-1}$) serves as a block-level identifier; the ℓ-th layer then only needs to disambiguate the relative order of two such identifiers, analogous to resolving order in a length-2 sequence as in Lemma 2.2.

3.3 Properties of SCOPE

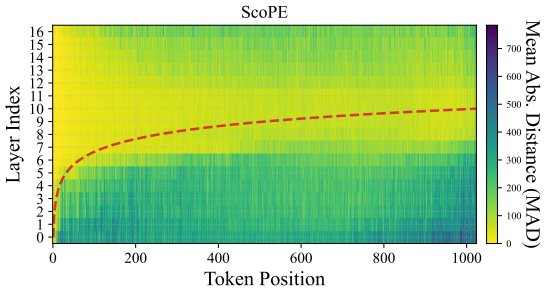
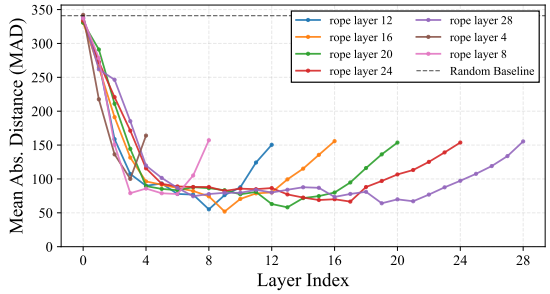


Figure 4: Position reconstruction error map for SCOPE. Brighter areas indicate lower error (higher accuracy). The superimposed red dotted line follows a logarithmic curve, empirically validating that the model's capacity to resolve sequence order grows exponentially with depth.

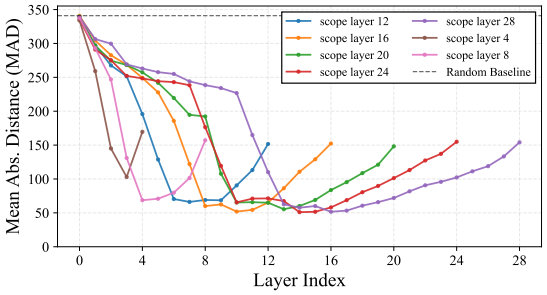
To elucidate the internal mechanisms of SCOPE, we employ linear probing analysis (Haviv et al., 2022). Using a small-scale Qwen3 (Yang et al., 2025) architecture (16 layers, 16 heads, 1024 context length), referred to as *Qwen-Nano*, we train linear classifiers (probes) on hidden states to predict token absolute positions. See Appendix B.1 for detailed configurations.

Exponential Growth of Effective Receptive Field. Figure 4 visualizes the position prediction error across layers and tokens. A clear pattern emerges: the "effective receptive field"—the region where the model can accurately resolve positions (indicated by brighter colors)—expands exponentially with network depth. The boundary of this resolved region aligns closely with a logarithmic curve (red dotted line, $y \propto log_2 x$). This empirical evidence strongly corroborates Theorem 3.1, confirming that SCOPE achieves global order awareness through a cascade of exponentially growing local scopes.

Adaptive Layer Allocation. We further investigate how position encoding behavior scales with



(a) RoPE (Front-loaded)



(b) SCOPE (Elastic)

Figure 5: Layer-wise position probe errors across varying model depths. Each line represents a model with a specific total number of layers. While RoPE (top) always resolves positions in the first ∼4 layers, SCOPE (bottom) scales adaptively, utilizing a proportional depth of the network for structure learning.

model depth by comparing SCOPE against RoPE (Su et al., 2024) across *Qwen-Nano* with varying numbers of layers. As shown in Figure 5, we observe a distinct behavioral divergence:

- **Front-loaded RoPE:** RoPE exhibits a rigid, "front-loaded" pattern. Regardless of the total model depth, it consistently resolves positional information primarily within the first few layers ($\approx$ 4 layers). Subsequent layers contribute minimally to position resolution, presumably focusing on semantic modeling.
- **Elastic SCOPE:** In contrast, SCOPE demonstrates an *adaptive allocation* strategy. Instead of confining position learning to a fixed initial budget, it dynamically partitions the network capacity, utilizing roughly the first half of the layers to progressively build positional context.

This "elastic" property suggests that SCOPE encourages a more distributed structural representation, which may facilitate better adaptation during length extension or fine-tuning compared to the rigid encoding of RoPE. For extended comparisons, including ALiBi, please refer to Appendix B.2.

3.4 Efficient Implementation

Implementation. Exponentially growing windows often expand too rapidly in practice. For large H, window sizes can far outpace the training length T. Simply capping these scopes at T causes multiple heads to collapse into redundant full-window attention, thereby diminishing both scope diversity and sparsity. To maintain a heterogeneous distribution of scopes, we constrain the maximum scope size S_{max} to maximum context length T and choose the growth factor as $\gamma = T^{1/H}$.

We leverage the *FlexAttention* framework (He et al., 2024) for efficient training. As shown in Figure 22 in Appendix E, the scope mask can be defined logically without materializing the full $T \times T$ matrix.

Complexity. Standard causal attention costs $O(\frac{1}{2}HT^2)$. SCOPE reduces this by limiting computation to active scopes. With geometric scopes $S_h = \gamma^h$ (where $S_H = T$), the cost is proportional to $T \sum S_h$. The reduction ratio approximates:

$$\text{Ratio} \approx \frac{\frac{1}{2}HT^2}{\frac{\gamma}{\gamma-1}T^2} = \frac{H(\gamma-1)}{2\gamma}. \quad (5)$$

When $\gamma = 2$ and $H = 32$, this yields a theoretical $\mathbf{8}\times$ attention FLOPs reduction, enabling efficient long-context processing.

4 Experiments

In this section, we evaluate SCOPE across three dimensions: (1) Length Extrapolation, (2) Retrieval Fidelity in ultra-long contexts, and (3) Downstream Capabilities on both general NLU and long-context benchmarks.

4.1 Experimental Setup

We conduct experiments using the LLaMA-3-8B architecture (Dubey et al., 2024). To rigorously benchmark long-context capabilities, we adopt a progressive training protocol consisting of three stages: Pre-training from scratch (4k context), Long-Context Fine-tuning (32k context), and Ultra-Long Adaptation (128k context). To demonstrate architectural universality, we also provide training details and results for Qwen architectures in Appendix D.

Baselines. For SCOPE, we trained a base model from scratch on 4k contexts (ScoPE-Base) and then fine-tuned it on 32k and 128k contexts. In the context scaling, we adopt a simple rule that aligns

the maximum scope $S_{\max}$ with the current max sequence length T, without introducing additional extrapolation heuristics. We compare against the following baselines.

(i) RoPE (Su et al., 2024): The standard positional encoding in modern LLMs. We pre-train the RoPE baseline at 4k and fine-tune it at 32k and 128k contexts. for long-context evaluation, we apply YaRN (Peng et al., 2024) during long-context evaluation for fair comparison.

(ii) ALiBi (Press et al., 2022): A linear-bias-based approach for length extrapolation. Due to memory constraints in our current implementation—where materializing the full bias matrix leads to Out-of-Memory (OOM) errors—we omit ALiBi from certain long-context experiments.

(iii) NoPE (Haviv et al., 2022): A dense baseline without explicit PE. Following the same protocol as SCOPE, we trained NoPE from scratch on a 4k context (NoPE-Base) and subsequently fine-tuned it on 32k and 128k contexts.

(iv) RoPE-SWA (Jiang et al., 2023): A sparse attention baseline using the same backbone as RoPE with an 8k sliding window, fine-tuned from the 128k RoPE checkpoint described in (i). It serves to test if local attention with explicit PE can rival the hierarchical global awareness of our approach.

Implementation. Models are trained using Torchtitan (Liang et al., 2025) on NVIDIA A100 clusters, utilizing FlexAttention (He et al., 2024) for efficient kernel implementation. For inference, we implemented a sliding-window-per-head version of FlashAttention (Dao et al., 2022) using Triton (Tillet et al., 2019). Detailed configurations for hyperparameters, training protocols, and data mixtures are provided in Appendix C.

4.2 Training Dynamics and Efficiency

One of the most compelling findings is the superior convergence behavior of SCOPE. As illustrated in Figure 6, SCOPE consistently maintains a lower training loss compared to the RoPE baseline throughout the pre-training stage. Notably, this performance advantage persists during the subsequent long-context fine-tuning stages (see Appendix Figure 16 for 32k and 128k loss curves), indicating that our method provides a more efficient optimization landscape.

