Rope to Nope and Back Again: A New Hybrid Attention Strategy

Bowen Yang[1] **Bharat Venkitesh**[1] **Dwarak Talupuru**[1]
Hangyu Lin[1] **David Cairuz**[1] **Phil Blunsom**[1] **Acyr Locatelli**[1]

[1] Cohere

{bowen, dwarak, john, phil}@cohere.com {bharat, davidcairuz, acyr}@cohere.ai

Abstract

Long context large language models (LLMs) have achieved remarkable advancements, driven by techniques like Rotary Position Embedding (RoPE) [61] and its extensions [12, 47, 54]. By adjusting RoPE parameters and incorporating training data with extended contexts, we can train performant models with considerably longer input sequences. However, existing RoPE-based methods exhibit performance limitations when applied to extended context lengths. This paper presents a comprehensive analysis of various attention mechanisms, including RoPE, No Positional Embedding (NoPE), and Query-Key Normalization (QK-Norm), identifying their strengths and shortcomings in long context modeling. Our investigation identifies distinctive attention patterns in these methods and highlights their impact on long context performance, providing valuable insights for architectural design. Building on these findings, we propose a novel architecture featuring a hybrid attention mechanism that integrates global and local attention spans. This design not only surpasses conventional RoPE-based transformer models with full attention in both long and short context tasks but also delivers substantial efficiency gains during training and inference.

1 Introduction

Developing language models capable of handling long context lengths poses several challenges. First, as the context length increases, an effective modeling of extended input sequences becomes increasingly critical. This often requires advancements in positional encoding [61], extrapolation techniques [23], or architectural innovations [69, 34]. Second, training long context large language models with billions of parameters demands significant computational resources. Overcoming this challenge requires scalable algorithms, high-quality datasets, and robust infrastructure. Lastly, deploying these models in real-world applications demands low latency and low memory usage, which requires meticulous optimization of both the model architecture and the serving infrastructure.

On the modeling front, two components of the transformer architecture are particularly crucial for long context capabilities: the attention mechanism and positional embeddings. Recent research has proposed various methods to enhance these components. For instance, Landmark Attention [53] trains attention modules to select relevant blocks using a representative token, referred to as a “landmark token", for efficient retrieval within extended text corpora. Similarly, Focused Transformer [69] adopts a contrastive training approach to prioritize attending the most relevant portions of the input sequence, allowing the model to focus on smaller, contextually significant subsets of tokens. Although these approaches improve long context modeling, stabilizing training remains a key challenge for extending transformer capabilities to longer sequences. Query-Key Normalization (QK-Norm) [31, 57] has

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

been introduced to address the stability issue, which normalizes the query-key vectors along the head dimension before computing attention. Although QK-Norm mitigates numerical instability during training and is widely used [66, 21, 42], it may impair long context capabilities.

In addition to the chosen attention mechanism, positional embeddings play a crucial role in long context modeling. Various approaches have been proposed to improve their effectiveness. Popular methods include Absolute Position Embedding (APE) [70], Relative Position Embedding [56], ALiBi [55], and Rotary Position Embedding (RoPE) [61]. Among these, RoPE has gained significant adoption in large language models (LLMs) [24, 75, 83] due to its simplicity and effectiveness. In particular, it has the ability to extrapolate context lengths by adjusting RoPE θ values during training [45, 2, 18]. Other techniques, such as relative bias [55, 56, 13] and contextualized position embeddings [26], introduce distance-based bias terms or condition the position information on input semantics. These methods often affect attention distribution by incorporating auxiliary information, such a positional indices or explicit recency bias. However, whether certain information or biases are beneficial to long context modeling or overall performance remains less explored. Additionally, the concept of No Positional Embedding (NoPE) has been explored by [38], suggesting that removing explicit positional embeddings and relying solely on implicit positional information derived from the causal mask can enhance long context performance.

Despite advances in long context modeling, training and serving such models remain challenging due to the quadratic complexity of standard attention. Techniques like Sliding Window Attention (SWA) [36, 67] mitigate this by restricting each token's attention to a local window, reducing compute while maintaining local coherence. Sparse attention methods [14, 7, 64] extend this by introducing structured sparsity, including random [78] and dilated patterns [22]. More recent approaches further compress attention, such as activation beacon [80], which sequentially summarizes local keys and values, and attention sink [73], which stabilizes long-sequence training by preserving early tokens with a sliding window. On the serving side, KV cache trimming methods [81, 41, 10] selectively discard cached states based on heuristics to reduce memory usage and boost inference efficiency. However, these gains often come with trade-offs in model quality, emphasizing the need for careful design.

In this paper, we begin analyzing attention patterns of different attention mechanisms, RoPE, NoPE, and QK-Norm and its impacts on long context performance trained up to 750 billion tokens. Building on these insights, we propose a novel hybrid attention architecture and extensively pretrain up to 5 trillion tokens, followed by supervised fine-tuning on a diverse set of datasets tailored for long context. We show that this architecture surpasses existing state-of-the-art extrapolation-based RoPE models [47] by a large margin, striking a balance between efficiency and performance.

2 Observation

In this section, we assess three models with different attention components on needles-in-a-haystack [37] (NIAH) and analyze the attention patterns to understand how these variants affect performance. Analysis in this section guides our architectural design choices throughout this work.

2.1 Experimental Setup

All model variants share a common configuration consisting of 8 billion parameters (including the token embedding parameters), with detailed architectural specifications provided in Table 1. For this set of experiments, the model is trained in two stages: a pretraining stage followed by a supervised fine-tuning (SFT) stage. Previous research shows that the SFT stage is necessary for long context evaluations, as it can reduce variance in long context tasks and enables the emergence of long context capabilities that may not manifest in models trained solely through pretraining [25].

We pretrain the model with a batch size of 4 million tokens. We use AdamW with a peak learning rate of $7e^{-3}$, a linear warmup of 2000 steps and a cosine learning rate schedule decaying to $3.5e^{-4}$ over 179,000 steps for a total of 750 billion tokens. For the SFT stage, we adopt an interleaved training strategy: we combine short- and long context data in a 3:1 ratio, with context lengths of 8192 and 65536 tokens, respectively. We use a batch size of 0.5 million tokens.

The 3 model variants we test are:

Parameters	Emb. Dim	FFN Dim	Non-linearity	Layers	Heads	KV Heads	Vocab Size
Values	4096	28672	swiglu	32	32	8	256000

Table 1: Model architecture details

1. **RoPE Model:** For this variant, we employ Rotary Position Embedding (RoPE) to encode positional information. During the pretraining stage, the RoPE parameter θ is set to 10,000. In the subsequent supervised fine-tuning (SFT) stage, θ is increased to 2 million to account for the increased context length. This variant serves as the baseline model configuration, maintaining an architecture similar to that of most existing models [24, 36, 17].
2. **QK-Norm Model:** Layer normalization [4] is applied to both the query and the key vectors before performing the angular rotation used in RoPE. All other hyperparameters, including the θ values and training methodology, remain identical to those of the RoPE variant.
3. **NoPE Model:** Previous research [72, 29] has demonstrated that transformer variants trained without positional embeddings (NoPE) can perform effectively on long context tasks. However, these models often exhibit inferior performance in terms of perplexity and downstream task evaluations within the trained sequence length [29]. In our study, the NoPE variant does not have QK-Norm. This variant is trained using the same methodology as the other two variants.

2.2 Evaluation and Attention Analysis

2.2.1 Evaluations

We evaluate the variants on a set of core evaluation benchmarks, including MMLU [30], HellaSwag [79], CommonsenseQA [63], ARC [15] for core model capabilities and NIAH benchmark [37] for long context capability. NIAH evaluates a model's ability to retrieve information accurately from a specific sentence (the "needle") embedded within a lengthy document (the "haystack"). The needle is randomly placed at varying depths within the sequence to examine performance across different context lengths. To improve robustness, we modify the original NIAH benchmark, where each context-depth combination is tested 16 times with different random seeds, creating diverse context compositions for comprehensive evaluation. The results of all standard benchmark evaluations and results with 65536 context length needles are presented in Table 2. Although prior research has emphasized the limitations of NIAH [74] for evaluating deeper and more general context understanding, our focus is solely on testing basic long context capabilities and gaining insights on model architecture design, for which this benchmark is sufficient.

