SWAN-GPT: An Efficient and Scalable Approach for Long-Context Language Modeling

Krishna C. Puvvada [*1] **Faisal Ladhak** [*1] **Santiago Akle Serrano** [1] **Cheng-Ping Hsieh** [1] **Shantanu Acharya** [1]
Somshubra Majumdar [1] **Fei Jia** [1] **Samuel Kriman** [1] **Simeng Sun** [1] **Dima Rekesh** [1] **Boris Ginsburg** [1]

Abstract

We present SWAN-GPT, a decoder-only Transformer architecture that generalizes to sequence lengths substantially longer than those seen during training. SWAN-GPT interleaves layers without positional encodings (NoPE) and sliding-window attention layers with rotary positional encodings (SWA-RoPE). Our experiments demonstrate strong performance on sequences significantly longer than the training length without specialized long-context training. This robust length extrapolation is achieved through our novel architecture, enhanced by dynamic scaling of attention scores during inference. Additionally, SWAN-GPT is more computationally efficient than standard GPT architectures, and existing pre-trained models can be efficiently converted to the SWAN architecture with minimal continued training.

1. Introduction

Large Language Models based on standard decoder-only transformer architectures (Brown et al., 2020; Grattafiori et al., 2024; Yang et al., 2025a) struggle with context lengths beyond their training distribution. Current approaches to extending context length either require specialized training on longer sequences (Grattafiori et al., 2024; Yang et al., 2025a; Peng et al., 2023b; Chen et al., 2023) or complex inference time modifications (An et al., 2024), incurring increased computational cost or implementation complexity. We propose SWAN-GPT, a decoder-only transformer architecture that natively handles sequences substantially longer than seen during training. By strategically interleaving global attention layers without positional encodings and local sliding-window attention layers with rotary position encodings, combined with dynamic attention scaling, SWAN-GPT maintains comparable performance to standard transformers while robustly extrapolating to longer sequences.

A central challenge in extending transformer context lengths is the handling of positional information. Transformers rely on positional encodings to track token order, but these encodings often become unreliable when models process sequences longer than seen during training. Rotary Positional Encodings (RoPE) (Su et al., 2023) have been widely adopted due to their effectiveness in capturing relative positions, but RoPE-based models degrade significantly when applied to sequences exceeding their training length as inter-token distances advance to ranges where the relative rotation angle is outside the trained distribution (Liu et al., 2024). To address this limitation, we explore two complementary approaches: Sliding window attention with RoPE (SWA-RoPE), which restricts attention to fixed-size windows of neighboring tokens, maintaining robustness to arbitrary sequence lengths but limiting long-range dependencies; and layers without positional encoding (NoPE) (Haviv et al., 2022; Kazemnejad et al., 2023), which allow unrestricted attention, but exhibit poor robustness beyond their training length (Kazemnejad et al., 2023; Wang et al., 2024).

We propose SWAN-GPT, which strategically interleaves these approaches. This hybrid design creates a synergistic effect: SWA-RoPE layers provide local positional structure, while NoPE layers integrate information across arbitrary distances. When interleaved, the NoPE layers develop more robust representations than in isolation, enabling the model to generalize beyond its training sequence length. Unlike standard RoPE-based transformers which experience performance collapse outside their training context, SWAN maintains robust performance on extended sequences with only straightforward rescaling of attention scores during inference. Our contributions include: (1) a novel architecture combining SWA-RoPE and NoPE layers with logarithmic attention scaling, (2) mechanistic analysis explaining why this architecture produces robust length extrapolation, (3) empirical results showing robust performance on sequences far exceeding training length, and (4) a practical approach for

[*]Equal contribution [1]NVIDIA, USA. Correspondence to: Krishna C. Puvvada <kpuvvada@nvidia.com>, Faisal Ladhak <fladhak@nvidia.com>.

Proceedings of the 2^{nd} Workshop on Long-Context Foundation Models, Vancouver, Canada. 2025.

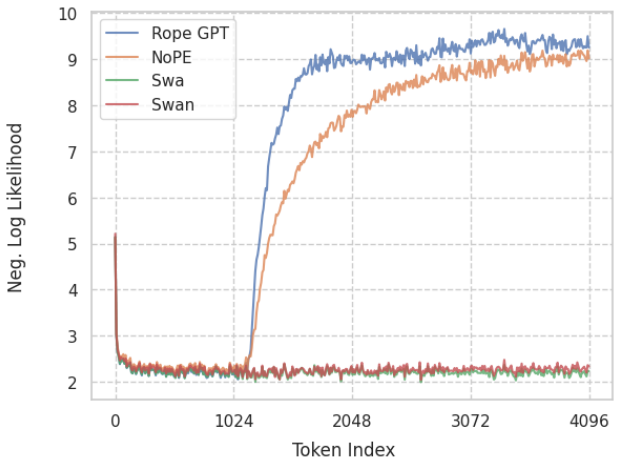


Figure 1: Mean negative log likelihood by token position. RoPE and NoPE models deteriorate beyond training sequence length (1024). SWA model doesn't experience such catastrophic failure due to its limited context. SWAN model behaves like a SWA model without the limitation of SWA model due to its global NoPE layers.

adapting existing models via continued pre-training (CPT).

