SCOPE: Boosting LLM Efficiency with Scoped Position Encoding

Anonymous ACL submission

Abstract

Positional encodings are fundamental to Transformers, yet explicit methods like RoPE often incur high computational overhead and struggle with length extrapolation. 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 distributed look-back scopes to attention heads. We theoretically demonstrate that this simple topological constraint transforms the model into a hierarchical processor, inducing exponential Order Awareness (OA) with network depth. Consequently, SCOPE is parameter-free and avoids the resolution decay typical of explicit methods. Empirically, it significantly enhances efficiency by masking the majority of attention computations—offering a theoretical $8\times$ reduction in FLOPs. 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.

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 extensive contexts remains a formidable architectural challenge.

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 and memory 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).

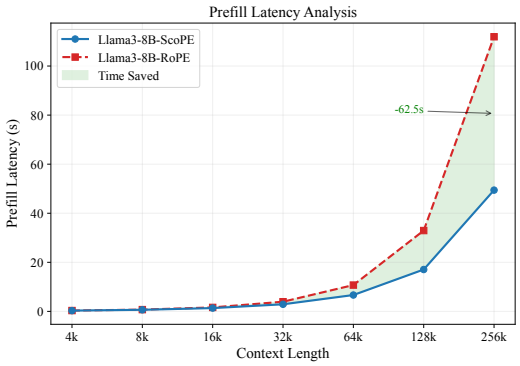


Figure 1: **Latency reduction analysis.** Comparison of absolute prefill latency between RoPE and SCOPE. The shaded green area highlights the substantial time savings achieved by our approach. At the maximum length of 256k, SCOPE reduces the wait time by approximately **62.5 seconds**.

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 often 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 an unresolved tension in the field.

In this work, we propose a paradigm shift by revisiting this challenge through the lens of **Position Encoding (PE)**. Traditionally, PE is treated as an explicit, arithmetic signal—injected via embeddings or rotational modulations (RoPE) (Su et al., 2024) or attention biases (ALiBi) (Press et al.,

2022). However, these explicit methods face two critical limitations in long-context scenarios: (1) they incur additional computational overhead; and (2) they suffer from numerical instability 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 resolution in shallow layers, failing to match the precision 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 is a potent position encoder*. Our core innovation lies in configuring attention heads with **exponentially distributed receptive 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 resolution capacity of Transformer compounds exponentially, transforming a cascade of sparse layers into a precise sequence processor capable of resolving global order 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 computational complexity while achieving superior length generalization and retrieval precision (surpassing explicit PEs). 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 distributed scopes. We provide a theoretical guarantee that this hierarchical structure achieves exponential resolution growth with network depth, eliminating the need for explicit arithmetic position encodings.
- **Elastic Structure & Efficiency:** We demonstrate that SCOPE exhibits a hierarchical allocation of positional capacity, adapting naturally to varying lengths, unlike RoPE. This design reduces attention computation by up to $8\times$ (theoretically) and achieves a $2\times$ measured speedup at 128k context.
- **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 routing 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, the attention weights depend solely on content ($\mathbf{q}^\top\mathbf{k}$), rendering it a permutation-invariant "bag-of-words" model incapable of resolving sequential order (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). A model f possesses OA of length T, denoted OA(T), if for any input $\mathbf{X} \in \mathbb{R}^{T \times d}$ and any non-identity permutation π, $f(\mathbf{X}) \neq f(\pi(\mathbf{X}))$.

This implies that distinct token orderings yield distinct representations. We can then derive a straightforward corollary.

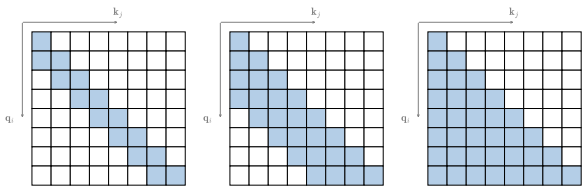
Corollary 2.1 (Monotonicity). *OA(T) $\implies$ OA(t) for all $1 \leq t < T$, as shorter sequences can be viewed as suffixes of fixed contexts.*

This property is fundamental to understanding how our proposed SCOPE leverages hierarchical composition to achieve exponential growth in sequential perception.