Table 2 shows that the RoPE and QK-Norm variants exhibit comparable performance on standard benchmarks, while the NoPE variant lags behind. This finding is consistent with previous studies [38, 72]. For long context evaluations, QK-Norm performs the worst among the three variants, despite its decent performance in other capabilities. This observation is consistent with the results from the comparisons between Command R and Command R+, where Command R, despite being a significantly smaller model, outperforms Command R+ overall on longer context benchmarks [32]. Although the NoPE variant has slightly lower needles score compared to the RoPE variant, it is decent given that its base capabilities is relatively low.

2.2.2 Attention Pattern Analysis

To better understand the impacts of different architectures, we also analyze the attention patterns within each model. This approach is inspired by previous studies [77, 39] where attention scores assigned to different parts of the context are closely examined.

We still utilize the NIAH task by first dividing the context into four segments:

- **Begin:** The first 10 tokens. This part of the context is also often referred to as the "attention sink" [73], where a disproportionately large amount of attention is typically allocated in a trained transformer model.

Model	Val Loss	MMLU	HellaSwag	CommonsenseQA	ARC-E	ARC-C	Needles 65k
RoPE	1.52	48.55	73.74	68.30	81.05	39.13	9.82
QK-Norm	1.53	48.21	73.68	68.23	80.54	38.98	7.93
NoPE	1.58	47.61	72.16	66.42	76.94	37.12	9.03

Table 2: Comparison of model performance across a range of benchmarks. All evaluations are based on the outputs of the SFT models. Red cells indicate lower performance.

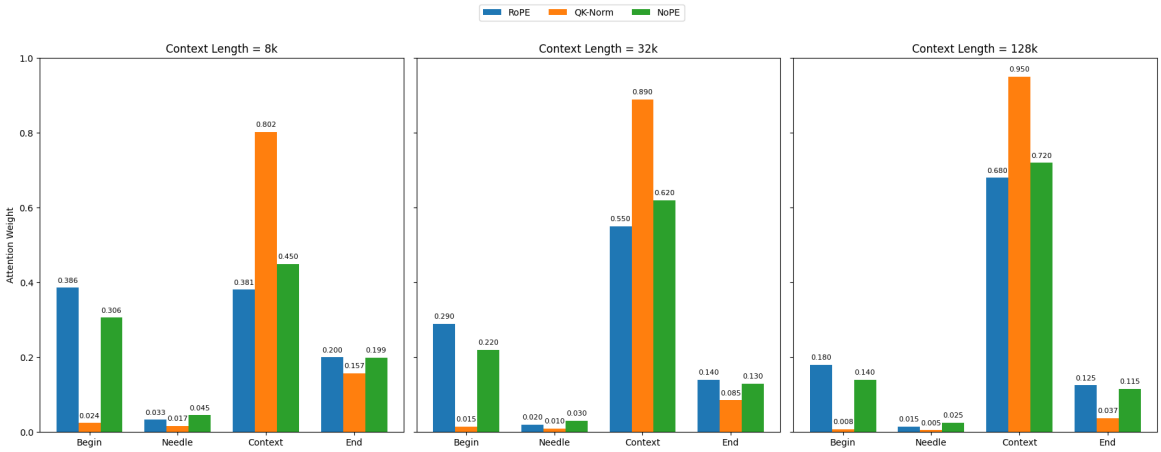


Figure 1: Attention Distribution across context lengths of each variant

- **Needle:** The tokens of the inserted needle sentence. Ideally, a well-trained model should assign a relatively high amount of attention to this part of the context.
- **End:** The query and completion tokens, which can represent recency bias.
- **Context:** The remainder of the context, typically consisting of noise or irrelevant content.

We position the needles at approximately 50% depth within the context to increase the complexity of the task, as most models suffer from the lost-in-the-middle problem, as highlighted in previous works [46, 5]. For each model, we first calculate the attention scores between the query tokens of "End" segment and the key tokens of all four segments across all heads and layers. The attention scores are summed within each segment and then aggregated across all heads and layers to obtain the average attention weight for each segment. These scores are further averaged across multiple samples at sequence lengths of 8,000 tokens, 32,000 tokens, and 128,000 tokens. We refer to this metric as "attention mass" in the following sections. The results of each variant are visualized in Figure 1.

We begin by comparing attention distributions across different sequence lengths from Figure 1. We observe a consistent decrease in attention mass allocated to the "Needle" segment and a corresponding increase in attention mass on the "Context" segment as sequence length increases. This trend indicates that retrieving relevant information becomes increasingly difficult as the context grows longer. When comparing across model variants, the NoPE variant consistently allocates the highest attention mass to the Needle" segment, followed by the RoPE variant, while the QK-Norm variant assigns the least attention to this segment. Furthermore, the QK-Norm variant exhibits markedly lower attention mass on the "Begin" segment and substantially higher attention mass on the "Context" segment relative to other variants. This indicates that models trained with the QK-Norm component exhibit a weak attention sink and are more susceptible to interference from noisy content. These patterns are consistent with QK-Norm's poor performance on the NIAH task. We argue that QK-Norm has this effect because the normalization operation mitigates magnitude information from the dot product of Query and Key vectors which tends to result in attention logits being closer in terms of magnitude and flatter in terms of distribution. A more detailed analysis of why QK-Norm is detrimental to long-context modeling is provided in Appendix B.

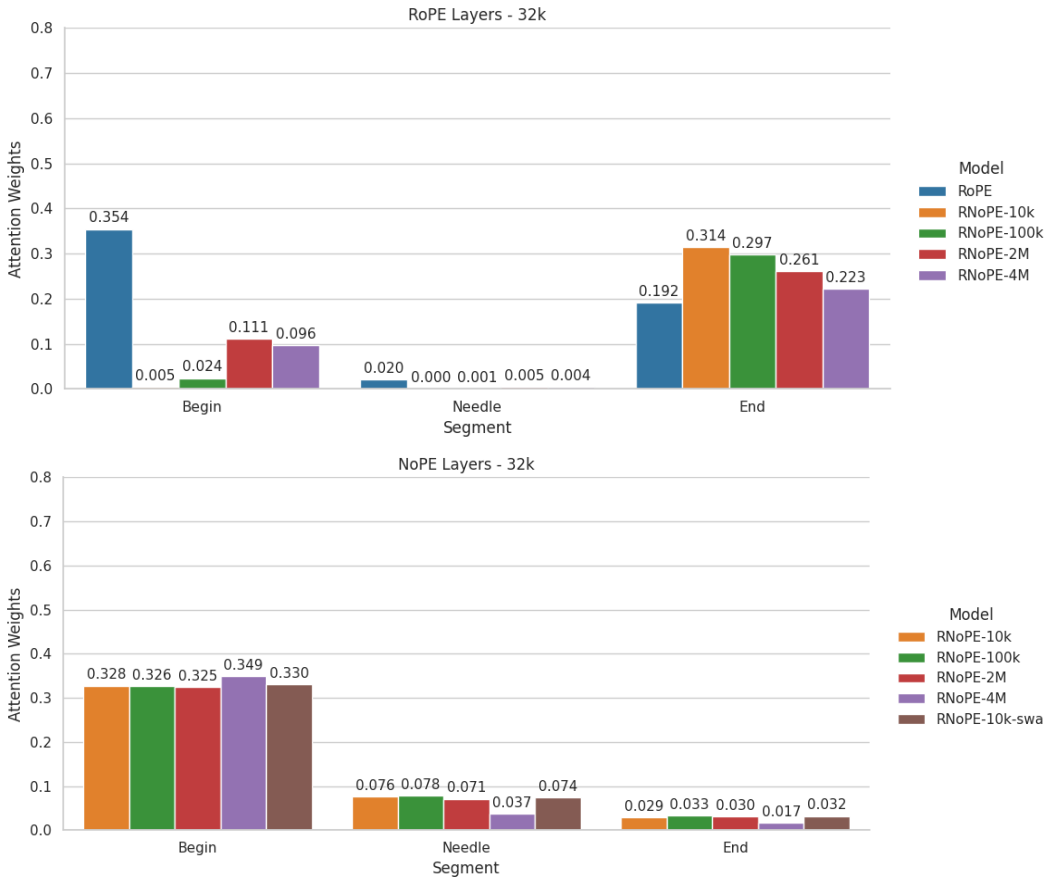


Figure 2: Attention Distribution at 32k context length for rope baseline and hybrid variants.