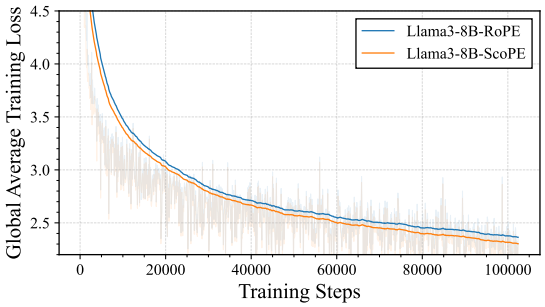


Figure 6: Training loss comparison on LLaMA-3-8B (Pre-training Stage). SCOPE (Orange) consistently achieves lower training loss compared to the RoPE baseline (Blue) throughout the training run. This indicates that the structured sparsity imposed by SCOPE serves as an effective inductive bias, facilitating more efficient convergence.

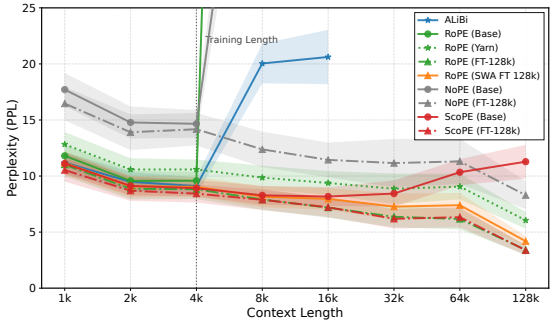


Figure 7: Perplexity extrapolation on long documents (up to 128k). We compare: (1) *Base*: Native 4k-trained models; (2) *Fine-tuned*: Models adapted to 128k. SCOPE exhibits superior native extrapolation (up to 64k) compared to RoPE.

4.3 Long-Context Modeling & Extrapolation

We utilize Perplexity (PPL) as the primary metric to evaluate long-range modeling and extrapolation capabilities. We evaluate models on a curated subset of 100 long documents (lengths ranging from 64k to 128k) from the *Proof-Pile* (Azerbayev et al., 2023) and *Gov-Report* (Huang et al., 2021) datasets. To measure performance stability across varying lengths, we compare the zero-shot perplexity of the last 256 tokens across different input lengths. Results are summarized in Figure 7.

In the *Base* regime (trained on 4k), SCOPE demonstrates exceptional robustness. While RoPE-Base (w/o Yarn) and ALiBi fail catastrophically immediately beyond the training window, SCOPE maintains stability up to 16k tokens ($4\times$ training length) without any parameter updates. Furthermore, even within the standard 32k window, the base SCOPE model yields consistently lower per-

plexity than the RoPE (w Yarn) baseline. This confirms that our hierarchical scope structure naturally generalizes to unseen lengths. For post-adaptation (128k fine-tuning), both models converge to comparable low perplexity, confirming that SCOPE preserves the capacity to learn from long-context data while offering superior initialization properties.

4.4 Needle-in-a-Haystack (NIAH)

We assess fine-grained retrieval fidelity using the Needle-In-A-Haystack (NIAH) Passkey Retrieval task (Kamradt, 2023). The results are visualized in Figure 8.

Zero-shot Extrapolation. We first evaluate base models (trained on 4k context) without fine-tuning. SCOPE exhibits remarkable intrinsic extrapolation capabilities. As shown in Figure 8b, it maintains high retrieval accuracy up to ∼16k tokens—4× its training context—before degradation. In contrast, the RoPE baseline (Figure 8a) fails to generalize, with performance collapsing immediately beyond the ∼6k boundary.

Ultra-Long Adaptation. Following 128k fine-tuning, SCOPE demonstrates superior consistency over the full context window. While RoPE (Figure 8c) suffers from attention degradation at deeper positions (averaging 86% accuracy), SCOPE (Figure 8d) preserves sharp attention focus, achieving 92% average accuracy and effectively eliminating "lost needles" across the entire 128k sequence. We additionally report 128k fine-tuning results for two strong baselines, NoPE and RoPE-SWA, in Appendix C.5; both remain strongly local (Avg. Acc. 0.33 and 0.34, respectively), succeeding mainly when the needle appears near the end of the context.

4.5 Downstream Benchmark Performance

We assess SCOPE in two distinct regimes: (1) General Natural Language Understanding (NLU), to ensure structural sparsity does not compromise fundamental modeling capabilities; and (2) Real-world Long-Context Understanding via LongBench.

General NLU Benchmarks. We evaluate the models on a suite of standard zero-shot and few-shot benchmarks utilizing the `lm-evaluation-harness` library (Gao et al., 2024). For a detailed breakdown of the evaluation protocol, including the specific number of few-shot examples and metric specifications for each task, please refer to Table 6 in Appendix C.3.

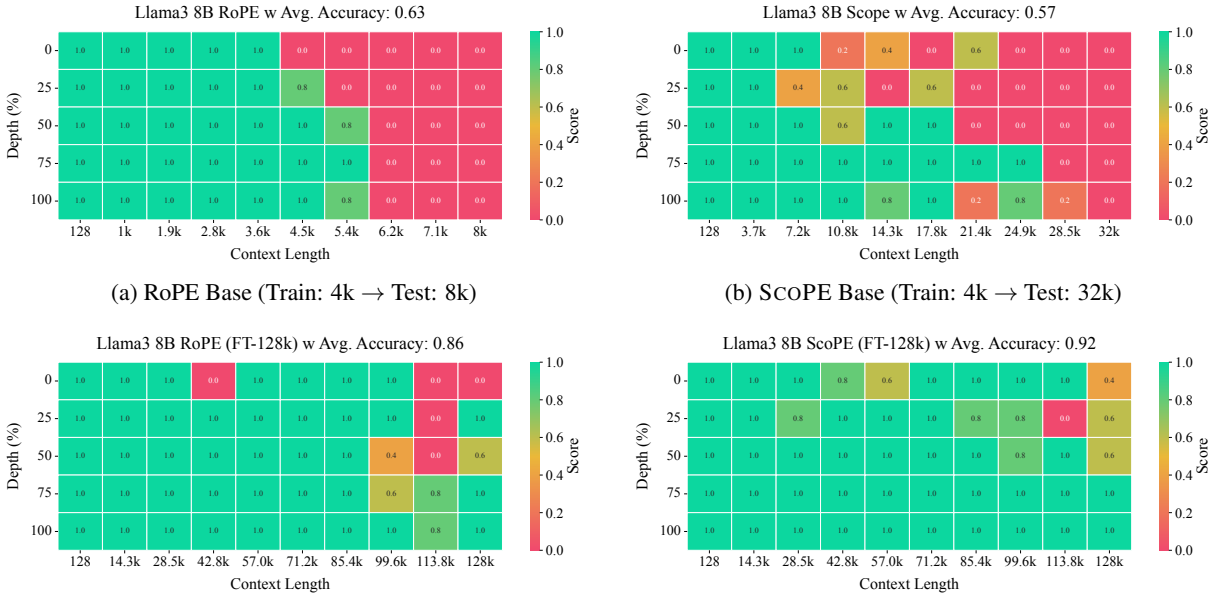
Table 1 presents the comparison. Despite enforcing structural sparsity, SCOPE maintains highly competitive performance. Notably, it outperforms RoPE on reasoning tasks such as ARC-Challenge (Clark et al., 2018) and HellaSwag (Zellers et al., 2019), suggesting that the hierarchical structure may benefit semantic abstraction. While there is a slight regression in logic-heavy tasks like GPQA (Rein et al., 2024), the overall performance confirms that SCOPE is a robust general-purpose mechanism. Furthermore, the performance gain from 4k to 32k indicates support for continuous learning without catastrophic forgetting.

LongBench Performance. To evaluate real-world capabilities, we utilize LongBench (Bai et al., 2024). Table 2 summarizes the results. After finetuning, SCOPE outperforms the dense baseline (RoPE + YaRN) in Few-shot Learning and Single-Document QA. Although strict-syntax tasks like Code Completion see minor degradation—likely due to the sensitivity of code to local mask constraints—SCOPE remains highly effective across most categories, offering a favorable trade-off between performance and efficiency.