3 Methodology

In this section, we detail SCOPE, a streamlined mechanism designed to instill OA into Transformers by imposing structured sparsity. By assigning exponentially distributed 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 2: **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* $S_h \in \mathbb{N}^+$. We implement via a modified head-specific mask $\mathcal{M}^{(h)}$ based on Equation 1:

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

To maximize coverage, we assign scopes exponentially across the H heads in each layer. Specifically, letting $S_0 = 1$, we ensure the scopes for heads $h \in \{1, \ldots, H\}$ satisfy the geometric progression:

$$S_{h-1} < S_h \leq 2S_{h-1}. \tag{3}$$

In practice, we set:

$$S_h = \lfloor \gamma^h \rfloor, \quad \text{where } 1 < \gamma \leq 2. \tag{4}$$

This distribution ensures that heads collectively cover the context history efficiently.

3.2 Scoped Position Encoding

We now demonstrate that stacking layers with these exponentially distributed scopes leads to an exponential growth in OA capability, effectively serving as an implicit position embedding.

Theoretical Setup: While scopes are distributed across heads h, the *effective resolution* of the model grows with depth l. Let the *effective resolution* of the model at layer l be denoted as S_l. Ideally, S_l expands exponentially with depth, i.e., $S_l \approx \gamma^l$. We define the embedding layer as $l = 0$ with scope $S_0 = 1$.

Theorem 3.1 (Exponential Order Awareness)**.** *Consider an L-layer Transformer with scoped attention. If S_l grows exponentially by Equation 4, and assuming the FFN and Attention are injective functions, then the L-layer model achieves Order Awareness of range $S_L = \gamma^L$ (denoted as OA(S_L)).*

A rigorous proof is provided in Appendix A.

To provide intuition for Theorem 3.1, Figure 3 visualizes this mechanism as a *recursive reduction*.

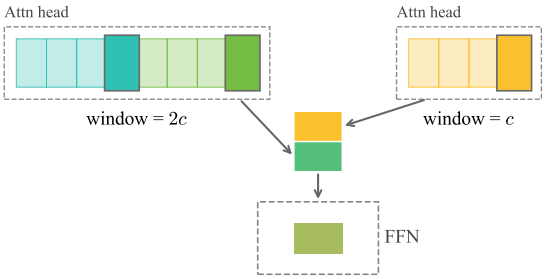


Figure 3: **Visual intuition of Theorem 3.1: Doubling the Order Awareness.** Assuming the input already possesses OA(c) (represented by unique colors for unique contexts), the representation of the last token of a length-c sequence acts as a unique signature. Consequently, the complex problem of resolving a length-$2c$ sequence reduces to the simpler problem of identifying the relative order of *two* such signatures. This is structurally equivalent to resolving a length-2 sequence using heads with varying scopes.

The core logic proceeds as follows: If the previous layer can achieve OA(c), the unique representation of a length-c sequence acts as a latent *signature*. Consequently, resolving the order of a length-$2c$ sequence reduces to determining the relative order of *two* such signatures. This scenario is structurally isomorphic to the base case of resolving a length-2 sequence—distinguishing (x_1, x_2) from (x_2, x_1)—which is solvable by contrasting a short-scope head (observing only x_2) against a long-scope head (observing both x_1 and x_2). By stacking layers, SCOPE recursively applies this logic, doubling its resolving power at each step.