2.2.3 Hybrid Model

Building on the findings above, we are inspired to combine RoPE and NoPE layer-wise to leverage the strengths of both variants. NoPE provides an effective attention mechanism for retrieving information based on vector similarity, while RoPE explicitly models positional information and introduces a recency bias. By combining them, we aim to enhance overall performance. We propose a new variant that alternates between NoPE and Rotary Position Embedding (RoPE) layers. Specifically, the two position-embedding strategies are interleaved, with NoPE applied in one layer and RoPE in the next. To ensure consistency and enable meaningful comparisons, the RoPE parameter θ is initially set to 10,000 during pre-training. The pre-training procedure follows the setup described in Section 2.1 in terms of data and training protocol. Subsequently, we perform multiple fine-tuning runs with varying θ values—10,000, 100,000, 2 million, and 4 million—to evaluate performance across different configurations, using the same training steps and data as in Section 2.1. We refer to these models as RNoPE-10k, RNoPE-100k, RNoPE-2M, and RNoPE-4M with the specific RoPE θ value used for each during supervised fine-tuning (SFT).

Model	RoPE	RNoPE-10k	RNoPE-100k	RNoPE-2M	RNoPE-4M	RNoPE-10k-swa
Needles-128k Score	7.40	8.04	7.46	7.02	6.20	9.56

Table 3: Needles score at 128k for all model variants

For the RoPE baseline model and all RNoPE variants, we report the needles evaluation score of all SFT models at a sequence length of 128,000 in Table 3. We also display the attention mass of all variants in Figure 2. The attention mass is aggregated separately for all RoPE and NoPE layers. For

simplicity, we present results based on samples with 32,000 sequence lengths, with the complete table for all sequence lengths provided in Appendix A.

The results in Table 3 reveal that increasing the RoPE parameter θ during fine-tuning of the RNoPE variants—where NoPE and RoPE layers are interleaved—leads to a decline in the model's long-context capability. In contrast, previous studies on pure RoPE architectures [50, 45] have shown that using a larger RoPE θ during pre-training or fine-tuning enhances long-context performance and expands the effective attention receptive field. To investigate this discrepancy, we compare the attention mass across different model variants from Figure 2.

First, when comparing the NoPE and RoPE layers across all RNoPE variants, we observe a clear divergence in their behavior. The NoPE layers exhibit strong retrieval capabilities, characterized by a pronounced spike in attention mass on the "Needle" segment and a phenomenon of attention sink [73] on the "Begin" segment. Additionally, these layers show a significantly weaker recency bias indicated by the low attention mass on the "End" segment. In contrast, the RoPE layers within the RNoPE variants demonstrate extremely weak retrieval performance, evidenced by near-zero attention mass on the "Needle" segment and only modest attention on the "Begin" segment—indicating an absence of an attention sink. However, these RoPE layers exhibit a much stronger recency bias compared to the pure RoPE baseline. In summary, interleaving RoPE and NoPE layers induces a spontaneous "division of labor" phenomenon, where RoPE layers focus on local information aggregation and NoPE layers specialize in long-range retrieval. Remarkably, this functional specialization emerged naturally during training, without any explicit training constraints, data augmentation strategies, or specific loss function designs.

Next, we examine the RNoPE variants fine-tuned with different θ values. As θ increases, we observe a consistent decline in the recency bias of the RoPE layers. Specifically, the attention mass on the "End" segment drops from 0.314 in RNoPE-10k to 0.223 in RNoPE-4M, indicating a flatter attention distribution that reaches deeper into the context. This aligns with prior findings and theoretical analyses suggesting that larger θ values expand the effective receptive field of the attention mechanism [50]. However, our empirical results indicate that this expanded receptive field in the RoPE layers introduces noise into the overall architecture, which disrupts the NoPE layers' ability to compute similarities and perform retrieval effectively. This degradation is reflected in both attention mass and task performance: the needle attention mass in the NoPE layers drops from 0.0765 to 0.0369, and the needle evaluation score decreases from 8.036 to 6.203 as θ increases from 10,000 to 4 million. These findings further underscore that the model spontaneously develops a "division of labor" mechanism during training, with distinct roles emerging between RoPE and NoPE layers.

From these observations, we draw the following insights:

1. **Division of Labour:** Combining NoPE and RoPE layers yields complementary strengths, with each type naturally assuming specialized roles after training. NoPE layers are adapted to information retrieval, while RoPE layers become effective at modeling local context due to their inherent recency bias.

2. **Potential Efficiency Gains:** Restricting the effective attention span of the RoPE layers in RNoPE models can reduce noise in the attention distribution and reinforce the functional specialization of each layer. Additionally, it lowers the computational cost (FLOPs) during training—particularly for longer context lengths—thereby improving training efficiency while maintaining or even enhancing performance.

Guided by these insights, we fine-tune a new variant, RNoPE-10k-swa, where "swa" denotes sliding window attention. This modification imposes a hard limit on the attention span of RoPE layers, operationalizing the second insight above. Specifically, the sliding window size for RoPE layers is set to 8,192, while the NoPE layers retain full attention to support long-context retrieval. All other training configurations remain identical to the RNoPE-10k variant, including the use of $\theta = 10{,}000$. Evaluation results, presented in Table 3, show a marked improvement. The RNoPE-10k-swa variant achieves a needles-128k score of 9.562, surpassing both the baseline and the original RNoPE-10k model. Moreover, its NoPE layers exhibit a well-structured attention pattern with high attention mass on both the "Begin" and "End" segments from 2, reflecting strong long-context retrieval capabilities.

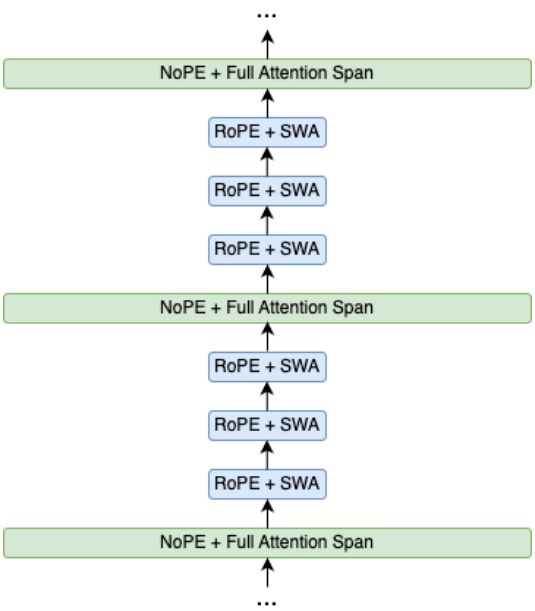


Figure 3: RNope-SWA Model Architecture. SWA stands for sliding window attention.

3 Model Architecture

Based on the analysis above, we make the following design choices on top of the Command R+ architecture [17]. First, we remove the QK-Norm component due to its poorly shaped attention patterns, which adversely impacted long context performance. Second, NoPE layers with a full attention span are introduced to enhance the model's retrieval capabilities. Third, a sliding window size of 4,096 is applied to RoPE embeddings, leveraging RoPE's inherent recency bias to improve performance on short-to-medium context ranges. In particular, the sliding-window approach has been employed in several prior works [67, 36, 11]. Regarding the number of layers, we perform an ablation study on the interleaving ratio of full attention and sliding window layers, testing the configurations of 1:1, 1:3, and 1:7. The results show that a 1:3 ratio strikes an optimal balance between computational efficiency and performance. We position the full attention layers at the end of each interleaving group, preceded by three sliding window layers. All other hyperparameters for the model architecture remain consistent with those outlined in Table 1. A visualization is shown in Figure 3.