4.6 Computational Efficiency

We stress-test the prefill phase with sequence lengths up to 256k tokens using a chunked prefill strategy (Agrawal et al., 2025), where the reported metrics represent the average of 10 independent runs. As shown in Figure 1 (Introduction) and Figure 9, SCOPE demonstrates significant gains. At 128k context, we observe a 2× speedup, expanding to 2.1× at 256k. We also provide a training time comparison in Appendix C.4.

These results confirm that SCOPE effectively mitigates the quadratic computational bottleneck of attention. Furthermore, we highlight a significant avenue for future engineering: converting the dense KV-cache to a sparse storage format. While our current implementation focuses on compute acceleration, a sparse KV-cache implementation would theoretically reduce memory usage proportional to the FLOPs savings, which we leave to future work.



(a) RoPE Base (Train: 4k → Test: 8k)

(b) SCOPE Base (Train: 4k → Test: 32k)

(c) RoPE Fine-tuned (128k Context)

(d) SCOPE Fine-tuned (128k Context)

Figure 8: Needle-in-a-Haystack (NIAH) Heatmaps. Top Row (Zero-shot Extrapolation): Comparison of base models trained on 4k context. (a) RoPE fails to retrieve information beyond ∼6k tokens. (b) SCOPE surprisingly maintains retrieval capabilities up to ∼16k tokens without any fine-tuning. Bottom Row (128k Adaptation): Comparison of models fine-tuned on 128k context. (c) RoPE achieves 86% average accuracy with visible degradation. (d) SCOPE achieves 92% average accuracy, demonstrating superior information retention over ultra-long sequences.

Table 1: Zero-shot and Few-shot performance on standard NLP benchmarks. We compare models at both the Pre-training stage (Base) and Long-Context Fine-tuning stage (32k). SCOPE achieves comparable or superior performance on 8 out of 10 tasks, validating that structural sparsity does not compromise general language capabilities.

Model	GPQA	BBH	Wino	ARC-C	Hella	BoolQ	TruthfulQA	Lambada	OBQA	PIQA
Stage 1: Pre-training (4k context)										
RoPE-Base	**0.255**	**0.250**	0.542	0.285	0.483	0.543	0.379	0.469	0.324	0.705
SCOPE-Base	0.243	0.229	**0.568**	**0.299**	**0.516**	**0.551**	**0.383**	**0.480**	**0.330**	**0.714**
Stage 2: Long-Context Fine-tuning (32k context)										
RoPE-32k	**0.268**	**0.242**	0.529	0.288	0.482	0.567	0.382	0.496	0.328	0.708
SCOPE-32k	0.248	0.232	**0.566**	**0.307**	**0.518**	**0.577**	**0.384**	**0.508**	**0.338**	**0.721**

5 Related Work

5.1 Positional Encodings

Position modeling has evolved from Absolute Positional Embeddings (Vaswani et al., 2017; Devlin et al., 2019) to Relative schemes (RPE) (Shaw et al., 2018; Raffel et al., 2020) and Rotary Embeddings (RoPE) (Su et al., 2024), with recent bias-based methods like ALiBi (Press et al., 2022) improving extrapolation. However, explicit methods introduce additional arithmetic and may suffer from precision issues under low-precision ultra-long contexts (Wang et al., 2024). In contrast, SCOPE induces order awareness structurally via hierarchical patterns rather than explicit arithmetic. We discuss the theoretical connection between SCOPE and ALiBi in Appendix B.4.

5.2 Length Extrapolation

Extending context windows typically requires post-hoc interpolation (e.g., PI (Chen et al., 2023), YaRN (Peng et al., 2024)) or efficient fine-tuning (Chen et al., 2024). Unlike these methods, SCOPE handles length variation intrinsically by simply expanding scope bounds. Moreover, our approach is orthogonal to attention-logit scaling techniques (Peng et al., 2024; Nakanishi, 2025), allowing potential integration with advanced extrapolation tricks for future enhancements.

Table 2: LongBench Results (Average Scores). Comparison between finetuned models: RoPE (with YaRN extrapolation) and SCOPE at 32k and 128k checkpoints. SCOPE shows strong performance in Few-shot learning and QA tasks.

	Context: 32k		Context: 128k	
Task Category	**RoPE**	**SCOPE**	**RoPE**	**SCOPE**
Code Completion	**0.261**	0.232	**0.263**	0.218
Few-shot Learning	0.385	**0.410**	0.396	**0.402**
Multi-Document QA	**0.063**	0.059	**0.067**	0.057
Single-Document QA	0.061	**0.073**	0.060	**0.072**
Summarization	**0.127**	0.121	**0.121**	0.119
Synthetic Tasks	0.028	**0.033**	0.031	**0.033**

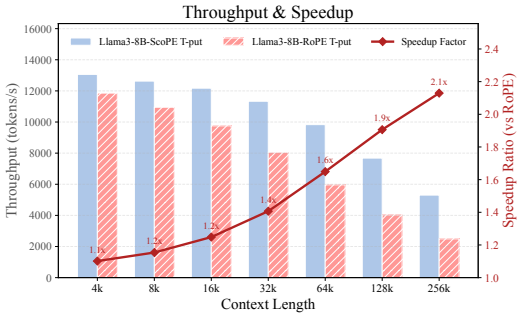


Figure 9: Prefill performance on Llama-3-8B. Comparison of throughput (bars) and speedup ratio (line) demonstrates that SCOPE effectively handles long-context inference. At 256k tokens, we achieve a 2.1× speedup over the RoPE baseline.

5.3 Implicit Positional Awareness (NoPE)

Recent studies suggest decoder-only models possess implicit order awareness via predecessor counting (Haviv et al., 2022), with Irie (2025). However, standard NoPE mechanisms rely on the gradual accumulation of counts across many layers to resolve positions. SCOPE *amplifies* this latent capability. By enforcing exponentially varying receptive fields, we transform implicit counting into an explicit hierarchical feature, enabling the model to resolve sequence order with exponential efficiency compared to standard NoPE transformers.

5.4 Efficient and Sparse Attention

Sparse attention variants like Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020), and newer block-based methods (Yuan et al., 2025; Lu et al., 2025) primarily aim to reduce $O(T^2)$ complexity. While SCOPE inherently incorporates sparsity, its fundamental motivation differs: we leverage structured sparsity primarily as a mechanism for implicit positional encoding, rather than solely as a heuristic for approximation.

It is worth noting that since our largest scope covers the full context history, the asymptotic complexity of SCOPE remains $O(T^2)$. However, by sparsifying the majority of heads with shorter scopes, we significantly reduce the computational coefficient. This design choice allows us to optimize training and inference efficiency with less compromise than strictly linear approximations.

Structurally, the closest parallel is Mistral's Sliding Window Attention (SWA) (Jiang et al., 2023; Xiao et al., 2024). Crucially, however, SWA applies a uniform window size, limiting the effective receptive field. In contrast, SCOPE employs varying scopes. This design creates a hierarchical resolution that preserves global context, effectively preventing the horizon blindness associated with fixed-window approaches.

6 Conclusion

In this work, we propose SCOPE, a novel mechanism that induces positional awareness in Transformers solely through structured sparsity. By replacing explicit arithmetic positional encodings with exponentially scaled attention scopes, we transform the model into a hierarchical sequence processor capable of resolving global order from local views.

Our theoretical analysis and empirical observations confirm that SCOPE achieves long-horizon OA that grows exponentially with depth, exhibiting an elastic allocation of positional capacity that adapts naturally to sequence length. Experimentally, SCOPE demonstrates superior properties over the prevailing RoPE baseline: it converges faster during pre-training, exhibits remarkable native length extrapolation (up to $4\times$ the training context) in perplexity experiment, and maintains high retrieval fidelity in ultra-long contexts up to 128k tokens. Furthermore, this structural efficiency is achieved without compromising performance on general NLU tasks.

These findings challenge the necessity of explicit, arithmetic-heavy positional embeddings, suggesting that appropriate topological constraints alone are sufficient for robust sequence modeling. Future work will explore scaling SCOPE to larger parameter regimes and investigating its synergy with other long-context techniques such as attention-logit scaling and state-space models.

Limitations

Despite the promising results, our work has several limitations that invite future research:

Theoretical Assumptions. Our analysis relies on a non-degeneracy assumption (e.g., almost-everywhere sensitivity) of the attention mapping within each head's scope, and uses a simplified residual MHA stack that omits LN/FFN. In practical finite-width Transformers, information can be compressed or distorted across layers, which may reduce the guaranteed OA horizon in worst-case high-entropy inputs.