2. The SWAN-GPT architecture

SWAN-GPT is a decoder-only architecture that addresses the challenge of length extrapolation by interleaving two attention mechanisms: global attention layers without positional encodings (NoPE) and local sliding-window attention layers with rotary positional encodings (SWA-RoPE). This hybrid design leverages the complementary strengths of both approaches to achieve robust length extrapolation capabilities, without specialized long-context training.

As detailed in Appendix A, we explored multiple configurations for interleaving these layer types. Our experiments revealed that beginning with a global NoPE layer followed by three consecutive sliding-window layers, repeating this pattern throughout the network, demonstrated superior performance on long-context tasks. This configuration achieves exceptional NIAH scores at context lengths 16 times longer than the training length, and maintains robust performance even at 32 times the training length when combined with appropriate attention scaling (subsection 2.2). The global NoPE layers permit unrestricted attention across the entire context, enabling the model to capture long-range dependencies. Meanwhile, the local SWA-RoPE layers operate with a fixed window size of 512 tokens, providing consistent positional information within a bounded context. The key insight is that when these mechanisms are interleaved, the NoPE layers develop more robust position-aware representations than they would in isolation, enabling the entire model to generalize effectively beyond training sequence lengths.

Figure 1 demonstrates this capability by comparing four models trained on sequences of up to 1024 tokens: a standard GPT model with RoPE, one with no positional encodings (NoPE), one with only sliding window attention (SWA) and one using our architecture (SWAN). We evaluate the model's predictions on 1280 validation sequences of length 4096. The plot shows the negative log likelihood at each sequence position averaged over all validation sequences, with lower values indicating better performance. Both RoPE and NoPE experience significant performance degradation beyond their training length, with negative log likelihood increasing sharply beyond 1024 tokens. In contrast, both SWAN and SWA maintain consistent predictive quality throughout the entire 4096-token range, demonstrating their robust length extrapolation capabilities. Notably, SWAN maintains this performance while retaining the ability to capture long-range dependencies that the purely local SWA approach cannot (see Appendix A).

2.1. Position Encoding Dynamics in SWAN-GPT

A key question is why NoPE layers in SWAN architecture demonstrate substantially more robust length extrapolation compared to identical layers within pure NoPE models. Despite the absence of explicit positional encoding, NoPE models implicitly learn to predict token positions (Chi et al., 2023). However, standard NoPE models exhibit poor length extrapolation. In our architecture, the interleaved SWA-RoPE layers appear to relieve NoPE layers from developing the brittle position encodings typically seen in pure NoPE implementations. Our analysis shows that position probes trained on SWAN's NoPE layers exhibit little positional information across all sequence positions, unlike those trained on pure NoPE models which fail beyond the training boundary. Further, attention pattern analysis reveals that SWAN maintains consistent attention patterns for sequences both within and beyond the training length. Detailed analyses of these phenomena are provided in Appendix D.

2.2. Dynamic Attention Scaling for Extended Context Processing

While our architecture demonstrates inherent sequence length extrapolation, further performance improvements can be achieved through proper scaling of attention logits during inference. This scaling is particularly important for the global NoPE layers, which must effectively integrate information across arbitrary distances as sequence length increases. We find that a logarithmic scaling function $\log_a(a+n)$ provides an excellent fit to the empirically determined optimal scaling factors, unlike YaRN scaling (Peng et al., 2023b) which fits poorly for NoPE layers, where a is a tunable constant and n is position index of token in sequence. See Appendix E for more details.

Model	ARC-E	ARC-C	Hellaswag	Winogrande	RACE	PIQA	SIQA	OBQA	Avg
GPT-1B	65.36	38.23	58.35	57.93	35.02	73.12	32.91	35.20	49.5
SWAN-1B	69.40	41.04	59.76	59.75	35.69	73.99	33.73	37.80	51.4

Table 1: Results for 1B models trained on 1T tokens. SWAN shows comparable or better performance across all benchmarks.

Model	8K	16K	32K	64K	128K	256k
GPT-1B	53.5	NA	NA	NA	NA	NA
SWAN-1B	52.4	45.8	36.9	30.6	24.4	14.9

Table 2: Long-context performance on RULER. Both models are trained on 8k sequences. SWAN maintains measurable performance at 32 times training length, while RoPE fails beyond training length.

3. Results

In the previous section, we introduced the SWAN architecture and motivated its robust length extrapolation via mechanistic analysis and empirical experiments. Here, we evaluate the effectiveness of the proposed architecture compared to standard RoPE-based transformer LLMs. Our goal is to demonstrate that SWAN models can maintain similar performance on standard LLM benchmarks while achieving substantial length extrapolation.