For additional context, the resulting design shares similarities with other well-explored long-sequence architectures, such as Mega [48] and state-space models (SSMs) [28, 27]. For example, [48] and [49] introduced a multi-dimensional damped version of an exponential moving average component in conjunction with the gated attention unit (GAU) architecture [33], aiming to balance local and long-term dependencies, a common challenge in time-series modeling. Similarly, previous studies have proposed a diagonal variant of the S4 architecture [28], incorporating an exponentially decaying measure to enhance the model's ability to capture long-range dependencies [65].

Earlier transformer variants, such as GPT-J [71] and GPT-NeoX [8], explored hybridizing RoPE and NoPE by applying rotational embeddings to a subset of the head dimensions. The work of p-RoPE [6] further advanced this line of research by analyzing RoPE in the frequency domain, revealing that low-frequency components capture semantic relationships while high-frequency components encode positional information. Similar to prior partial-RoPE approaches, p-RoPE removes selected low-frequency components along the head dimensions. These findings align with our observation that NoPE primarily facilitates retrieval (semantic focus), whereas RoPE exhibits a stronger recency bias (positional focus). However, prior works did not identify the cross-layer patterns underlying the "division of labor" phenomenon described in Section 2.2.3, which enables our approach to achieve both improved performance and practical engineering benefits.

Model	MMLU	HellaSwag	ARC-E	ARC-C	SATEn	SATMath	GSM8K	Winogrande	MBPP
Baseline	57.5	75.8	**84.6**	48.5	70.0	**30.9**	40.9	68.5	39.1
RNope-SWA	**59.5**	**76.2**	82.5	**48.8**	**71.9**	30.5	**42.7**	**69.5**	**39.3**

Table 4: Comparison of models on a variety of benchmarks. All the evaluations are based on the performance of the SFT-models.

4 Experiments

In this section, we detail the experiments conducted on the model architectures, covering the stages of pretraining, cooldown, supervised fine-tuning, and evaluations. Alongside long context evaluations, we provide a comprehensive assessment of short-context benchmarks, a dimension often disregarded in other long context studies. We train two models: one with the RoPE architecture as a baseline and another employing the architecture introduced in Section 3.

Pretraining and Cooldown. The models are pretrained for 5 trillion tokens of diverse data with batch size of 8 million tokens using FP8 precision format [51]. We use a cosine learning rate schedule of 5e-3 peak learning rate and 5% end learning rate with 8,000 linear warmup steps. From the pre-trained model, we linearly anneal the learning rate from 2.5e-4 to 1e-6 for 50,000 steps in BF16 precision. The context length was initially maintained at 8k for the first 35,000 steps, then extended to 32k and 128k for 10,000 steps and 5,000 steps, respectively. For the baseline model, the RoPE θ values were increased to 1,000,000 for 32k and 8,000,000 for 128k contexts during the length extrapolation phase, while remaining constant for the RNope-SWA variant. Both models utilized the same interleaved training strategy outlined in Section 2.1.

Supervised Finetuning. We supervise fine-tune on top of the pretrained models. As the primary focus of this study is to evaluate the impact of architectural design on downstream tasks, preference training is deferred to future work. To preserve the long context capabilities of the model, the fine-tuning process utilizes interleaved datasets containing 8k and 128k prompt-response pairs. The long context SFT data at 128k includes Retrieval-Augmented Generation (RAG) [40] datasets, multilingual translation datasets with extended passages, long code instruction datasets, and long context retrieval datasets. Training was performed for two epochs across the entire dataset.

5 Experimental Results

Our evaluation contains a comprehensive analysis over standard benchmarks below 8k context length such as MMLU [30], HellaSwag [79], ARC [15], SAT [82], GSM8k [16], Winograde [58] and MBPP [3], as well as popular long context benchmarks with needles-in-a-haystack [37] and the retrieval and QA portion of Ruler [32]. We test the context lengths up to 256k so we can examine the impact of these choices in the extrapolation capability of the model. We denote the baseline model trained with RoPE scaling as “Baseline" and the architecture with interleaved attention span and position embeddings as “RNope-SWA".

5.1 Standard Benchmarks

In this set of evaluations, we evaluate baseline and RNope-SWA on a standard LLM benchmark covering various language, math and code capabilities. The results are shown in Table 4. We observe that the model is better or on-par on most of the metrics compared to the baseline and has some gains over the baseline numbers on certain benchmarks (+2.0% on MMLU and +1.8% on GSM8k). This set of results also indicates that although RNope-SWA explicitly removed position embeddings from 25% of all its layers, positional information is retained by the interleaving attention span and captured by RoPE from previous layers. RNope-SWA does not have the performance loss due to the removal of explicit position embeddings, as previous works have shown [38, 72].

5.2 Long Context Benchmarks

To evaluate the long context performance of these models, we use NIAH and the retrieval and QA components of Ruler [32]. To better understand how architectural choices affect long context

Model	Needles-128k	Needles-256k
Baseline	9.99	8.25
RNope-SWA	9.99	9.97

Table 5: Needles Score of Baseline and RNoPE-SWA up to 256k sequence length.

performance, we also evaluate with context lengths extending beyond training sequence length. This allows us to assess how well these models can interpolate as well as extrapolate to unseen context lengths and how specific architectural decisions influence these capabilities.

5.2.1 NIAH Evaluations

Following the settings of section 2.2, we run NIAH test 256k context lengths. The scores are reported in Table 5 . The figure indicates that although both models are able to get close to perfect scores below the context length seen during training, RNope-SWA has better extrapolation capabilities and achieves almost no loss up to 256k context length while the baseline fails to extrapolate well – despite using a high RoPE θ value of 8 million. We attach the graphical visualization with depth and length dimension expanded in Appendix C.

5.2.2 Ruler Evaluations

Model	8k	16k	32k	64k	128k	256k
Baseline	**96.6**	94.4	**95.1**	89.1	83.0	57.1
RNope-SWA	96.1	**96.1**	94.9	**92.0**	**90.0**	**74.8**

Table 6: Ruler Retrieval Evaluation

Model	8k	16k	32k	64k	128k	256k
Baseline	53.5	50.0	52.5	45.5	36.0	30.0
RNope-SWA	**55.5**	**52.5**	**55.5**	**49.0**	**46.0**	**42.5**

Table 7: Ruler QA Evaluation

First introduced in [32], the Ruler benchmark aims to provide a set of more difficult tasks than NIAH. It covers a wider range of retrieval under a Multi-Query/Key/Value settings, more realistic tasks with a long-context Question-Answering format and more. Although our modification of NIAH introduced more context variants and proves to be more difficult than the vanilla version, it still cannot test the limits of the model. Therefore, we evaluate our models on the retrieval and QA portion of the Ruler so we can better separate their performance.

From the results, we can observe that the baseline model with RoPE θ scaling approach suffers from a sharp drop in the longer context range, especially 64k and longer. Comparing the difference between the scores obtained at 8k and 256k, the baseline model dropped from 96.6 to 57.1 (about 41%) on retrieval and from 53.5 to 30.0 (about 44%) on QA, whereas the RNope-SWA model dropped from 96.1 to 74.8 (about 22.1%) on retrieval and from 55.5 to 42.5 (about 23.4%) on QA respectively. From the original Ruler Paper [32], models that adopt similar RoPE scaling approaches have shown a similar degradation over longer context lengths [24, 75, 1] as the baseline.

5.3 Impacts on Training and Inference

We also report the differences in training and inference speed, as well as memory requirements, of RNope-SWA compared to the baseline model. Let S denote the sliding window size and L represent the full training context length. During training, 75% of all layers now operate with a computational complexity of $\mathcal{O}(SL)$, rather than the quadratic complexity of $\mathcal{O}(L^2)$. This results in the model being approximately 50% faster than the baseline at a 64k context length and nearly 2x faster at 128k in terms of training throughput, using flash attention [20, 19, 59] and a sequence-parallel scheme similar

to [35, 76]. With alternative implementations, such as Ring Attention [43, 44] or its variants [9], sliding window adoption can reduce key-value block communication overhead if carefully sharded along the sequence dimension, potentially improving speed. In theory, KV cache memory and time complexity of RNoPE-SWA can be reduced by up to 75%. Empirically, we observed a 44% latency reduction at 132k input and 96 output tokens, and nearly 70% at 990k input and 8 output tokens—approaching the theoretical limit as sequence length grows. Increasing the ratio of sliding window to full attention layers can further improve speed and memory efficiency.