Engineering Realization. While SCOPE theoretically reduces FLOPs, translating these gains into wall-clock speedups requires specialized kernels. We currently rely on *FlexAttention* (He et al., 2024) and Triton kernels (Tillet et al., 2019); naive implementations lacking kernel fusion may fail to fully manifest the efficiency advantages of our method.

Learnable Scopes. Learnable or input-dependent scopes introduce dynamic sparsity patterns, which can be difficult to compile into efficient static banded/block-sparse kernels and may increase kernel launch overhead and memory traffic, and we leave this for future work.

Generalization to Other Architectures. SCOPE is primarily designed for Decoder-only (Autoregressive) models where the "scopes" act as a look-back window. Its applicability to Encoder-only (Bidirectional) models or multi-dimensional modalities (e.g., 2D images, 3D point clouds) remains underexplored. Extending SCOPE to these non-causal settings—potentially by adapting the "scopes" from a "left-aligned" causal window to a "center-aligned" neighborhood—is a promising avenue for future work.

Acknowledgments

This work is supported in part by National Key R&D Program of China (2023YFB4502400), in part by National Natural Science Foundation of China under Grant 62271452.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Arney Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S. Gulavani, Alexey Tumanov, and Ramachandran Ramjee. 2025. Efficient llm inference via chunked prefills. *SIGOPS Oper. Syst. Rev.*, 59(1):9–16.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics. *Preprint*, arXiv:2310.10631.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 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, Bangkok, Thailand. Association for Computational Linguistics.

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

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Valérie Castin, Pierre Ablin, and Gabriel Peyré. 2024. How smooth is attention? In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

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

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024. Longlora: Efficient fine-tuning of long-context large language models. *Preprint*, arXiv:2309.12307.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *Preprint*, arXiv:1904.10509.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: fast and memory-efficient exact attention with io-awareness. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

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

Horace He, Driss Guessous, Yanbo Liang, and Joy Dong. 2024. Flexattention: The flexibility of pytorch with the performance of flashattention. https://pytorch.org/blog/flexattention/. Accessed: 2025-01-01.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. *Preprint*, arXiv:2104.02112.

Kazuki Irie. 2025. Why are positional encodings nonessential for deep autoregressive transformers? a petroglyph revisited. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 551–559, Vienna, Austria. Association for Computational Linguistics.

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. *Preprint*, arXiv:2310.06825.

Greg Kamradt. 2023. Needle in a haystack - pressure testing LLMs. https://github.com/gkamradt/LLMTest_NeedleInAHaystack. GitHub repository, Accessed: 2025-10-20.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. 2022. The stack: 3 tb of permissively licensed source code. *Preprint*, arXiv:2211.15533.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, and 48 others. 2023. Starcoder: may the source be with you! *Preprint*, arXiv:2305.06161.

Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, Sanket Purandare, Gokul Nadathur, and Stratos Idreos. 2025. Torchtitan: One-stop pytorch native solution for production ready LLM pretraining. In *The Thirteenth International Conference on Learning Representations*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024. Deepseek-v2: A strong, economical, and efficient

mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, and 6 others. 2025. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Ken M. Nakanishi. 2025. Scalable-softmax is superior for attention. *Preprint*, arXiv:2501.19399.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C).

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6).

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Philippe Tillet, H. T. Kung, and David Cox. 2019. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, MAPL 2019, page 10–19, New York, NY, USA. Association for Computing Machinery.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

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

Haonan Wang, Qian Liu, Chao Du, Tongyao Zhu, Cunxiao Du, Kenji Kawaguchi, and Tianyu Pang. 2024. When precision meets position: Bfloat16 breaks down rope in long-context training. *Preprint*, arXiv:2411.13476.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. 2025. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23078–23097, Vienna, Austria. Association for Computational Linguistics.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: transformers for longer sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Theoretical Analysis

In this section, we provide theoretical and empirical evidence that SCOPE enables exponential order awareness by stacking layers, effectively achieving OA on long sequences.

A.1 A Simplified Transformer Block

A standard Transformer block is typically formulated as

$$\mathbf{h} = \mathbf{x} + \mathrm{MHA}(\mathrm{LayerNorm}(\mathbf{x})),$$
$$\mathbf{z} = \mathbf{h} + \mathrm{FFN}(\mathrm{LayerNorm}(\mathbf{h})).$$

Here $\mathbf{x} = [x_1, \ldots, x_t]$ is the input sequence of token embeddings. Since the ability to discern sequence order is primarily governed by the attention mechanism, we analyze the following residualized form, which isolates the contribution of attention to order information:

$$\mathbf{z} = \mathbf{x} + \mathrm{MHA}(\mathbf{x}). \tag{6}$$

This simplification is motivated by empirical observations that LayerNorm and FFN modules typically do not induce representation collapse that would erase order information injected by attention. Furthermore, we assume that head outputs are summed rather than concatenated. we remove the output mixing projections of attention and set the value projection to identity mapping to focus on the structural impact. [2]

A.2 When Attention is Sufficiently Sensitive

It is generally assumed that attention layers exhibit almost-everywhere sensitivity to any input perturbations within the visible window (Castin et al., 2024). Under the almost-everywhere local sensitivity assumption, any nontrivial change to the visible window alters the head output for almost all inputs. We prove exponential growth of the order-aware horizon by induction on depth.

Proof. We proceed by induction on depth ℓ.

Base Case ($\ell = 1$): Consider a sequence of length 2 denoted as $\mathbf{x} = (x_1, x_2)$. In the first layer, the output $z_2^{(1)}$ is a weighted sum of the tokens plus a residual connection, which can be simplified as:

$$\begin{aligned} z_2^{(1)}(\mathbf{x}) &= x_2 + \big(\alpha x_1 + (1-\alpha)x_2\big) \\ &= \alpha x_1 + (2-\alpha)x_2, \end{aligned}$$

where α represents the attention weight. Under the attention almost everywhere sensitivity assumption, $\alpha \in (0, 1)$, therefore $\alpha \neq 2 - \alpha$. Consequently, swapping the input results in a different output as long as $x_1 \neq x_2$. Thus, the model satisfying $\mathrm{OA}(f) \geq 2^1$.

Inductive Step: Consider a sequence of length $2c$ denoted as $\mathbf{x} = (x_1, \ldots, x_{2c})$. The output at layer ℓ is:

$$z_{2c}^{(\ell)} = z_{2c}^{(\ell-1)} + \sum_{j=1}^{2c} \alpha_j z_j^{(\ell-1)},$$

[2] This analysis serves as an idealized characterization of the structural mechanism; it does not account for potential precision loss or challenges in extremely long contexts.

where α_j are the attention weights. Assume layer $(\ell-1)$ achieves OA $\geq c$ at all position $t \geq c$, meaning that any non-identity permutation within the window c changes $z_t^{(\ell-1)}$. We partition the window of length $2c$ into a prefix block $\{0, \ldots, c\}$ and a suffix block $\{c+1, \ldots, 2c\}$. We analyze two primary cases for the permutation.

Case (i): Suffix changes. The residual connection preserves distinct features and the sum is highly unlikely to exactly cancel out the change in the residual term. Thus, $z_{2c}^{(\ell)}$ changes.

Case (ii): Prefix changes while suffix is fixed. In this case, the residual term $z_{2c}^{(\ell-1)}$ remains unchanged. However, at least $z_c^{(\ell-1)}$ will change by the inductive hypothesis. Then under the almost everywhere sensitivity assumption, the weighted sum will change.

Therefore, in both cases, $z_{2c}^{(\ell)}(\mathbf{x}) \neq z_{2c}^{(\ell)}(\pi_{2c}(\mathbf{x}))$, implying OA $\geq 2c$ at depth ℓ. Iterating over layers yields an exponentially growing order-aware horizon. □

In recent LLM architectures, the attention mechanism typically comprises multiple heads (e.g., 32 heads in LLaMA3-8B). Consequently, 2^{32} is far larger than the maximum context length required in current applications. We therefore restrict our analysis to $\gamma \in (1, 2]$, simplifying the proof by eliminating the discussion of order-blind block swaps.