We trained both RoPE GPT and SWAN models of 1B size from scratch using 1T tokens at 8K sequence length with a token batch size of 6M. The SWAN model followed 1:3 global:local ratio, with sliding window attention layers using a 512-token window size. We evaluated both models on standard LLM benchmarks using the LM Evaluation Harness (Gao et al., 2024). The SWAN model performs comparably or better than the RoPE model across all benchmarks, achieving an avg 51.4% vs. 49.5% (Table 1). SWAN's primary advantage emerges on sequences longer than training length. RULER (Hsieh et al., 2024) results (Table 2) show that while both models perform similarly within training distribution ($\leq 8K$), the RoPE GPT model fails completely at longer sequences, while SWAN exhibits graceful degradation even at substantially longer contexts.

3.1. Efficient Adaptation of Pre-trained Models to SWAN Architecture

While training from scratch demonstrates SWAN's superior length extrapolation capabilities, adapting existing pre-trained models would significantly enhance practical utility of our approach. Since transformer knowledge is primarily encoded in feed-forward layers (Geva et al., 2021), and SWAN only modifies attention computation, we hypothesize existing models can be efficiently converted to SWAN architecture without losing accumulated knowledge.

We adapted an 8B RoPE GPT model pre-trained on 15T tokens at 8K context (Su et al., 2024). The conversion involved initializing SWAN weights from the pre-trained model and implementing 1:3 global-local attention pattern with 512-token windows. Following initialization, we performed CPT for 315B tokens (2% of original pre-training budget) at 32K context length, with Fill-in-Middle augmentation (Bavarian et al., 2022) for the final 15B tokens to further enhance the model's contexual understanding. Post-training for RoPE GPT model was conducted in two stages, with the first stage focusing on math and code followed by a general SFT in the second stage. Post-training for SWAN followed similar procedure, but with the sequence length extended to 32K through concatenation of shorter examples. To enhance long-context capabilities, SFT data was augmented with a variety of tasks designed to exercise the model's ability to reason over extended contexts. These included questions referring to previous turns in concatenated examples and synthetic tasks such as filling in the middle, recalling portions of context based on keywords, tracing linked lists, executing basic SQL queries on made-up table data, and multi-hop reasoning (Chen et al., 2024b) tasks modified to 32K sequence length.

Table 4 compares our adapted SWAN GPT-8B model with the original RoPE GPT-8B model. The results demonstrate that SWAN adaptation maintains comparable performance across diverse set of tasks, including mathematical reasoning (GSM8k, MATH500), coding (MBPP, HumanEval), and general language understanding (MMLU, IFEval, MT-Bench), with minimal performance decrease from 71.55% to 70.95% on average. This confirms that substantial architectural modifications to the attention mechanism can be implemented with brief adaptation while preserving fundamental capabilities. The primary advantage, however, is substantial improvement in length extrapolation.

Table 3 compares SWAN GPT-8B against state-of-the-art models on the RULER (Hsieh et al., 2024). Despite training at only 32K context length, SWAN demonstrates remarkable extrapolation: 80.5 at 64K context ($2\times$ training length), 73.2 at 256K context ($8\times$), 63.5 at 1M context ($32\times$) and 60.1 at 2M context ($64\times$). This robust extrapolation contrasts sharply with other models' performance dropoff patterns. Qwen2.5-7B-Instruct (128K), also trained at 32K maximum length, drops from 82.3 at 64K to 55.1 at 128K tokens, while SWAN exhibits much more gradual degradation. Even com-

Model	MTL	4K	8K	16K	32K	64K	128K	256k	512k	1M	2M
Llama3.1-8B	128K	95.5	93.8	91.6	87.4	84.7	77.0	-	-	-	-
Qwen2.5-7B-Instruct	32K	96.7	95.1	93.7	89.4	82.3	55.1	-	-	-	-
Qwen2.5-7B-Instruct-1M	256K	96.8	95.3	93.0	91.1	90.4	84.4	75.3	64.6	-	-
SwanGPT-8B	32K	93.8	90.8	88.1	84.4	80.5	77.8	73.2	67.3	63.5	60.1

Table 3: Long-context performance comparison on RULER benchmark. MTL=Maximum training length. RoPE based models degrade fast with increased sequence length whereas SWAN exhibits a more graceful dropoff.