6 Discussions and Future Work

In this paper, we introduced RNope-SWA, an architecture that interleaves NoPE and RoPE position embeddings with varying attention spans (RNope-SWA). RNope-SWA is able to strike a balance between effective attention modeling and computational efficiency, achieving a nearly 4x reduction in KV cache size and significantly boosting both training and inference speeds without compromising performance. The integration of NoPE layers with full attention spans enhances long context capabilities, eliminating the need for RoPE scaling. This simplification improves the stability of training and delivers excellent long context performance.

Our findings align with recent work, such as YoCo [62], Jamba-1.5 [68], and MiniMax-01 [52], which demonstrate that hybrid attention mechanisms generally outperform full attention mechanisms in handling long contexts. However, the underlying reasons behind this seemingly counterintuitive observation remain largely unexplored. This opens an intriguing area of study, particularly as models push towards multi-million-token context lengths. Re-visiting and re-thinking the foundational components of transformer architectures, such as attention mechanisms, may become essential to accommodating these extreme requirements. Recent works [77, 39, 60] have begun to explore this direction by focusing on reducing attention noise across large context windows, a promising approach to refine the performance of attention modules.

References

[1] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, W. Chen, Y.-C. Chen, Y.-L. Chen, H. Cheng, P. Chopra, X. Dai, M. Dixon, R. Eldan, V. Fragoso, J. Gao, M. Gao, M. Gao, A. Garg, A. D. Giorno, A. Goswami, S. Gunasekar, E. Haider, J. Hao, R. J. Hewett, W. Hu, J. Huynh, D. Iter, S. A. Jacobs, M. Javaheripi, X. Jin, N. Karampatziakis, P. Kauffmann, M. Khademi, D. Kim, Y. J. Kim, L. Kurilenko, J. R. Lee, Y. T. Lee, Y. Li, Y. Li, C. Liang, L. Liden, X. Lin, Z. Lin, C. Liu, L. Liu, M. Liu, W. Liu, X. Liu, C. Luo, P. Madan, A. Mahmoudzadeh, D. Majercak, M. Mazzola, C. C. T. Mendes, A. Mitra, H. Modi, A. Nguyen, B. Norick, B. Patra, D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilac, L. Ren, G. de Rosa, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saied, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, Y. Shen, S. Shukla, X. Song, M. Tanaka, A. Tupini, P. Vaddamanu, C. Wang, G. Wang, L. Wang, S. Wang, X. Wang, Y. Wang, R. Ward, W. Wen, P. Witte, H. Wu, X. Wu, M. Wyatt, B. Xiao, C. Xu, J. Xu, W. Xu, J. Xue, S. Yadav, F. Yang, J. Yang, Y. Yang, Z. Yang, D. Yu, L. Yuan, C. Zhang, C. Zhang, J. Zhang, L. L. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, and X. Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL `https://arxiv.org/abs/2404.14219`.

[2] . AI, :, A. Young, B. Chen, C. Li, C. Huang, G. Zhang, G. Zhang, H. Li, J. Zhu, J. Chen, J. Chang, K. Yu, P. Liu, Q. Liu, S. Yue, S. Yang, S. Yang, T. Yu, W. Xie, W. Huang, X. Hu, X. Ren, X. Niu, P. Nie, Y. Xu, Y. Liu, Y. Wang, Y. Cai, Z. Gu, Z. Liu, and Z. Dai. Yi: Open foundation models by 01.ai, 2024. URL `https://arxiv.org/abs/2403.04652`.

[3] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, and C. Sutton. Program synthesis with large language models, 2021. URL `https://arxiv.org/abs/2108.07732`.

[4] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016. URL `https://arxiv.org/abs/1607.06450`.

[5] G. A. Baker, A. Raut, S. Shaier, L. E. Hunter, and K. von der Wense. Lost in the middle, and in-between: Enhancing language models' ability to reason over long contexts in multi-hop qa, 2024. URL https://arxiv.org/abs/2412.10079.

[6] F. Barbero, A. Vitvitskyi, C. Perivolaropoulos, R. Pascanu, and P. Veličković. Round and round we go! what makes rotary positional encodings useful?, 2025. URL https://arxiv.org/abs/2410.06205.

[7] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

[8] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, and S. Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. In A. Fan, S. Ilic, T. Wolf, and M. Gallé, editors, *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.bigscience-1.9. URL https://aclanthology.org/2022.bigscience-1.9/.

[9] W. Brandon, A. Nrusimha, K. Qian, Z. Ankner, T. Jin, Z. Song, and J. Ragan-Kelley. Striped attention: Faster ring attention for causal transformers, 2023. URL https://arxiv.org/abs/2311.09431.

[10] Z. Cai, Y. Zhang, B. Gao, Y. Liu, T. Liu, K. Lu, W. Xiong, Y. Dong, B. Chang, J. Hu, and W. Xiao. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling, 2024. URL https://arxiv.org/abs/2406.02069.

[11] Character.AI. Optimizing ai inference at character.ai, 2024.

[12] S. Chen, S. Wong, L. Chen, and Y. Tian. Extending context window of large language models via positional interpolation, 2023. URL https://arxiv.org/abs/2306.15595.

[13] T.-C. Chi, T.-H. Fan, P. J. Ramadge, and A. I. Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation, 2022. URL https://arxiv.org/abs/2205.09921.

[14] R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509.

[15] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

[16] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

[17] Cohere For AI. c4ai-command-r-plus (revision 432fac1), 2024. URL https://huggingface.co/CohereForAI/c4ai-command-r-plus.

[18] J. Dang, S. Singh, D. D'souza, A. Ahmadian, A. Salamanca, M. Smith, A. Peppin, S. Hong, M. Govindassamy, T. Zhao, S. Kublik, M. Amer, V. Aryabumi, J. A. Campos, Y.-C. Tan, T. Kocmi, F. Strub, N. Grinsztajn, Y. Flet-Berliac, A. Locatelli, H. Lin, D. Talupuru, B. Venkitesh, D. Cairuz, B. Yang, T. Chung, W.-Y. Ko, S. S. Shi, A. Shukayev, S. Bae, A. Piktus, R. Castagné, F. Cruz-Salinas, E. Kim, L. Crawhall-Stein, A. Morisot, S. Roy, P. Blunsom, I. Zhang, A. Gomez, N. Frosst, M. Fadaee, B. Ermis, A. Üstün, and S. Hooker. Aya expanse: Combining research breakthroughs for a new multilingual frontier, 2024. URL https://arxiv.org/abs/2412.04261.

[19] T. Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

[20] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

[21] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, R. Jenatton, L. Beyer, M. Tschannen, A. Arnab, X. Wang, C. Riquelme, M. Minderer, J. Puigcerver, U. Evci, M. Kumar, S. van Steenkiste, G. F. Elsayed, A. Mahendran, F. Yu, A. Oliver, F. Huot, J. Bastings, M. P. Collier, A. Gritsenko, V. Birodkar, C. Vasconcelos, Y. Tay, T. Mensink, A. Kolesnikov, F. Pavetić, D. Tran, T. Kipf, M. Lučić, X. Zhai, D. Keysers, J. Harmsen, and N. Houlsby. Scaling vision transformers to 22 billion parameters, 2023. URL https://arxiv.org/abs/2302.05442.

[22] J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, N. Zheng, and F. Wei. Longnet: Scaling transformers to 1,000,000,000 tokens, 2023. URL https://arxiv.org/abs/2307.02486.

[23] Y. Ding, L. L. Zhang, C. Zhang, Y. Xu, N. Shang, J. Xu, F. Yang, and M. Yang. Longrope: Extending llm context window beyond 2 million tokens, 2024. URL https://arxiv.org/abs/2402.13753.