3.3 Properties of SCOPE

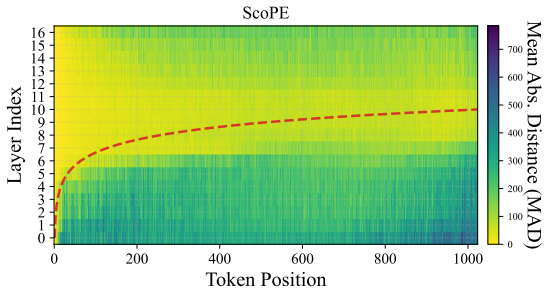
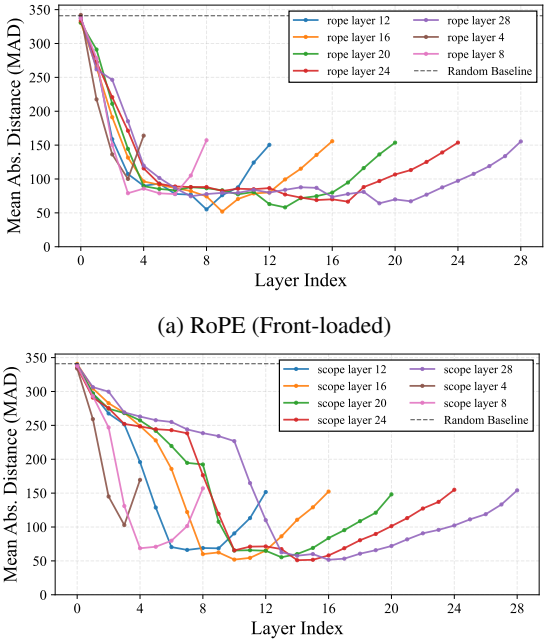


Figure 4: **Position reconstruction error map for SCOPE.** Brighter areas indicate lower error (higher accuracy). The superimposed red dotted line follows a logarithmic curve, empirically validating that the model's capacity to resolve sequence order grows exponentially with depth.

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.



(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.

Adaptive Layer Allocation. We further investigate how position encoding behavior scales with 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. We leverage the *FlexAttention* framework (He et al., 2024) for efficient training. As shown in Figure 19 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 $\sum S_h$. The reduction ratio approximates:

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

For $\gamma = 2$ and $H = 32$, this yields a theoretical $\mathbf{8}\times$ 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 (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. Our primary baseline is RoPE (Su et al., 2024), the de facto standard for modern LLMs. For fair comparison in long-context stages, we employ YaRN (Peng et al., 2024) as the extrapolation method for RoPE. For SCOPE, we adhere to a simple scaling strategy by aligning the maximum scope S_{max} with the current sequence length T, without employing additional complex extrapolation tricks.

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). Source code will be made publicly available upon acceptance. Detailed configurations for hyperparameters, training protocols, and data mixtures are provided in Appendix C.

4.2 Training Dynamics and Efficiency

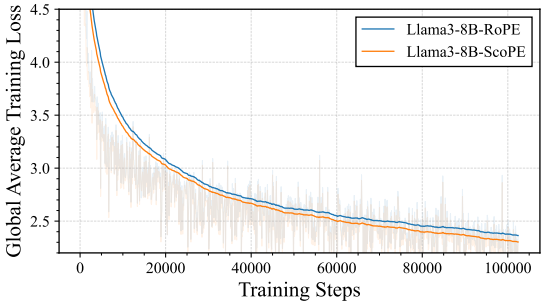


Figure 6: **Training loss comparison on LLaMA-3-8B (Pre-training Stage).** SCOPE (Orange) consistently achieves lower training loss compared to the RoPE baseline (Blue) throughout the training run. This indicates that the structured sparsity imposed by SCOPE serves as an effective inductive bias, facilitating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

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 14 for 32k and 128k loss curves), indicating that our method provides a more efficient optimization landscape.

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.

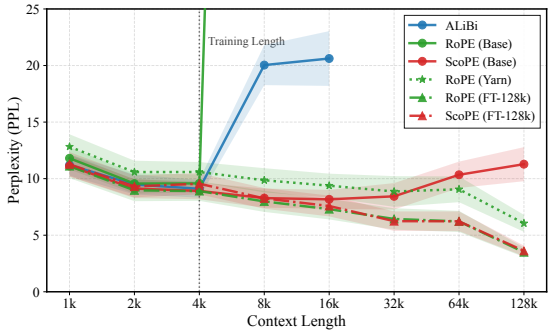


Figure 7: **Perplexity extrapolation on long documents (up to 128k).** We compare: (1) *Base*: Native 4k-trained models; (2) *Fine-tuned*: Models adapted to 128k. SCOPE exhibits superior native extrapolation (up to 64k) compared to RoPE.

In the *Base* regime (trained on 4k), SCOPE demonstrates exceptional robustness. While RoPE and ALiBi fail catastrophically immediately beyond the training window, SCOPE maintains stability up to 16k tokens (4× training length) without any parameter updates. Furthermore, even within the standard 32k window, the base SCOPE model yields consistently lower perplexity than the untuned RoPE baseline (even when aided by YaRN extrapolation). 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.