A.3 When Attention is Top-1 Selection

Well-trained models may exhibit sparse attention. We simulate this using Top-1 hard selection. For qualitative analysis, we assume zero-entropy attention, where each head selects the single most relevant token from its context window, denoted as $\mathrm{Top1}(\cdot)$. To avoid modeling content, we approximate this by randomly selecting one token within the window. The layer operations are thus: For NoPE,

$$z_t = x_t + \sum_{k=1}^{H} \mathrm{Top1}_k(x_{1:t}).$$

For SCOPE,

$$z_t = x_t + \sum_{k=1}^{H} \mathrm{Top1}_k\left(x_{\max(1,\, t-2^k+1):t}\right).$$

Under these assumptions, the transformer block becomes a (data-dependent) linear combination of inputs. We use $r = u/T$, the ratio of unique coefficients in the linear combination u with the sequence length T, as a practical proxy for OA, as formalized by the following sufficient condition.

Lemma A.1 (A sufficient condition for OA)**.** *Suppose the final representation at position t can be written as $z_t = \sum_{i=1}^{t} \alpha_i x_i$. If the coefficients $\{\alpha_i\}_{i=1}^{t}$ are pairwise distinct, then for any non-identity permutation π, we have $z_t(\mathbf{x}) \neq z_t(\pi(\mathbf{x}))$ holds for almost all sequences $\mathbf{x}$.*

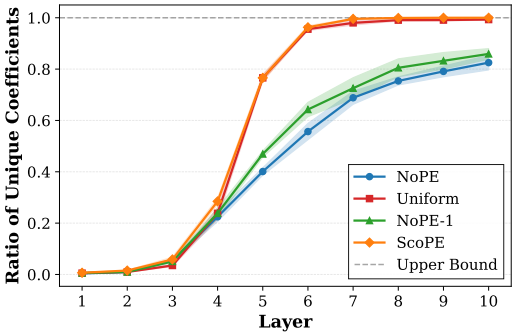
Sketch. The equality $z_t(\mathbf{x}) = z_t(\pi(\mathbf{x}))$ implies $\sum_{i=1}^{t}(\alpha_i - \alpha_{\pi(i)})x_i = 0$. Since the coefficients are pairwise distinct, $\alpha_i \neq \alpha_{\pi(i)}$ for at least one index where $i \neq \pi(i)$. Thus, the equation represents a non-trivial linear combination of the input embeddings summing to zero. The solutions to this form a proper linear subspace of $(\mathbb{R}^d)^t$, which has Lebesgue measure zero. □

Remark. The diversity (or uniqueness) of the coefficients across positions serves as a simple proxy for the attention mechanism's ability to encode order.

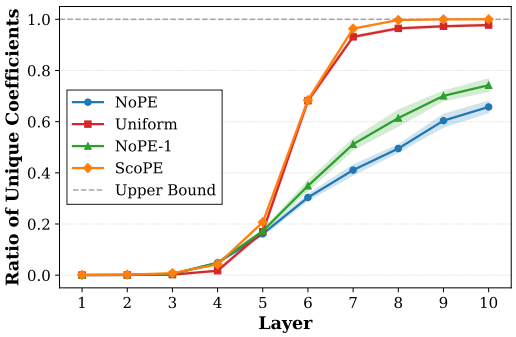
We hypothesize that SCOPE can still achieve OA of the full input sequence in this regime through its exponentially scaled receptive fields. We validate this by simulating random transition matrices M under various window configurations, multiplying them across layers, and computing the ratio r for the final row. When $r = 1$, the mapping assigns distinct weights to all positions, which (by Lemma A.1) indicates order awareness on input sequence.

Experimental Setup. In our experiments, we set the number of attention heads H and the sequence length T to satisfy $2^H = T$, and we set the maximum number of layers to 10. We primarily consider the following attention-window configurations:

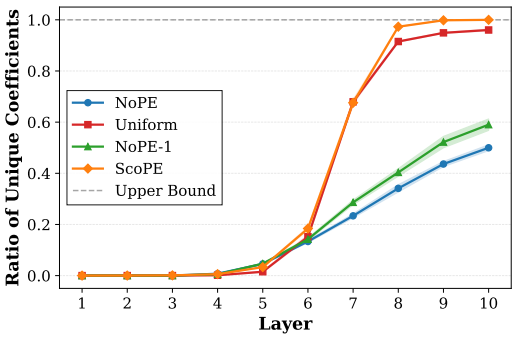
1. (NoPE): no restriction is imposed on the attention span.
2. (SCOPE): the window sizes are set to $[2^1, 2^2, \ldots, 2^H]$.
3. (NoPE-1): $H - 1$ heads use NoPE, and one head uses a window size of 1.
4. (Uniform): the window sizes grow uniformly as $\left[\, \lfloor T/i \rfloor \;\middle|\; i \in \{1, 2, \ldots, H\} \,\right]$.



(a) Sequence length 512 with 10 heads.



(b) Sequence length 4096 with 13 heads.



(c) Sequence length 32768 with 16 heads.

Figure 10: The ratio $r = u/T$ of unique coefficients in the induced linear mapping to the last position under different window configurations.

Specifically, we calculate

$$v = eM_1M_2 \cdots M_{\mathrm{L}},$$

where $e = [0, \ldots, 0, 1]$ is a row vector selecting the last row, and M_ℓ is the transition matrix for layer ℓ (see Figure 24 for the construction logic). A higher ratio r of unique values in v indicates better order awareness of the stacked layers, and $r = 1$ indicates that the stacked layers can achieve OA over the entire sequence.

Results. Figure 10 illustrates the results averaged over 10 random trials for $T \in \{512, 4096, 32768\}$. ScoPE consistently achieves $r \approx 1$ within 10 layers, implying its capacity for full-sequence order awareness. The Uniform baseline exhibits slower convergence toward 1. While the NoPE-1 configuration outperforms the vanilla NoPE baseline, it remains substantially less effective than Uniform.

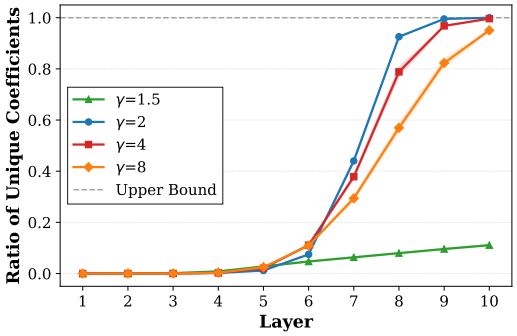


Figure 11: Sensitivity to the growth factor γ under the Top-1 proxy (sequence length $T = 32768$, $H = 16$). Performance remains robust as long as the largest scope covers the sequence.

Sensitivity to the growth factor γ. The above experiment corresponds to $\gamma = 2$, i.e., window sizes grow as 2^k. We further vary γ while keeping $T = 32768$ and $H = 16$, using window sizes $[\gamma^1, \gamma^2, \ldots, \gamma^H]$ (capped at T once exceeding the sequence length). Figure 11 shows that performance is robust as long as the maximum scope covers the whole sequence, i.e., $\gamma^H \geq T$, in which case r still rapidly approaches 1. We also observe that larger γ may require more layers to reach $r \approx 1$ (e.g., $\gamma = 8$).

Summary. We analyze attention under two extreme regimes: an almost-everywhere sensitivity regime (plausible early in training) and a Top-1 selection regime (a proxy for low-entropy attention after convergence). In both cases, ScoPE admits full sequence order-awareness. Real models typically operate between these extremes, which helps explain why the error map in Figure 4 exhibits an approximately logarithmic boundary.

B Supplementary of Probing

B.1 Experiment Configurations

We performed probing experiments using the *Qwen-Nano* architecture, trained from scratch on the Refined-Web dataset (Penedo et al., 2023) with the Llama tokenizer (Touvron et al., 2023) on NVIDIA A100 GPUs. Detailed hyperparameters are provided in Table 3.

For probing, we sampled 100 sequences (length > 1024) from the Proof-Pile dataset (Azerbayev

et al., 2023). Following Haviv et al. (2022), we extracted layer-wise hidden states and trained a 2-layer MLP probe on 80% of the data for 5 epochs to predict absolute token positions. The remaining 20% served as the test set for calculating the Mean Absolute Distance (MAD) error.

B.2 Comparative Analysis of PE Dynamics

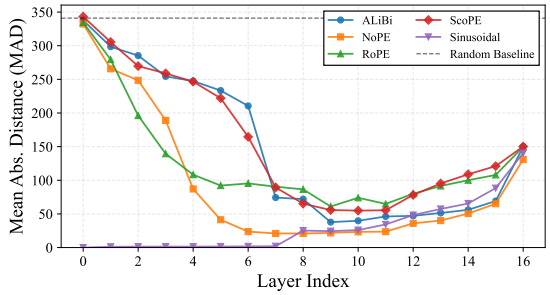


Figure 12: Layer-wise probing error across different PEs. SCOPE exhibits a learning trajectory most similar to ALiBi, characterized by a smooth, progressive resolution of positions in early layers.