[24] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, and e. a. Angela Fan. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[25] T. Gao, A. Wettig, H. Yen, and D. Chen. How to train long-context language models (effectively), 2024. URL https://arxiv.org/abs/2410.02660.

[26] O. Golovneva, T. Wang, J. Weston, and S. Sukhbaatar. Contextual position encoding: Learning to count what's important, 2024. URL https://arxiv.org/abs/2405.18719.

[27] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. URL https://arxiv.org/abs/2312.00752.

[28] A. Gu, K. Goel, and C. Ré. Efficiently modeling long sequences with structured state spaces, 2022. URL https://arxiv.org/abs/2111.00396.

[29] A. Haviv, O. Ram, O. Press, P. Izsak, and O. Levy. Transformer language models without positional encodings still learn positional information. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1382–1390, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.99. URL https://aclanthology.org/2022.findings-emnlp.99/.

[30] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

[31] A. Henry, P. R. Dachapally, S. S. Pawar, and Y. Chen. Query-key normalization for transformers. In T. Cohn, Y. He, and Y. Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4246–4253, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.379. URL https://aclanthology.org/2020.findings-emnlp.379/.

[32] C.-P. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, Y. Zhang, and B. Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

[33] W. Hua, Z. Dai, H. Liu, and Q. V. Le. Transformer quality in linear time, 2022. URL https://arxiv.org/abs/2202.10447.

[34] Y. Huang, J. Xu, J. Lai, Z. Jiang, T. Chen, Z. Li, Y. Yao, X. Ma, L. Yang, H. Chen, S. Li, and P. Zhao. Advancing transformer architecture in long-context large language models: A comprehensive survey, 2024. URL https://arxiv.org/abs/2311.12351.

[35] S. A. Jacobs, M. Tanaka, C. Zhang, M. Zhang, S. L. Song, S. Rajbhandari, and Y. He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models, 2023. URL https://arxiv.org/abs/2309.14509.

[36] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[37] G. Kamradt. Needle in a haystack–pressure testing llms, 2023.

[38] A. Kazemnejad, I. Padhi, K. N. Ramamurthy, P. Das, and S. Reddy. The impact of positional encoding on length generalization in transformers, 2023. URL https://arxiv.org/abs/2305.19466.

[39] Y. Leviathan, M. Kalman, and Y. Matias. Selective attention improves transformer, 2024. URL https://arxiv.org/abs/2410.02703.

[40] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

[41] Y. Li, Y. Huang, B. Yang, B. Venkitesh, A. Locatelli, H. Ye, T. Cai, P. Lewis, and D. Chen. Snapkv: Llm knows what you are looking for before generation, 2024. URL https://arxiv.org/abs/2404.14469.

[42] Z. Li, J. Zhang, Q. Lin, J. Xiong, Y. Long, X. Deng, Y. Zhang, X. Liu, M. Huang, Z. Xiao, and D. C. et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding, 2024. URL https://arxiv.org/abs/2405.08748.

[43] H. Liu and P. Abbeel. Blockwise parallel transformer for large context models, 2023. URL https://arxiv.org/abs/2305.19370.

[44] H. Liu, M. Zaharia, and P. Abbeel. Ring attention with blockwise transformers for near-infinite context, 2023. URL https://arxiv.org/abs/2310.01889.

[45] H. Liu, W. Yan, M. Zaharia, and P. Abbeel. World model on million-length video and language with blockwise ringattention, 2024. URL https://arxiv.org/abs/2402.08268.

[46] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

[47] X. Liu, H. Yan, S. Zhang, C. An, X. Qiu, and D. Lin. Scaling laws of rope-based extrapolation, 2024. URL https://arxiv.org/abs/2310.05209.

[48] X. Ma, C. Zhou, X. Kong, J. He, L. Gui, G. Neubig, J. May, and L. Zettlemoyer. Mega: Moving average equipped gated attention, 2023. URL https://arxiv.org/abs/2209.10655.

[49] X. Ma, X. Yang, W. Xiong, B. Chen, L. Yu, H. Zhang, J. May, L. Zettlemoyer, O. Levy, and C. Zhou. Megalodon: Efficient llm pretraining and inference with unlimited context length, 2024. URL https://arxiv.org/abs/2404.08801.

[50] X. Men, M. Xu, B. Wang, Q. Zhang, H. Lin, X. Han, and W. Chen. Base of rope bounds context length, 2024. URL https://arxiv.org/abs/2405.14591.

[51] P. Micikevicius, D. Stosic, N. Burgess, M. Cornea, P. Dubey, R. Grisenthwaite, S. Ha, A. Heinecke, P. Judd, J. Kamalu, N. Mellempudi, S. Oberman, M. Shoeybi, M. Siu, and H. Wu. Fp8 formats for deep learning, 2022. URL https://arxiv.org/abs/2209.05433.

[52] MiniMax, A. Li, B. Gong, B. Yang, B. Shan, C. Liu, C. Zhu, C. Zhang, C. Guo, D. Chen, D. Li, E. Jiao, G. Li, G. Zhang, H. Sun, H. Dong, J. Zhu, J. Zhuang, J. Song, J. Zhu, J. Han, J. Li, J. Xie, J. Xu, J. Yan, K. Zhang, K. Xiao, K. Kang, L. Han, L. Wang, L. Yu, L. Feng, L. Zheng, L. Chai, L. Xing, M. Ju, M. Chi, M. Zhang, P. Huang, P. Niu, P. Li, P. Zhao, Q. Yang, Q. Xu, Q. Wang, Q. Wang, Q. Li, R. Leng, S. Shi, S. Yu, S. Li, S. Zhu, T. Huang, T. Liang, W. Sun, W. Sun, W. Cheng, W. Li, X. Song, X. Su, X. Han, X. Zhang, X. Hou, X. Min, X. Zou, X. Shen, Y. Gong, Y. Zhu, Y. Zhou, Y. Zhong, Y. Hu, Y. Fan, Y. Yu, Y. Yang, Y. Li, Y. Huang, Y. Li,

Y. Huang, Y. Xu, Y. Mao, Z. Li, Z. Li, Z. Tao, Z. Ying, Z. Cong, Z. Qin, Z. Fan, Z. Yu, Z. Jiang, and Z. Wu. Minimax-01: Scaling foundation models with lightning attention, 2025. URL https://arxiv.org/abs/2501.08313.

[53] A. Mohtashami and M. Jaggi. Landmark attention: Random-access infinite context length for transformers, 2023. URL https://arxiv.org/abs/2305.16300.

[54] B. Peng, J. Quesnelle, H. Fan, and E. Shippole. Yarn: Efficient context window extension of large language models, 2023. URL https://arxiv.org/abs/2309.00071.

[55] O. Press, N. A. Smith, and M. Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022. URL https://arxiv.org/abs/2108.12409.

[56] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

[57] O. Rybakov, M. Chrzanowski, P. Dykas, J. Xue, and B. Lanir. Methods of improving llm training stability, 2024. URL https://arxiv.org/abs/2410.16682.

[58] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

[59] J. Shah, G. Bikshandi, Y. Zhang, V. Thakkar, P. Ramani, and T. Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision, 2024. URL https://arxiv.org/abs/2407.08608.

[60] Y. J. Soh, H. Huang, Y. Tian, and J. Zhao. You only use reactive attention slice for long context retrieval, 2024. URL https://arxiv.org/abs/2409.13695.

[61] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

[62] Y. Sun, L. Dong, Y. Zhu, S. Huang, W. Wang, S. Ma, Q. Zhang, J. Wang, and F. Wei. You only cache once: Decoder-decoder architectures for language models, 2024. URL https://arxiv.org/abs/2405.05254.

[63] A. Talmor, J. Herzig, N. Lourie, and J. Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019. URL https://arxiv.org/abs/1811.00937.

[64] Y. Tay, D. Bahri, L. Yang, D. Metzler, and D.-C. Juan. Sparse sinkhorn attention, 2020. URL https://arxiv.org/abs/2002.11296.