Benchmark	RoPE	SWAN
Math		
GSM8k	87.7	87.7
MATH500	70.4	68.4
Code		
MBPP	76.2	75.7
MBPP+	66.1	65.3
HumanEval	74.4	75.0
HumanEval+	68.3	68.3
General		
MT-Bench	7.35	7.43
MMLU (gen.)	68.0	65.4
IFEval (P)	63.0	62.7
IFEval (I)	72.7	72.2
Tool Use / Long Context		
BFCL v2 Live	68.7	68.9
RULER (128k)	NA	77.8
Avg. (excl. MT, RULER)	71.55	70.95

Table 4: Comparison of RoPE vs. SWAN when adapting a pre-trained model. SWAN maintains comparable performance while attaining long-context capabilities.

pared to models trained on longer contexts (Llama3.1-8B (128K training) and Qwen2.5-7B-Instruct (1M) (256K training)), SWAN remains competitive. SWAN's 77.8 at 128K tokens matches Llama3.1-8B's 77.0, despite being trained on contexts one-fourth as long. SWAN outperforms Qwen2.5-7B-Instruct (1M) at 512K context length, despite the latter's eight-times-longer training sequences. Even at 2M context (64$\times$ training length), SWAN achieves 60.1, demonstrating that SWAN architecture enables efficient adaptation of existing pre-trained models to handle significantly longer contexts without sacrificing short context performance, providing a practical, compute-efficient path for upgrading already deployed models without full retraining.

4. Related Work

Extending LLM context length to hundreds of thousands of tokens presents significant architectural, computational, and data challenges (Lv et al., 2024; Gao et al., 2025; Liu et al., 2025). Current approaches primarily involve inference-time modifications to RoPE or attention mechanisms (e.g., NTK-aware scaling (bloc97, 2023b;a), ReRoPE (Su, 2023), StreamingLLM (Xiao et al., 2024)), which can degrade performance or require careful tuning. Alternatively, training-based strategies, including continued pre-training (CPT) or staged pre-training with progressively longer sequences (e.g., YaRN (Peng et al., 2023b), Llama 3 (Grattafiori et al., 2024; Meta AI, 2024a;b)), are effective but often computationally expensive. To address efficiency bottlenecks due to quadratic complexity in long-context, methods like sparse attention (Beltagy et al., 2020) or alternative architectures like Mamba (Gu & Dao, 2024) have emerged, typically requiring training from scratch. In contrast, SWAN-GPT offers a novel architectural solution designed for inherent length extrapolation. Our hybrid design, combining efficient local SWA-RoPE layers with global NoPE layers, uniquely enables the adaptation of existing pre-trained models with minimal continued pre-training, distinguishing it from models like Gemma (Team & et al., 2024; 2025) which retain RoPE globally, and concurrent interleaving strategies (Yang et al., 2025b) that lack our dynamic attention scaling. See Appendix G for a more comprehensive discussion.

5. Conclusion

We introduced SWAN-GPT, a decoder-only transformer architecture that achieves robust length extrapolation without specialized long-context training. By interleaving NoPE and SWA-RoPE layers with dynamic attention scaling, our approach maintains consistent performance on sequences substantially longer than those seen during training. Our mechanistic analysis revealed this hybrid architecture creates a synergistic effect where SWA-RoPE layers provide stable positional grounding that relieves NoPE layers from developing brittle positional representations. We also demonstrated that existing pre-trained models can be efficiently adapted to the SWAN architecture through continued pre-training, offering a practical, cost-effective path for upgrading deployed models to handle significantly longer contexts without performance degradation on standard benchmarks. This approach shifts away from training directly on increasingly longer sequences, providing a more computationally efficient path toward long-context language modeling.

Impact Statement

This paper presents work whose goal is to advance the field of large language models by improving their efficiency and scalability for long-context processing. The potential societal consequences, both beneficial (e.g., enhanced comprehension and reasoning over significantly longer contexts) and challenging (e.g., enabling the creation of more sophisticated and extended deceptive or harmful content), are largely extensions of those already associated with highly capable language models. As such, while these extended capabilities warrant the same ongoing ethical considerations as the broader field, this work does not introduce unique societal impacts that we feel must be specifically highlighted beyond those general discussions.

References

An, C., Huang, F., Zhang, J., Gong, S., Qiu, X., Zhou, C., and Kong, L. Training-free long-context scaling of large language models, 2024. URL https://arxiv.org/abs/2402.17463.

Bavarian, M., Jun, H., Tezak, N., Schulman, J., McLeavey, C., Tworek, J., and Chen, M. Efficient training of language models to fill in the middle. *arXiv:2207.14255*, 2022.

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document Transformer. *arXiv: 2004.05150*, 2020.

bloc97. Dynamically scaled rope further increases performance of long context llama with zero fine-tuning. Reddit post, July 2023a. URL https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/.

bloc97. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. Reddit post, June 2023b. URL https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

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

Chen, Y., Qian, S., Tang, H., Lai, X., Liu, Z., Han, S., and Jia, J. Longlora: Efficient fine-tuning of long-context large language models, 2024a. URL https://arxiv.org/abs/2309.12307.

Chen, Z., Chen, Q., Qin, L., Guo, Q., Lv, H., Zou, Y., Che, W., Yan, H., Chen, K., and Lin, D. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. *arXiv:2409.01893*, 2024b.