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.

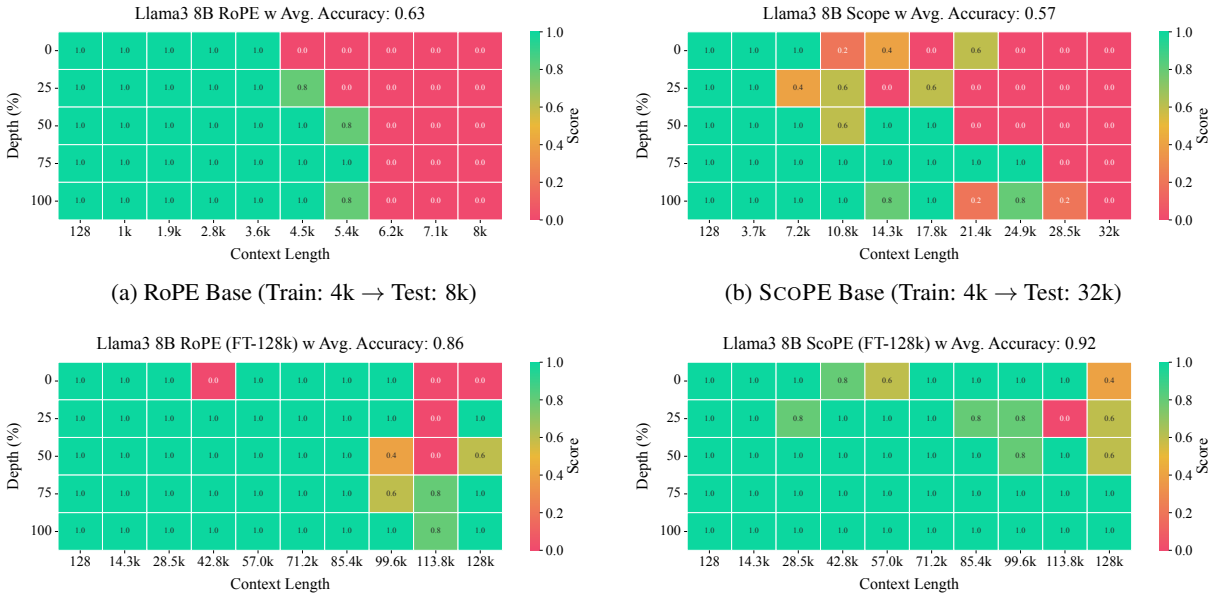
General NLU Benchmarks. We evaluate the models on a suite of standard zero-shot and few-shot benchmarks utilizing the `lm-evaluation-harness` library (Gao et al., 2024). For a detailed breakdown of the evaluation protocol, including the specific number of few-shot examples and metric specifications for 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 fine-tuning, 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



(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. The X-axis represents context length, and the Y-axis represents needle depth.

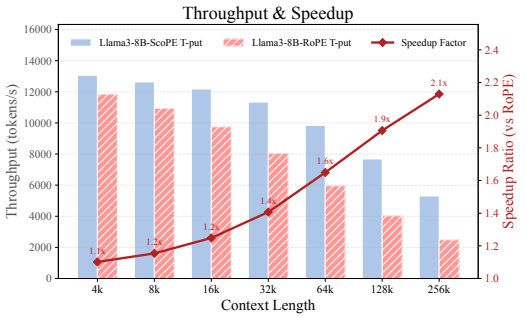


Figure 9: **Prefill performance on Llama-3-8B.** Comparison of throughput (bars) and speedup ratio (line) demonstrates that SCOPE effectively handles long-context inference. At 256k tokens, we achieve a **2.1**× speedup over the RoPE baseline.

128k context, we observe a 2× speedup, expanding to 2.1× at 256k. We also provide a training time comparison in Appendix C.4.

This empirical evidence validates our theoretical complexity analysis, confirming that SCOPE effectively mitigates the quadratic cost of attention, making it highly suitable for ultra-long context applications.

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, these explicit methods rely on arithmetic operations that incur computational overhead and potential precision instability in 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

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**

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**

potential integration with advanced extrapolation tricks for future enhancements.