[65] Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler. Long range arena: A benchmark for efficient transformers, 2020. URL https://arxiv.org/abs/2011.04006.

[66] C. Team. Chameleon: Mixed-modal early-fusion foundation models, 2024. URL https://arxiv.org/abs/2405.09818.

[67] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, and e. a. Bobak Shahriari. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

[68] J. Team, B. Lenz, A. Arazi, A. Bergman, A. Manevich, B. Peleg, B. Aviram, C. Almagor, C. Fridman, D. Padnos, D. Gissin, D. Jannai, D. Muhlgay, D. Zimberg, E. M. Gerber, E. Dolev, E. Krakovsky, E. Safahi, E. Schwartz, G. Cohen, G. Shachaf, H. Rozenblum, H. Bata, I. Blass, I. Magar, I. Dalmedigos, J. Osin, J. Fadlon, M. Rozman, M. Danos, M. Gokhman, M. Zusman, N. Gidron, N. Ratner, N. Gat, N. Rozen, O. Fried, O. Leshno, O. Antverg, O. Abend, O. Lieber, O. Dagan, O. Cohavi, R. Alon, R. Belson, R. Cohen, R. Gilad, R. Glozman, S. Lev, S. Meirom, T. Delbari, T. Ness, T. Asida, T. B. Gal, T. Braude, U. Pumerantz, Y. Cohen, Y. Belinkov, Y. Globerson, Y. P. Levy, and Y. Shoham. Jamba-1.5: Hybrid transformer-mamba models at scale, 2024. URL https://arxiv.org/abs/2408.12570.

[69] S. Tworkowski, K. Staniszewski, M. Pacek, Y. Wu, H. Michalewski, and P. Miłoś. Focused transformer: Contrastive training for context scaling, 2023. URL https://arxiv.org/abs/2307.03170.

[70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[71] B. Wang and A. Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[72] J. Wang, T. Ji, Y. Wu, H. Yan, T. Gui, Q. Zhang, X. Huang, and X. Wang. Length generalization of causal transformers without position encoding, 2024. URL https://arxiv.org/abs/2404.12224.

[73] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

[74] X. Xu, Q. Ye, and X. Ren. Stress-testing long-context language models with lifelong icl and task haystack, 2024. URL https://arxiv.org/abs/2407.16695.

[75] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, and Z. Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

[76] A. Yang, J. Yang, A. Ibrahim, X. Xie, B. Tang, G. Sizov, J. Reizenstein, J. Park, and J. Huang. Context parallelism for scalable million-token inference, 2024. URL https://arxiv.org/abs/2411.01783.

[77] T. Ye, L. Dong, Y. Xia, Y. Sun, Y. Zhu, G. Huang, and F. Wei. Differential transformer, 2024. URL https://arxiv.org/abs/2410.05258.

[78] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. Big bird: Transformers for longer sequences, 2021. URL https://arxiv.org/abs/2007.14062.

[79] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

[80] P. Zhang, Z. Liu, S. Xiao, N. Shao, Q. Ye, and Z. Dou. Long context compression with activation beacon, 2024. URL https://arxiv.org/abs/2401.03462.

[81] Z. Zhang, Y. Sheng, T. Zhou, T. Chen, L. Zheng, R. Cai, Z. Song, Y. Tian, C. Ré, C. Barrett, Z. Wang, and B. Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models, 2023. URL https://arxiv.org/abs/2306.14048.

[82] W. Zhong, R. Cui, Y. Guo, Y. Liang, S. Lu, Y. Wang, A. Saied, W. Chen, and N. Duan. Agieval: A human-centric benchmark for evaluating foundation models, 2023. URL https://arxiv.org/abs/2304.06364.

[83] A. Üstün, V. Aryabumi, Z.-X. Yong, W.-Y. Ko, D. D'souza, G. Onilude, N. Bhandari, S. Singh, H.-L. Ooi, A. Kayid, F. Vargus, P. Blunsom, S. Longpre, N. Muennighoff, M. Fadaee, J. Kreutzer, and S. Hooker. Aya model: An instruction finetuned open-access multilingual language model, 2024. URL https://arxiv.org/abs/2402.07827.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract clearly states the contributions of this paper is analyzing existing attention approaches and proposing a new hybrid attention architecture with better performance and efficiency.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The approach described by this paper is robust and verified with ample training and evaluations. Computational efficiency is also covered in Section 5.3. We discuss limitation in Section 6 where the reason behind hybrid architectures working better than traditional dense architecture is still underexplored.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper is empirical and involve no proof on results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: This paper gives detailed explanation on the model architecture, training methodology and type of data used. Although we don't open source the data or the model, the process should be easy to follow for people who work on this area.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We did not open source the data or code for training.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper covers details on training steps, learning rates, hyper-parameters, optimizers used and the essential information on the data used in Section 2 and Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The results of this paper is very clear and conspicuous among model variants.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

Justification: Compute resources required to train varies depending on the type of hardware and frameworks used. It is also not very relevant to the paper's focus.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper completely conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This is not related to this paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Not related to this paper.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: This is done properly.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No assets are introduced in this paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper doesn't involve the crowdsourcing or human subjects).

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not applicable to this paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core content of this research does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Attention Distribution of All Lengths

This table contains the attention distribution of RoPE and RNoPE variants from Section 2.2.2 over 8k, 32k and 128k sequence lengths.

Context Length	Model	NoPE Layers				RoPE Layers			
		Begin	Needle	Context	End	Begin	Needle	Context	End
8k	RoPE	-	-	-	-	0.3863	0.0328	0.3809	0.2000
	RNoPE-10k	0.3900	0.0952	0.4736	0.0412	0.1255	0.0102	0.5340	0.3302
	RNoPE-100k	0.3854	0.0932	0.4783	0.0430	0.2135	0.0136	0.4558	0.3171
	RNoPE-2M	0.3775	0.0902	0.4874	0.0449	0.2041	0.0126	0.4952	0.2881
	RNoPE-4M	0.4153	0.0546	0.5072	0.0229	0.1389	0.0136	0.6162	0.2313
	RNoPE-10k-swa	0.3830	0.1025	0.4702	0.0443	0.2040	0.0110	0.5938	0.1911
32k	RoPE	-	-	-	-	0.3541	0.0201	0.4343	0.1915
	RNoPE-10k	0.3275	0.0765	0.5672	0.0287	0.0049	0.0004	0.6805	0.3142
	RNoPE-100k	0.3263	0.0778	0.5633	0.0327	0.0241	0.0005	0.6782	0.2972
	RNoPE-2M	0.3250	0.0712	0.5735	0.0303	0.1111	0.0046	0.6233	0.2611
	RNoPE-4M	0.3486	0.0369	0.5981	0.0165	0.0960	0.0039	0.6774	0.2227
	RNoPE-10k-swa	0.3303	0.0742	0.5634	0.0321	-	-	-	-
128k	RoPE	-	-	-	-	0.3463	0.0010	0.4751	0.1776
	RNoPE-10k	0.2991	0.0444	0.6430	0.0135	0.0000	0.0001	0.7230	0.2769
	RNoPE-100k	0.2454	0.0419	0.7016	0.0111	0.0001	0.0000	0.7749	0.2250
	RNoPE-2M	0.2600	0.0401	0.6836	0.0162	0.0417	0.0008	0.7516	0.2059
	RNoPE-4M	0.2949	0.0307	0.6635	0.0109	0.0663	0.0022	0.7115	0.2230
	RNoPE-10k-swa	0.2760	0.0467	0.6615	0.0159	-	-	-	-