Chi, T.-C., Fan, T.-H., Chen, L.-W., Rudnicky, A. I., and Ramadge, P. J. Latent positional information is in the self-attention variance of Transformer language models without positional embeddings. *arXiv: 2305.13571*, 2023.

DeepSeek-AI, Liu, A., and et.al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

Fu, Y. Challenges in deploying long-context transformers: A theoretical peak performance analysis, 2024. URL https://arxiv.org/abs/2405.08944.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 2024. URL https://zenodo.org/records/12608602.

Gao, T., Wettig, A., Yen, H., and Chen, D. How to train long-context language models (effectively), 2025. URL https://arxiv.org/abs/2410.02660.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In *EMNLP*, 2021.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv:2407.21783*, 2024.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv: 2312.00752*, 2024.

Han, C., Wang, Q., Peng, H., Xiong, W., Chen, Y., Ji, H., and Wang, S. Lm-infinite: Zero-shot extreme length generalization for large language models, 2024. URL https://arxiv.org/abs/2308.16137.

Harper, E. et al. NeMo: a toolkit for Conversational AI and Large Language Models, 2019. URL https://github.com/NVIDIA/NeMo.

Haviv, A., Ram, O., Press, O., Izsak, P., and Levy, O. Transformer language models without positional encodings still learn positional information. In *EMNLP*, 2022.

Hooper, C., Kim, S., Mohammadzadeh, H., Mahoney, M. W., Shao, Y. S., Keutzer, K., and Gholami, A. Kvquant: Towards 10 million context length llm inference with kv cache quantization, 2024. URL https://arxiv.org/abs/2401.18079.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. RULER: What's the real context size of your long-context language models? In *COLM*, 2024.

Jin, H., Han, X., Yang, J., Jiang, Z., Liu, Z., Chang, C.-Y., Chen, H., and Hu, X. Llm maybe longlm: Self-extend llm context window without tuning, 2024. URL https://arxiv.org/abs/2401.01325.

Kazemnejad, A., Padhi, I., Natesan Ramamurthy, K., Das, P., and Reddy, S. The impact of positional encoding on length generalization in Transformers. *NeurIPS*, 2023.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Liu, X., Yan, H., Zhang, S., An, C., Qiu, X., and Lin, D. Scaling laws of rope-based extrapolation. *arXiv:2310.05209*, 2024.

Liu, X., Li, R., Huang, M., Liu, Z., Song, Y., Guo, Q., He, S., Wang, Q., Li, L., Liu, Q., Zhou, Y., Huang, X., and Qiu, X. Thus spake long-context large language model. *arXiv:2502.17129*, 2025.

Lv, K., Liu, X., Guo, Q., Yan, H., He, C., Qiu, X., and Lin, D. Longwanjuan: Towards systematic measurement for long text quality, 2024. URL https://arxiv.org/abs/2402.13583.

Meta AI. Introducing meta llama 3: The most capable openly available llm to date. Blog Post, April 2024a. URL https://ai.meta.com/blog/meta-llama-3/.

Meta AI. Introducing meta llama 3.1: The most capable and versatile openly available models to date. Blog Post, July 2024b. URL https://ai.meta.com/blog/meta-llama-3-1/.