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 sharing the property of sparsity, SCOPE fundamentally utilizes it for *positional encoding* rather than solely for computational reduction. The closest parallel is Mistral's Sliding Window Attention (SWA) (Jiang et al., 2023; Xiao et al., 2024). Crucially, however, Mistral applies a *uniform* window size, limiting the effective receptive field. In contrast, SCOPE employs *geometrically distributed* 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 distributed 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 exponential Order Awareness (OA) 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 pretraining, exhibits remarkable native length extrapolation (up to $4\times$ the training context), 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 proof of Order Awareness (Theorem 3.1) assumes that FFN and Attention layers function as injective mappings. In practice, the finite hidden dimension may act as an information bottleneck relative to input complexity. Consequently, strictly resolving all positional ambiguities might be challenging in high-entropy scenarios due to potential information loss.

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.

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.

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.

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 Proof of 3.1

Proof. We proceed by induction on depth l.

Base Case ($l = 0$): $S_0 = 1$. A sequence of length 1 has no non-identity permutations. Holds trivially.

Inductive Step: Assume the first $(l-1)$ layers achieve OA(S_{l-1}). We show that layer l achieves OA(S_l) where $S_l \leq 2S_{l-1}$. We partition a sequence of length S_l into a *suffix* of length S_{l-1} (the most recent tokens) and a *prefix* of length $S_l - S_{l-1}$. Any non-identity permutation must alter the suffix or the prefix:

- **Change in Suffix:** By the inductive hypothesis, the input to layer l (output of $l-1$) already distinguishes these permutations. The FFN preserves this distinction.
- **Change in Prefix:** The prefix length is $\leq S_{l-1}$, so previous layers distinguish its local order. Crucially, layer l has a scope S_l that covers this prefix. The attention mechanism effectively "sees" the distinct representation of the prefix (aggregated from previous layers) relative to the suffix. Since the representation of the prefix is unique, the attention output changes.

Therefore, layer l is capable of resolving permutations over the entire length S_l. By induction, an L-layer model achieves OA(S_L) where S_L grows exponentially. □

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

Layer-wise Functional Transition. Figure 10 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.

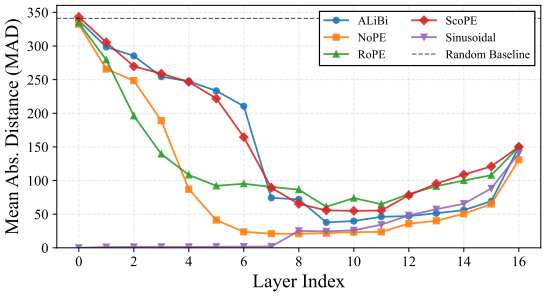


Figure 10: **Layer-wise probing error across different PEs.** SCOPE exhibits a learning trajectory most similar to ALiBi, characterized by a smooth, progressive resolution of positions in early layers.

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.

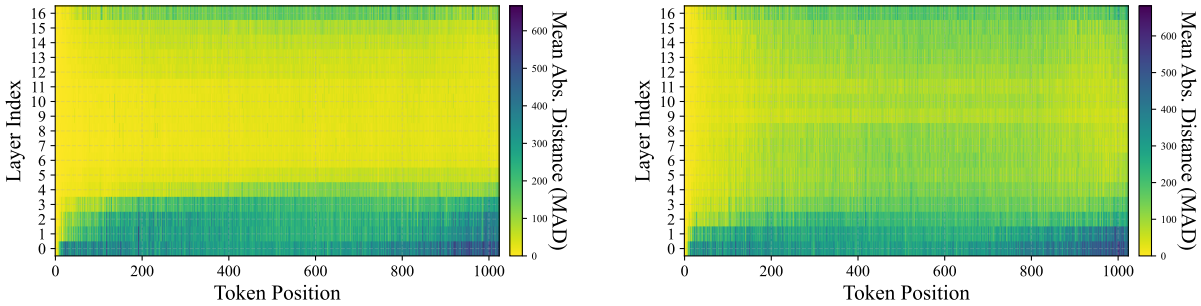
Visualizing Representation. Figure 11 provides a granular visualization. SCOPE effectively synthesizes the strengths of existing methods: it mirrors the structural clarity of ALiBi in early layers while maintaining the robust representational capacity characteristic of RoPE in deeper layers.