Table 8: Needles Attention Pattern: RoPE and RNoPE variants

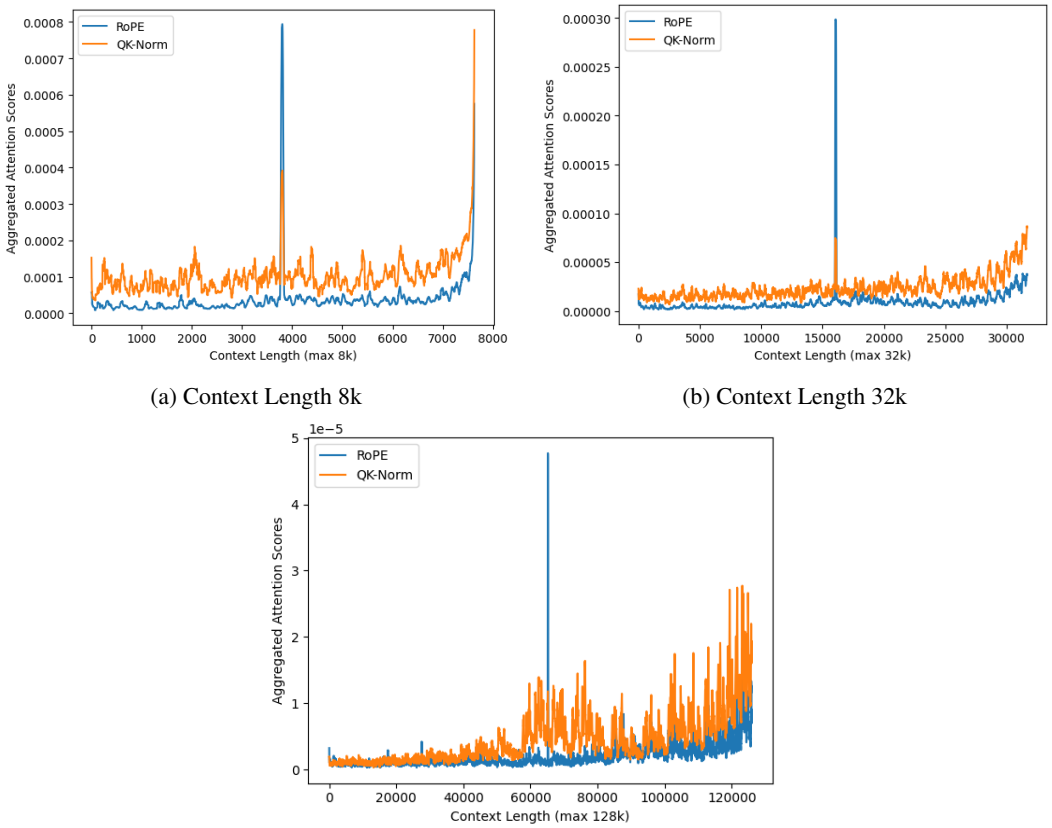
B Attention Distribution of RoPE and QK-Norm variants

In this section, we further investigate the suboptimal performance of the QK-Norm variant. We present three plots comparing the attention distribution between the RoPE and QK-Norm variants across sequence lengths of 8k, 32k, and 128k on needle samples, following the setup outlined in Section 2.2.2. Additionally, we provide the aggregated attention entropy for each variant to quantitatively support the arguments.

To enhance the clarity of the distribution plots, we preprocess the attention distribution array by removing the first 10 tokens and the last 3% of tokens from each sequence. This preprocessing step mitigates the disproportionate attention mass resulting from the attention sink effect and the recency bias observed in RoPE, thereby making the attention patterns more interpretable. We then compute a moving average with a window size of 100 tokens and average the results across all samples and layers to generate the final distributions.

Model	8k	32k	128k
RoPE	**6.02**	**6.95**	**7.62**
QK-Norm	10.71	12.46	14.14

Table 9: Entropy values of aggregated attention distribution



(a) Context Length 8k

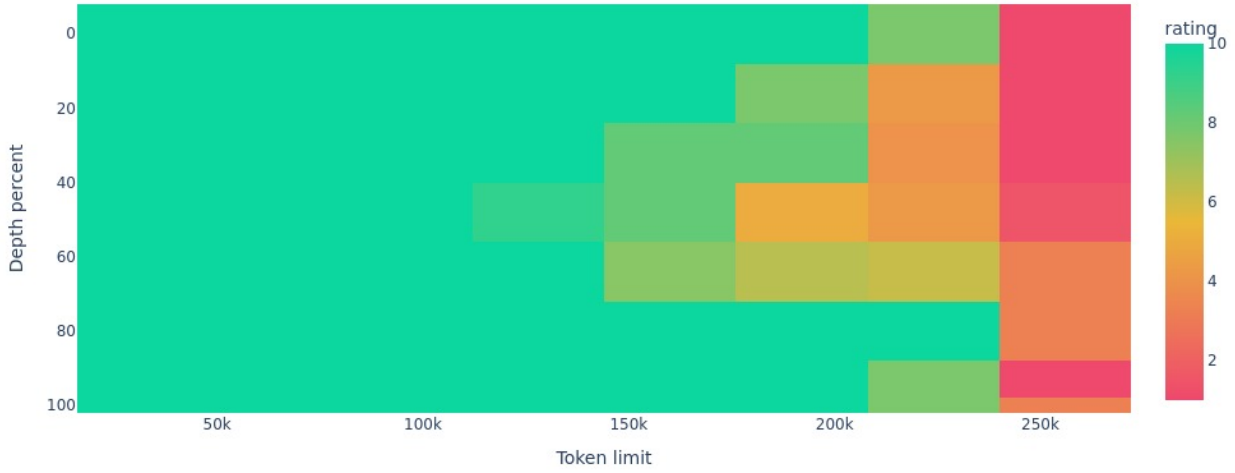
(b) Context Length 32k

(c) Context Length 128k

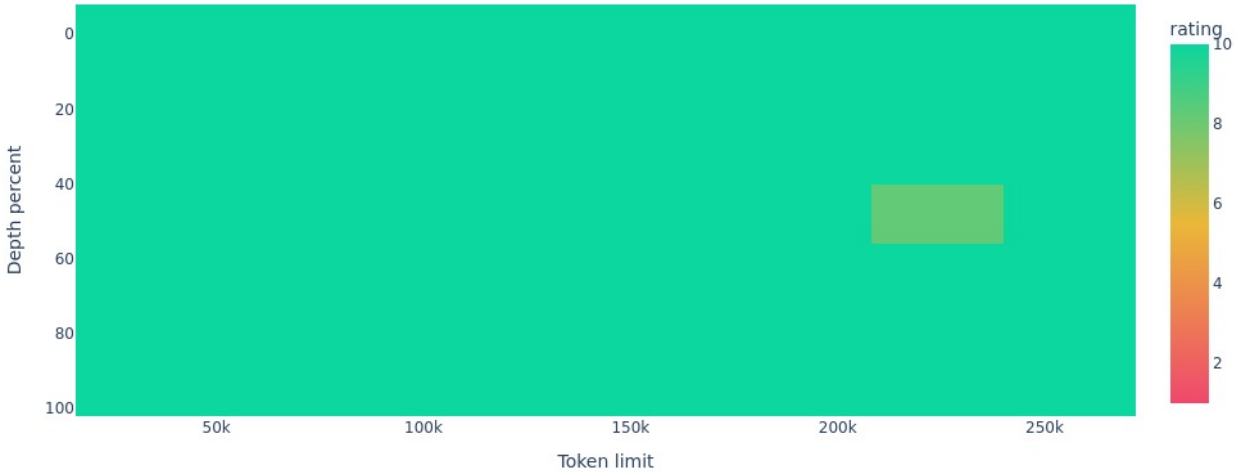
Figure 4: Attention Distribution Across Sequence lengths

From Figure 4, we observe that the QK-Norm variant exhibits a lower spike on needle tokens but distributes more attention mass across context tokens. However, it also demonstrates a stronger recency bias compared to the RoPE variant. This characteristic results in a lower signal-to-noise ratio for the QK-Norm variant, which hampers its ability to effectively retrieve relevant information from long contexts. To further quantify this observation, we calculate the entropy values of the attention distributions for both variants, averaging across samples and layers at each sequence length. The results, listed in Table 9, show that the QK-Norm variant has significantly higher entropy values than the RoPE variant. This aligns with its weaker performance in long context retrieval tasks, as higher entropy reflects a more dispersed and less focused attention distribution.

C Needles Score at 256k



(a) Baseline



(b) RNoPE-SWA

Figure 5: Needle Evaluation of Baseline and RNoPE-SWA on 256k sequence length