NVIDIA, :, Blakeman, A., Basant, A., Khattar, A., Renduchintala, A., Bercovich, A., Ficek, A., Bjorlin, A., Taghibakhshi, A., Deshmukh, A. S., Mahabaleshwarkar, A. S., Tao, A., Shors, A., Aithal, A., Poojary, A., Dattagupta, A., Buddharaju, B., Chen, B., Ginsburg, B., Wang, B., Norick, B., Butterfield, B., Catanzaro, B., del Mundo, C., Dong, C., Harvey, C., Parisien, C., Su, D., Korzekwa, D., Yin, D., Gitman, D., Mosallanezhad, D., Narayanan, D., Fridman, D., Rekesh, D., Ma, D., Pykhtar, D., Ahn, D., Riach, D., Stosic, D., Long, E., Segal, E., Evans, E., Chung, E., Galinkin, E., Bakhturina, E., Dobrowolska, E., Jia, F., Liu, F., Prasad, G., Shen, G., Liu, G., Chen, G., Qian, H., Ngo, H., Liu, H., Li, H., Gitman, I., Karmanov, I., Moshkov, I., Golan, I., Kautz, J., Scowcroft, J. P., Casper, J., Seppanen, J., Lu, J., Sewall, J., Zeng, J., You, J., Zhang, J., Zhang, J., Huang, J., Xue, J., Huang, J., Conway, J., Kamalu, J., Barker, J., Cohen, J., Jennings, J., Parmar, J., Sapra, K., Briski, K., Chumachenko, K., Luna, K., Santhanam, K., Kong, K., Sivamani, K., Pawelec, K., Anik, K., Li, K., McAfee, L., Derczynski, L., Pavao, L., Vega, L., Voegtle, L., Bala, M., de Melo, M. R., Sreedhar, M. N., Chochowski, M., Kliegl, M., Stepniewska-Dziubinska, M., Le, M., Novikov, M., Samadi, M., Andersch, M., Evans, M., Martinez, M., Chrzanowski, M., Ranzinger, M., Blaz, M., Smelyanskiy, M., Fawzy, M., Shoeybi, M., Patwary, M., Lee, N., Tajbakhsh, N., Xu, N., Rybakov, O., Kuchaiev, O., Delalleau, O., Nitski, O., Chadha, P., Shamis, P., Micikevicius, P., Molchanov, P., Dykas, P., Fischer, P., Aquilanti, P.-Y., Bialecki, P., Varshney, P., Gundecha, P., Tredak, P., Karimi, R., Kandu, R., El-Yaniv, R., Joshi, R., Waleffe, R., Zhang, R., Kavanaugh, S., Jain, S., Kriman, S., Lym, S., Satheesh, S., Muralidharan, S., Narenthiran, S., Anandaraj, S., Bak, S., Kashirsky, S., Han, S., Acharya, S., Ghosh, S., Sreenivas, S. T., Clay, S., Thomas, S., Prabhumoye, S., Pachori, S., Toshniwal, S., Prayaga, S., Jain, S., Das, S., Kierat, S., Majumdar, S., Han, S., Singhal, S., Niverty, S., Alborghetti, S., Panguluri, S., Bhendigeri, S., Akter, S. N., Migacz, S., Shiri, T., Kong, T., Roman, T., Ronen, T., Saar, T., Konuk, T., Rintamaki, T., Poon, T., De, U., Noroozi, V., Singh, V., Korthikanti, V., Kurin, V., Ahmad, W. U., Du, W., Ping, W., Dai, W., Byeon, W., Ren, X., Xu, Y., Choi, Y., Zhang, Y., Lin, Y., Suhara, Y., Yu, Z., Li, Z., Li, Z., Zhu, Z., Yang, Z., and Chen, Z. Nemotron-h: A family of accurate and efficient hybrid mamba-transformer models, 2025. URL https://arxiv.org/abs/2504.03624.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., He, X., Hou, H., Lin, J., Kazienko, P., Kocon, J., Kong, J., Koptyra, B., Lau, H., Mantri, K. S. I., Mom, F., Saito, A., Song, G., Tang, X., Wang, B., Wind, J. S., Wozniak, S., Zhang, R., Zhang, Z., Zhao,

Q., Zhou, P., Zhou, Q., Zhu, J., and Zhu, R.-J. Rwkv: Reinventing rnns for the transformer era, 2023a. URL https://arxiv.org/abs/2305.13048.

Peng, B., Quesnelle, J., Fan, H., and Shippole, E. Yarn: Efficient context window extension of large language models. *arXiv:2309.00071*, 2023b.

Shen, G., Wang, Z., Delalleau, O., Zeng, J., Dong, Y., Egert, D., Sun, S., Zhang, J., Jain, S., Taghibakhshi, A., Ausin, M. S., Aithal, A., and Kuchaiev, O. Nemo-aligner: Scalable toolkit for efficient model alignment, 2024.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020. URL https://arxiv.org/abs/1909.08053.

Su, D., Kong, K., Lin, Y., Jennings, J., Norick, B., Kliegl, M., Patwary, M., Shoeybi, M., and Catanzaro, B. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. *arXiv:2412.02595*, 2024.

Su, J. Rectified rotary position embeddings. https://github.com/bojone/rerope, 2023.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv:2104.09864*, 2023.

Team, G. and et al. Gemma: Open models based on Gemini research and technology. *arXiv: 2403.08295*, 2024.

Team, G. and et al. Gemma 3 technical report. *arXiv: 2503.19786*, 2025.

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

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Xiong, W., Liu, J., Molybog, I., Zhang, H., Bhargava, P., Hou, R., Martin, L., Rungta, R., Sankararaman, K. A., Oguz, B., Khabsa, M., Fang, H., Mehdad, Y., Narang, S., Malik, K., Fan, A., Bhosale, S., Edunov, S., Lewis, M., Wang, S., and Ma, H. Effective long-context scaling of foundation models, 2023. URL https://arxiv.org/abs/2309.16039.

Yang, A., Yu, B., Li, C., Liu, D., Huang, F., Huang, H., Jiang, J., Tu, J., Zhang, J., Zhou, J., et al. Qwen2. 5-1m technical report. *arXiv:2501.15383*, 2025a.

Yang, B., Venkitesh, B., Talupuru, D., Lin, H., Cairuz, D., Blunsom, P., and Locatelli, A. Rope to nope and back again: A new hybrid attention strategy, 2025b. URL https://arxiv.org/abs/2501.18795.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. Big bird: Transformers for longer sequences. *arXiv: 2007.14062*, 2021.