B.3 Synergy with Causal Masking

Our theoretical analysis (Theorem 3.1) establishes a sufficiency 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).

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 11: **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).

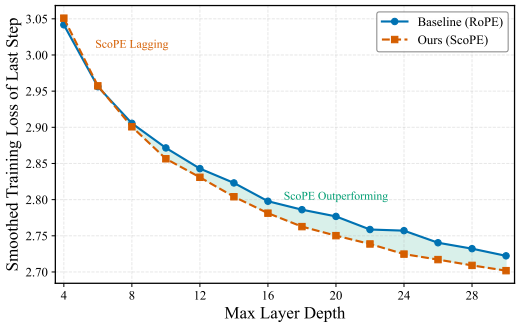


Figure 12: **Performance scaling with model depth.** In shallow networks (e.g., 2 or 4 layers), RoPE outperforms SCOPE. However, as the number of layers increases, enabling a deeper cascade, SCOPE surpasses RoPE.

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 12, 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 approaches zero, 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 13, 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.

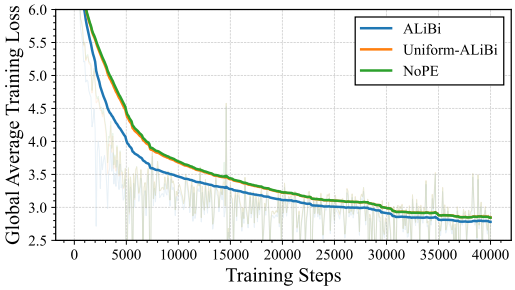


Figure 13: **Training loss comparison.** Standard ALiBi (geometric slopes) performs well, whereas "Uniform-ALiBi" (identical slopes) degrades to the performance of NoPE. This confirms that hierarchical diversity is the primary driver of performance.

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

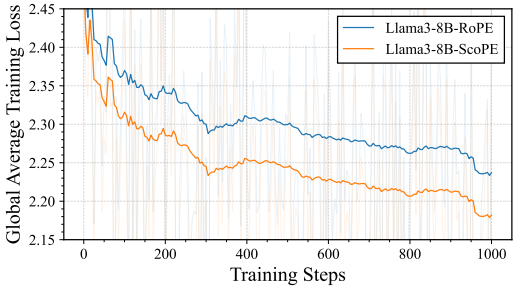
C Experimental Details

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.

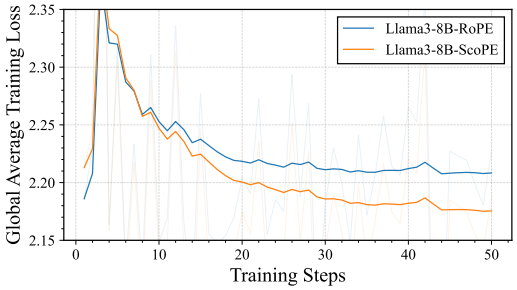
C.1 Experimental Setup

C.2 Additional Training Loss Curve

The superior convergence of SCOPE is not limited to the pre-training phase. As shown in Figure 14, our method maintains a consistent loss advantage over RoPE during both the 32k Long-Context Fine-tuning and the 128k Ultra-Long Adaptation stages.



(a) Loss curve for 32k context parallel training



(b) Loss curve for 128k context parallel training

Figure 14: 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.

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 15, the average training time per step at 32k context is reduced from 232s (RoPE) to 163s (SCOPE). This advantage widens dramatically at 128k context, where SCOPE completes a step in 1180s compared to 2223s for RoPE—a near $\mathbf{2}\times$ throughput increase.

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

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

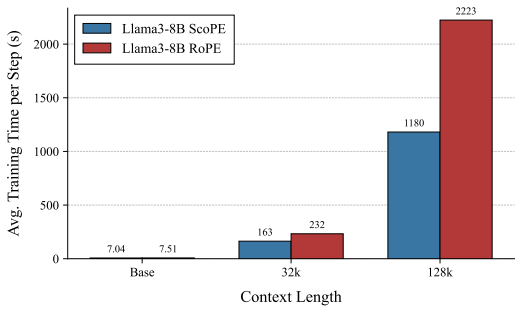


Figure 15: **Training efficiency on Llama-3-8B.** Average training time per step (in seconds) drastically reduces with SCOPE, dropping from 2223s to 1180s at 128k context.