Layer-wise Functional Transition. Figure 12 illustrates that most positional encoding (PE) schemes (excluding sinusoidal) exhibit a "U-shaped" error curve. This trend aligns with Haviv et al. (2022), suggesting a network-wide functional transition: early layers prioritize *position resolution* (reconstructing structural order), while deeper layers shift focus to *semantic prediction*, leading to a gradual abstraction of exact positions.

Note that reduced probe performance in deeper layers may also reflect the nonlinear integration of positional signals into semantic representations, rather than a loss of information.

Dynamics Comparison. Notably, SCOPE's learning dynamic closely mirrors ALiBi. Both achieve a smooth, concave resolution of position in initial layers. In contrast, RoPE and NoPE exhibit a sharper, almost immediate resolution trajectory. This suggests SCOPE encourages the model to progressively aggregate *local* structural information into global awareness hierarchically, rather than resolving global positions strictly in the first few layers.

Visualizing Representation. Figure 13 provides a granular visualization. SCOPE effectively synthesizes the strengths of existing methods: it mirrors the structural clarity of ALiBi in early layers while maintaining the robust representational capacity characteristic of RoPE in deeper layers.

B.3 Synergy with Causal Masking

Our theoretical analysis (Theorem 3.1) establishes a sufficient condition where doubling the scope size guarantees global order awareness. However, empirical results (e.g., Figure 5) indicate that SCOPE can resolve positions faster than this bound suggests (e.g., an 8-layer model resolving 1024 positions in ~ 4 layers).

This acceleration occurs because SCOPE operates in *synergy* with the causal mask. The causal mask inherently leaks positional information by encoding the number of predecessors (Haviv et al., 2022; Irie, 2025). SCOPE amplifies this intrinsic "NoPE" mechanism. However, this synergy has limits: when layer depth is severely restricted, the benefits of the exponential cascade cannot fully materialize. As shown in Figure 14, performance degrades in very shallow networks (2-4 layers), where RoPE outperforms SCOPE. As depth increases, the hierarchical advantage of SCOPE dominates.

B.4 Theoretical Connection with ALiBi

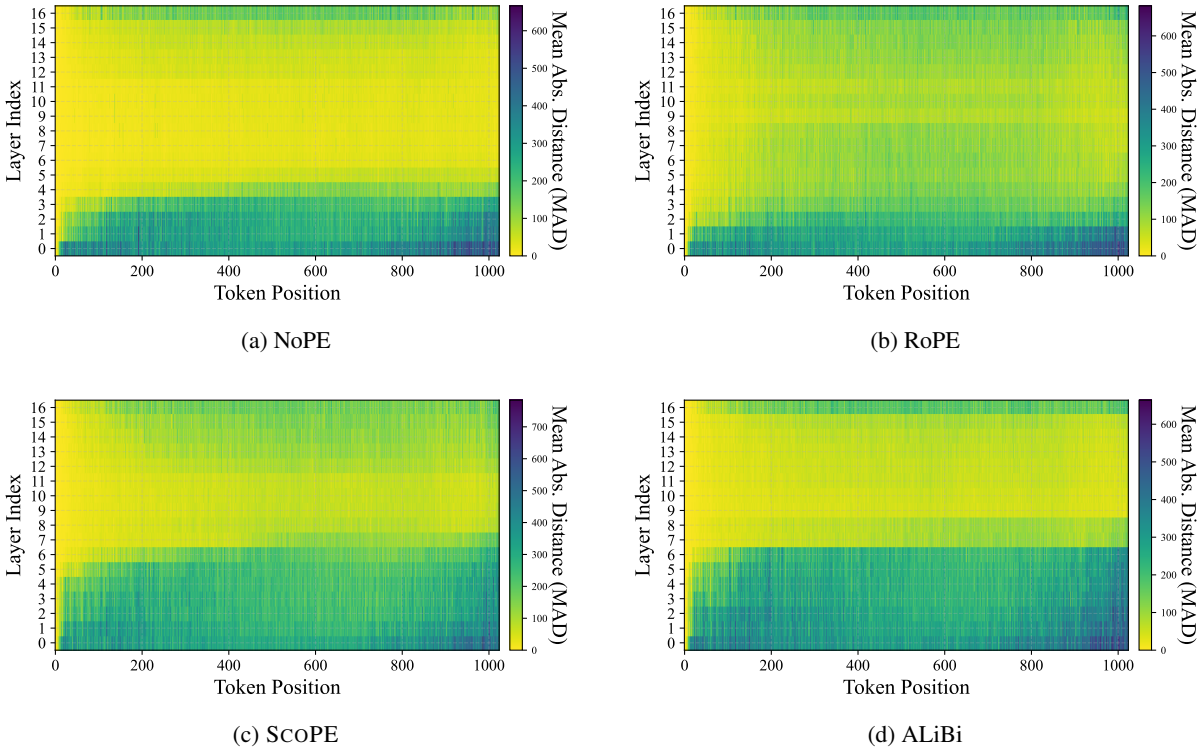
Our analysis suggests a fundamental link between SCOPE and ALiBi (Press et al., 2022). ALiBi introduces a linear bias $-m \cdot |t - i|$ to attention scores. We hypothesize that this bias functions as a *soft scope*: when the penalty term is sufficiently large, the effective attention weight becomes negligible, mathematically mimicking a hard mask.

Hypothesis Validation: Uniform-ALiBi. To test this, we trained a "Uniform-ALiBi" variant where all heads share an identical slope (using the smallest slope value intended for long-range dependencies). As shown in Figure 15, this variant fails to outperform NoPE, showing a nearly identical training loss curve. This implies that ALiBi's effectiveness—much like SCOPE—is driven by the *hierarchical diversity* of effective receptive fields across heads, rather than the bias term itself.

Soft Decay vs. Hard Hierarchy. Despite this connection, a key distinction remains: SCOPE enforces a *hard*, deterministic hierarchy via explicit masking, whereas ALiBi relies on *soft*, data-dependent decay. Because ALiBi's effective scope fluctuates with the magnitude of attention logits (which evolve during training), it may introduce unnecessary noise. SCOPE eliminates this ambi-

Table 3: Configuration of Qwen-Nano. We adopt a standard configuration for small-scale language modeling experiments to ensure reproducibility.

Model Hyperparameter	**Value**	**Training Hyperparameter**	**Value**
Layers (L)	16	Training Seq. Length	1024
Hidden Dim (d)	768	Batch Size	96
Heads (H)	16	Learning Rate	$3e^{-4}$
KV Heads (H_{kv})	16	Training Steps	40000
Head Dim (d_h)	64	Optimizer	AdamW
FFN Dim	3072	Betas	$(0.9, 0.999)$
Params	212M	Precision	BFloat16



(a) NoPE (b) RoPE

(c) SCOPE (d) ALiBi

Figure 13: Position reconstruction heatmaps. Brighter colors indicate lower error. SCOPE (c) achieves similar position resolution comparable to ALiBi (d) but retains the representational characteristics similar to RoPE (b).

guity by structurally defining the information flow, ensuring a stable hierarchy.

C Experimental Details

In this section, we provide comprehensive configurations for our experiments. Table 4 details the data composition across different training stages, ensuring a balance between domain diversity and long-context modeling. Table 5 lists the specific model hyperparameters and progressive training settings used for the LLaMA-3-8B experiments.

C.1 Experimental Setup

C.2 Additional Training Loss Curve

The superior convergence of SCOPE is not limited to the pre-training phase. As shown in Figure 16, our method maintains a consistent loss advantage over RoPE during both the 32k Long-Context Fine-tuning and the 128k Ultra-Long Adaptation stages.

C.3 Benchmark Configurations

We evaluate the models on a suite of standard zero-shot and few-shot benchmarks. The detailed configuration (shots and metrics) follows standard practices: GPQA (0-shot) (Rein et al., 2024), BBH (3-shot) (Suzgun et al., 2022), HellaSwag (10-shot) (Zellers et al., 2019), and others listed in Table 6.