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

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

A. Ablations

To investigate the impact of different hybrid attention configurations on length extrapolation capabilities, we conducted an ablation study using models with 0.5B parameters. Each model consisted of 24 transformer decoder layers, with 16 attention heads per layer, 1024 hidden units, and a feed-forward dimension of 4096. We trained these models on a 350B token dataset using the AdamW optimizer, with a global batch size of 4096. We employed a cosine decay learning rate schedule that peaked at $3e^{-3}$ after 2000 warmup steps. All sliding window attention layers used a window size of 512 tokens with RoPE. For hybrid attention models we maintained a consistent 3:1 ratio between local (sliding window) and global attention layers and used attention scaling during inference (though we include a control without attention scaling).

A.1. Model Configurations

Below is a brief description of each of the models: **local only** - Implements sliding window attention across all layers.

global only (RoPE) - Standard transformer language model utilizing global attention with RoPE across all layers.

global only (NoPE) - Implements global attention with NoPE across all layers.

global_start - Begins with a global NoPE layer followed by three consecutive sliding window layers, repeating this pattern throughout. For inference, we additionally evaluate a version without attention scaling to establish a baseline.

local_start - Begins with three sliding window layers followed by a global NoPE layer, repeating this pattern throughout.

all_global_first - Concentrates all six global NoPE layers in the first positions, followed by sliding window layers.

all_local_first - Places all sliding window layers first, followed by six global NoPE layers.

A.2. Results

Table 5 shows results for the NIAH task from the RULER benchmark (Hsieh et al., 2024).[1] Among the baseline non-hybrid attention models, the **local only** model struggles to maintain high NIAH scores beyond its local window size (512), despite being trained on sequences of length 1k. However, unlike the **global only** attention baselines (RoPE and NoPE), which completely fail beyond the training distribution, the **local only** model demonstrates a modest capacity for length extrapolation. In contrast, all hybrid attention variants show substantial improvements in generalizing beyond the training length.

When comparing the hybrid variants we find that interspersing global and local attention layers yields superior performance compared to grouping them together, as evidenced by the relatively poor performance of both **all_global_first** and **all_local_first** configurations. In particular, our best-performing model (**global_start** achieves exceptional NIAH scores (> 0.9) at context lengths of 16k — 16 times the context length seen during training. It can also maintain robust performance (NIAH score > 0.7) even at 32k tokens, representing a 32-fold length extrapolation.

The critical role of attention scaling is demonstrated by our control experiment with **global_start (no scale)**. While both scaled and unscaled variants maintain strong performance up to 2k tokens, their behaviors diverge dramatically at longer contexts. The unscaled version shows rapid performance degradation beyond 4k tokens, dropping from 0.820 to 0.171 at 8k tokens and essentially failing (0.005) at 16k tokens. In contrast, the scaled version maintains exceptional performance at 8k tokens (0.957) and continues to achieve strong results at 16k tokens (0.907), and even maintains moderately good results at 32k tokens. This stark difference in length generalization — 4-fold extrapolation without scaling versus 32-fold with scaling — establishes attention scaling as a crucial mechanism for effective inference beyond the training length distribution. The graceful performance decline of the scaled model, compared to the abrupt deterioration of its unscaled counterpart, suggests that attention scaling helps maintain the model's ability to capture long-range dependencies even at extreme sequence lengths.

B. Architecture & Training

Table 6 shows model configuration for SWAN-1B and SWAN-8B models. Both RoPE-GPT-1B and SWAN-GPT-1B are trained from scratch with a batch size of 6M tokens (at 8k sequence length) with peak LR of 3e-3 for 1T tokens. We performed CPT for SWAN-8B with 32k sequence length and 6M token batch size at constant LR of 1e-5 for 300B tokens and ramped down to a LR of 5e-8 over another 15B tokens. Post-training for SWAN-8B model was performed in two stages. The first stage focused on a math and code blend with constant LR of 5e-6 followed by a second stage of general SFT at a constant LR of 1e-6. CPT was performed using Megatron-LM (Shoeybi et al., 2020) where as post-training used NeMo (Harper et al., 2019) and NeMo-Aligner (Shen et al., 2024). Megatron-LM is distributed under Apache 2.0 and MIT licenses, where as NeMo and NeMo Aligner are covered by Apache 2.0 license. Training used NVIDIA H100 80GB GPUs, with CPT consuming $\approx 18k$ GPU hours and post-training consuming $\approx 19k$ GPU hours.

[1] For simplicity we only evaluate the single NIAH task.

Model	512	1k	2k	4k	8k	16k	32k
local only	1.000	0.601	0.285	0.127	0.057	0.022	0.010
global only (RoPE)	1.000	0.985	0.000	0.000	0.000	0.000	0.000
global only (NoPE)	1.000	1.000	0.000	0.000	0.000	0.000	0.000
global_start (no scale)	1.000	1.000	0.983	0.820	0.171	0.005	0.003
global_start	1.000	1.000	0.999	0.998	0.957	0.907	0.702
local_start	1.000	1.000	0.999	0.895	0.808	0.725	0.530
all_global_first	1.000	0.599	0.316	0.113	0.044	0.017	0.010
all_local_first	1.000	1.000	0.993	0.564	0.183	0.057	0.027

Table 5: NIAH scores across different context lengths for various SWAN configurations.