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 four variants from 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.

Implementation Details. We utilize the `Torchtitan` framework (v0.2.1) for distributed training. While we incorporated a community patch to enable Sequence Parallelism for our LLaMA experiments, this modification proved incompatible with the Qwen3 architecture. Conse-

quently, 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

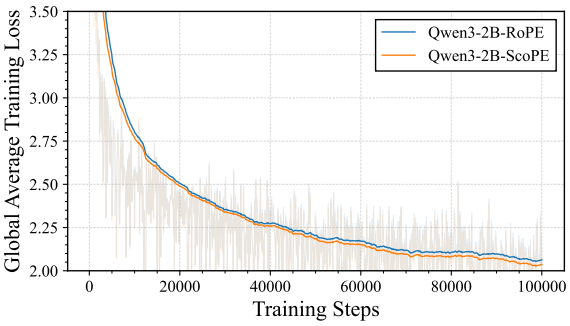


Figure 16: **Training loss curve of Qwen3-2B.** Consistent with our LLaMA-3 findings, SCOPE (Orange) exhibits superior convergence compared to both RoPE (Blue) and NoPE (Green) baselines.

D.2 Perplexity

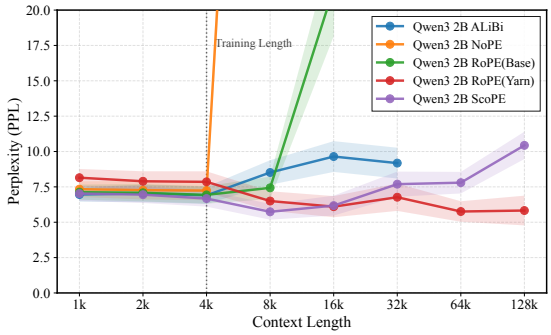
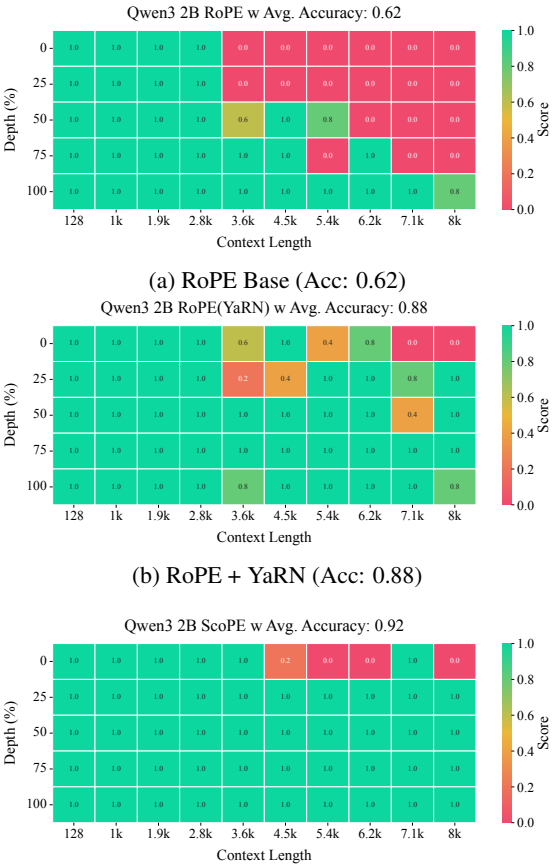


Figure 17: **Perplexity extrapolation on Qwen3-2B.** SCOPE demonstrates robust length extrapolation capabilities, maintaining lower perplexity on long documents compared to RoPE, mirroring the trends observed in the 8B experiments.

Figure 17 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× training length) using the NIAH task.



(a) RoPE Base (Acc: 0.62)

(b) RoPE + YaRN (Acc: 0.88)

(c) SCOPE (Acc: **0.92**)

Figure 18: **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**) natively, without additional extrapolation tricks.

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 19. Furthermore, Figure 20 compares the implementation of NoPE, ALiBi, and SCOPE via the `score_mod` interface.

```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 19: **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 20: **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.