C.4 Training Time comparision

Beyond inference, SCOPE significantly accelerates training. As detailed in Figure 17, the average training time per step at 32k context is reduced from 232s (RoPE) to 163s (SCOPE). This advan-

Table 4: Data Mixtures across Training Stages. We increase the proportion of long-context data during fine-tuning to encourage long-range dependency modeling.

Data Source	Domain	Pre-training	LCFT (32k) & (128k)
RefinedWeb (Penedo et al., 2023)	English	70%	30%
mC4 (Raffel et al., 2020)	Chinese	20%	20%
StarCoder (Li et al., 2023) / The Stack (Kocetkov et al., 2022)	Code	10%	10%
SlimPajama (Soboleva et al., 2023) ($L > 16k$)	Long Context	–	**40%**

Table 5: Configuration of LLaMA-3-8B Experiments. We detail the model architecture and the progressive training hyperparameters across the three stages: Pre-training (PT), Long-Context Fine-tuning (LCFT), and Ultra-Long Adaptation (Adapt).

Model Hyperparameter	Value	Training Hyperparameter	Value
Architecture	LLaMA-3	**Context Window**	
Layers (L)	32	*Stage 1: Pre-training*	4,096
Hidden Dim (d)	4096	*Stage 2: LCFT*	32,768
Heads (H)	32	*Stage 3: Adaptation*	131,072
KV Heads (H_{kv})	8 (GQA)	**Training Duration**	
Head Dim (d_h)	128	*Stage 1: Pre-training*	50B Tokens
FFN Multiplier	1.3	*Stage 2: LCFT*	2B Tokens
Vocab Size	128,256	*Stage 3: Adaptation*	50 Steps
RoPE θ (Base)	10,000	**Optimization**	
Norm	RMSNorm	Optimizer	AdamW
Precision	BFloat16	Learning Rate (Max)	$3e^{-4}$, $3e^{-5}$, $1e^{-5}$
Activation	SwiGLU	Weight Decay	0.1
Params	$\approx$ 8B	Global Batch Size	500K, 2M, 4M

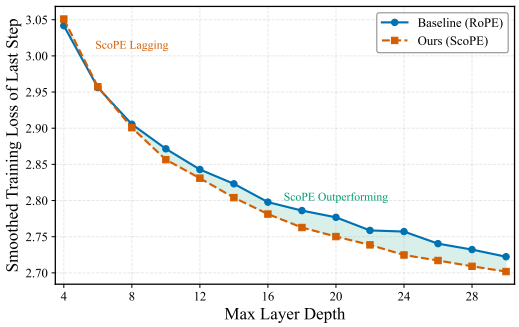


Figure 14: Performance scaling with model depth. In shallow networks (e.g., 2 or 4 layers), RoPE outperforms SCOPE. However, as the number of layers increases, enabling a deeper cascade, SCOPE surpasses RoPE.

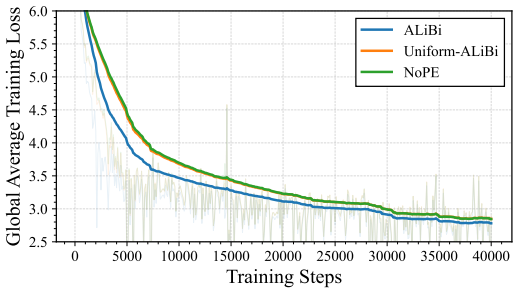


Figure 15: Training loss comparison. Standard ALiBi (geometric slopes) performs well, whereas "Uniform-ALiBi" (identical slopes) degrades to the performance of NoPE.

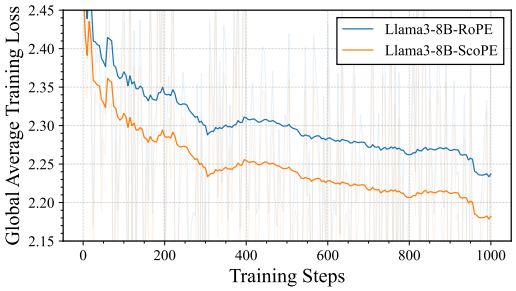
tage widens dramatically at 128k context, where SCOPE completes a step in 1180s compared to 2223s for RoPE—a near **2**$\times$ throughput increase.

C.5 Additional NIAH Results

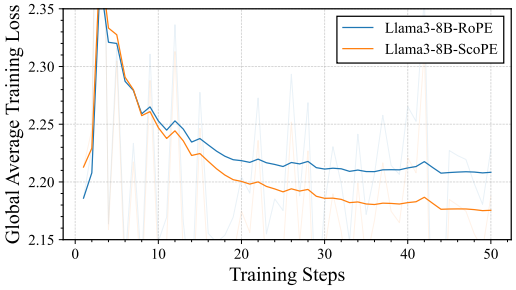
We further report NIAH results for baseline long-context fine-tuning on Llama3-8B. Specifically, Llama3-8B NoPE (FT-128k) achieves an average accuracy of 0.33, and Llama3-8B RoPE-SWA (FT-128k) achieves 0.34. As shown in Figure 18, both baselines mainly succeed when the needle appears close to the end of the prompt, and their retrieval accuracy drops sharply as the needle is placed deeper in the context, indicating limited effective long-range retrieval despite being fine-tuned with a 128k context window. In particular, the strong locality of RoPE-SWA is expected due to its 8k sliding window, which limits direct access to earlier tokens even when training uses a 128k context length.

D Supplementary Experiments of Qwen3

To verify the architectural universality of our method, we conduct additional experiments using the Qwen3-2B architecture (Yang et al., 2025). For a fair comparison, we train three variants from



(a) Loss curve for 32k context parallel training



(b) Loss curve for 128k context parallel training

Figure 16: Training loss comparison on LLaMA-3-8B (Fine-tuning Stage). SCOPE (Orange) consistently achieves lower training loss compared to the RoPE baseline (Blue) throughout the training run.

Table 6: Configuration of General NLU Benchmarks. We report the number of few-shot examples used for each task during evaluation.

Benchmark	Shots	Metric
GPQA (Rein et al., 2024)	0-shot	Accuracy
PIQA (Bisk et al., 2020)	0-shot	Accuracy
TruthfulQA (Lin et al., 2022)	0-shot	Accuracy
BoolQ (Clark et al., 2019)	0-shot	Accuracy
Lambada (Paperno et al., 2016)	0-shot	Accuracy
OpenBookQA (Mihaylov et al., 2018)	0-shot	Accuracy
BBH (Suzgun et al., 2022)	3-shot	Exact Match
WinoGrande (Sakaguchi et al., 2019)	5-shot	Accuracy
HellaSwag (Zellers et al., 2019)	10-shot	Accuracy
ARC-Challenge (Clark et al., 2018)	25-shot	Accuracy

scratch: NoPE, RoPE (Su et al., 2024), and SCOPE. To standardize the vocabulary across experiments, all 2B models utilize the LLaMA tokenizer with a vocabulary size of 32k.

Model Configuration. To accelerate training and strictly isolate the impact of position encoding, we standardize the vocabulary across all models using the LLaMA tokenizer (32k vocabulary size), deviating from the standard Qwen vocabulary. All models follow the Qwen3-2B specification: 32 layers, a hidden dimension of 2048, 32 attention heads, and a head dimension of 128. Training is conducted on the *RefinedWeb* (Penedo et al., 2023) dataset for 100,000 steps with a global batch size of 96 and a sequence length of 4,096 tokens.

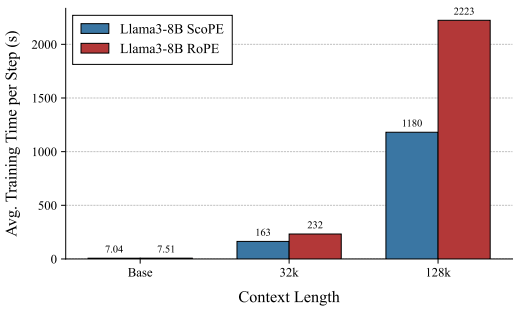


Figure 17: Training efficiency on Llama-3-8B. Average training time per step (in seconds) drastically reduces with SCOPE, dropping from 2223s to 1180s at 128k context.

Implementation Details. We utilize the `Torchtitan` framework (v0.2.1) for distributed training. While we incorporated a community patch (PR #1857)[3] to enable Sequence Parallelism for our LLaMA experiments, this modification proved incompatible with the Qwen3 architecture. Consequently, our Qwen3 experiments are restricted to the pre-training stage with a 4k context window, without the subsequent long-context fine-tuning stages employed in the main experiments.

D.1 Training Loss Curve

D.2 Perplexity

Figure 20 reports the zero-shot perplexity on long documents. SCOPE exhibits superior stability when extrapolating to unseen lengths, confirming that the hierarchical scope mechanism generalizes effectively even in smaller architectures.

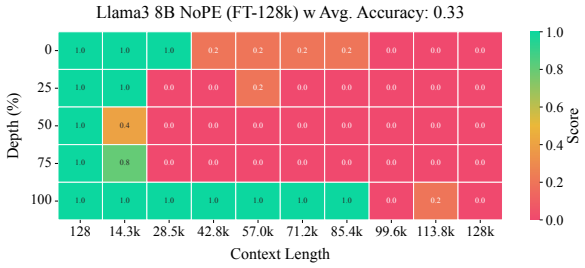
D.3 NIAH

To assess zero-shot context extension, we evaluate the 4k-trained models on an 8k context window ($2\times$ training length) using the NIAH task.

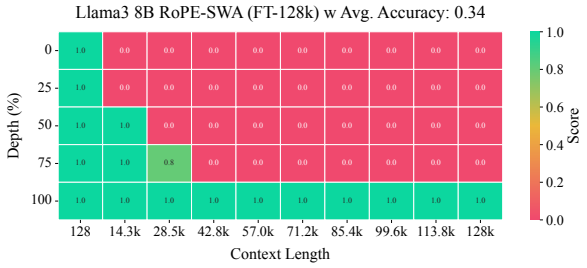
E Code Implementation

We provide the core implementation details for SCOPE using the `FlexAttention` API. The mask generation logic is concise and efficient, as illustrated in Figure 22. Furthermore, Figure 23 compares the implementation of NoPE, ALiBi, and SCOPE via the `score_mod` interface.

[3] https://github.com/pytorch/torchtitan/pull/1857



(a) Llama3-8B NoPE fine-tuned with 128k context.



(b) Llama3-8B RoPE-SWA fine-tuned with 128k context.

Figure 18: Additional Needle-in-a-Haystack (NIAH) results on Llama3-8B. Both baselines exhibit strong locality: they can reliably retrieve needles placed near the end of the context, but fail for most deeper insertions.

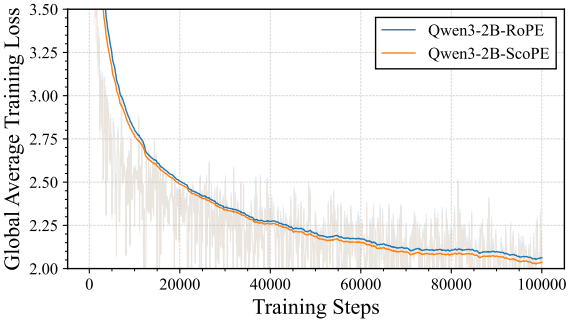


Figure 19: Training loss curve of Qwen3-2B. Consistent with our LLaMA-3 findings, SCOPE (Orange) exhibits superior convergence compared to both RoPE (Blue) and NoPE (Green) baselines.

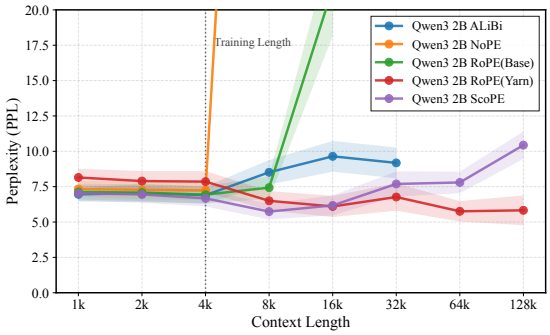
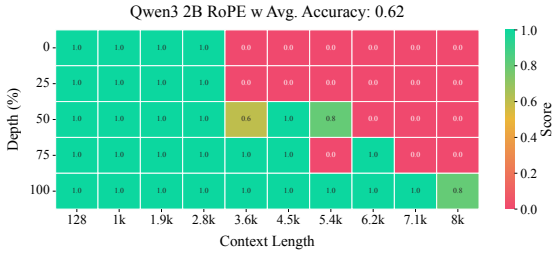
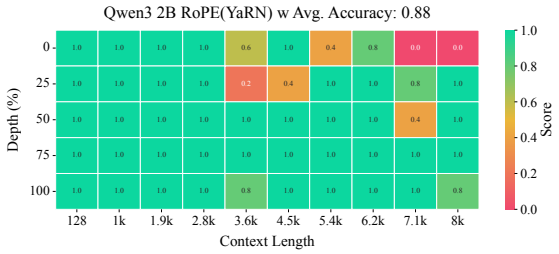


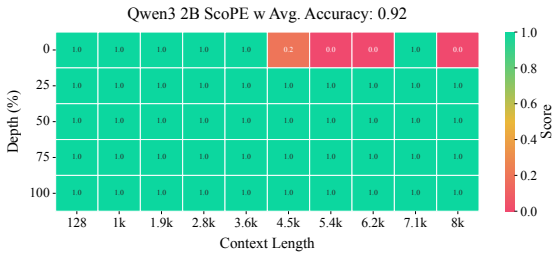
Figure 20: Perplexity extrapolation on Qwen3-2B. SCOPE demonstrates robust length extrapolation capabilities, maintaining lower perplexity on long documents compared to RoPE, mirroring the trends observed in the 8B experiments.



(a) RoPE Base (Acc: 0.62)



(b) RoPE + YaRN (Acc: 0.88)



(c) SCOPE (Acc: 0.92)

Figure 21: Needle-in-a-Haystack (NIAH) Heatmaps on Qwen3-2B. Models trained on 4k context are tested at 8k length. (a) RoPE fails to generalize (0.62 accuracy). (b) While YaRN improves RoPE (0.88 accuracy), it still shows degradation. (c) SCOPE achieves the highest performance (0.92 accuracy).

```python
from torch.nn.attention.flex_attention import flex_attention

S = [T**(i/H) for i in range(1, H+1)]
def scope_mask_mod(b, h, q_idx, kv_idx):
    return (q_idx - kv_idx <= S[h]) & (q_idx >= kv_idx)

flex_attention(query, key, value, block_mask=scope_mask_mod).sum().backward()
```

Figure 22: Pseudo-code for SCOPE Masking in FlexAttention. The logical mask avoids materializing the full $T \times T$ matrix, ensuring memory efficiency.

```python
from torch.nn.attention.flex_attention import flex_attention

scopes = generate_scopes() # [num_heads]
alibi_bias = generate_alibi_bias() # [num_heads]

def nope_score_mod(
    score: torch.Tensor,
    b: torch.Tensor,
    h: torch.Tensor,
    q_idx: torch.Tensor,
    kv_idx: torch.Tensor,
):
    return score

def alibi_score_mod(
    score: torch.Tensor,
    b: torch.Tensor,
    h: torch.Tensor,
    q_idx: torch.Tensor,
    kv_idx: torch.Tensor,
):
    alibi_bias = (kv_idx - q_idx) * alibi_bias[h]
    return score + alibi_bias.to(score.dtype)

def scope_score_mod(
    score: torch.Tensor,
    b: torch.Tensor,
    h: torch.Tensor,
    q_idx: torch.Tensor,
    kv_idx: torch.Tensor,
):
    return torch.where((q_idx - kv_idx) <= scopes[h], score, -float("inf"))

flex_attention(query, key, value, score_mod=scope_score_mod).sum().backward()
```

Figure 23: Comparison of Score Mod Implementations. We contrast the implementation of NoPE, ALiBi, and SCOPE. Note that SCOPE employs a hard masking strategy (via `torch.where`), whereas ALiBi adds a soft bias.

```python
def generate_triangular_matrix(n, w_list, device):
    matrix = torch.eye(n, dtype=torch.float64, device=device)
    rows = torch.arange(n, dtype=torch.int64, device=device)
    for w in w_list:
        lows = torch.clamp(rows - w + 1, min=0)
        highs = rows
        ranges = highs - lows + 1
        random_floats = torch.rand(n, dtype=torch.float32, device=device)
        random_offsets = (random_floats * ranges).long()
        matrix[rows, lows + random_offsets] += 1
    return matrix
```

Figure 24: Pseudo-code for constructing the random causal transition matrix used in the top-1-selection simulation. Each row starts from the residual identity and adds one randomly selected predecessor from each visible attention window.