Parameter	SWAN-1B	SWAN-8B
n_{layers}	24	32
d_{model}	1536	4096
n_{heads}	16	32
d_{head}	96	128
RoPE base	1,000,000	1,000,000
Normalization	RMSNorm	RMSNorm
global:local	1:3	1:3
SWA size	512	512

Table 6: Architecture details for SWAN-1B and SWAN-8B models.

C. Data

For details about the pre-training data, please refer to Section 2.2 of NVIDIA et al. (2025) and Su et al. (2024). Math and code focussed stage-1 SFT consisted of $\approx$ $670k$ sequences, with each sequence upto 32k tokens. A more general instruction-following stage-2 consisted of $\approx$ $200k$ sequences, with each sequence upto 32k tokens. In both stages, longer sequences were obtained by concatenating `(prompt, response)` pairs and were of the form `[(prompt-1, response-1), (prompt-2, response-2), ..., (prompt-N, response-N)]`.

D. Stabilizing Implicit Position Encodings for Robust Length Extrapolation

A key question in our investigation is understanding why the NoPE layers within our SWAN architecture demonstrate substantially more robust length extrapolation capabilities compared to identical layers within a model built purely of NoPE layers.

Despite the absence of explicit positional encoding, prior work has demonstrated that trained NoPE models implicitly learn to predict token positions after processing through a few layers (Chi et al., 2023). This implicit position embedding emerges from the autoregressive nature of decoder-only models, where tokens later in the sequence have access to more context than earlier tokens, creating distinct distributions at different positions. These distributional differences enable NoPE models to infer positional information and incorporate it into their predictions (Chi et al., 2023).

However, standard NoPE models exhibit poor robustness to sequences exceeding the training length, with performance degrading rapidly beyond the training boundary. In our SWAN architecture, the interleaved SWA-RoPE layers appear to relieve NoPE layers from developing the brittle position encodings typically seen in pure NoPE implementations, resulting in more robust processing of longer sequences.

To test these hypotheses, we conducted experiments with both pure NoPE and SWAN models trained on sequences of 1024 tokens and evaluate them on sequences of 2048 tokens. We employed two complementary analysis techniques: (1) position prediction probes to quantitatively measure positional information in model representations, and (2) attention pattern visualization to examine how attention mechanisms behave when processing sequences beyond training length.

D.1. Position Prediction Probes

To provide evidence for our hypothesis, we trained probes that predict token positions from token embeddings. We evaluated these probes on held-out tokens from positions both within and beyond the models' training range. Figure 2 shows predictions from eight different probes, each trained with tokens sampled from ranges demarcated by dashed lines. Each of the four subplots shows results from two probes - one trained on NoPE model embeddings (blue) and one on SWAN model embeddings (red) - with each probe trained on tokens from different context regions demarcated

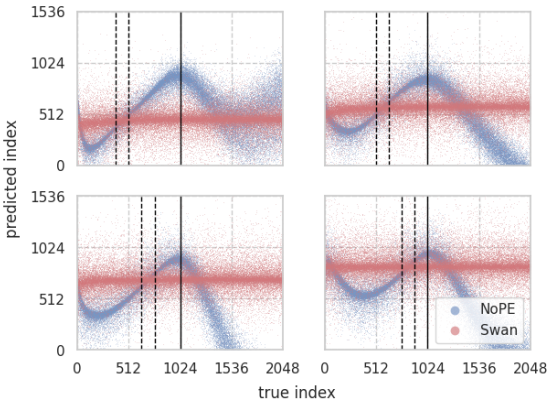


Figure 2: Predictions of token indices by 8 different probes. Each probe is trained with tokens from one model and different context regions (demarcated by dashed lines). Probes on NoPE models (blue) extrapolate correctly up until the maximum NoPE training length (solid line). Probes on SWAN (red) are not predictive of token indices.

by dashed lines.

For pure NoPE models (blue points), the probe predictions extrapolate well up to the boundary of the model's training range (solid black line). However, probes cease to be predictive beyond this boundary. Probes trained in different sub-regions all fail at the same location, consistent with the position prediction mechanism failing beyond the training range. In contrast, position probes trained on SWAN's NoPE layers (red points) show little positional information across all sequence positions. These layers do not develop the brittle position encoding seen in pure NoPE models. This supports our hypothesis that the interleaved SWA-RoPE layers free the NoPE layers from tracking absolute positions, allowing them to focus on integrating information across arbitrary distances while SWA-RoPE layers handle local positional structure.

D.2. Attention Pattern Analysis

To further investigate this phenomenon we examine the average attention values at different token positions for different sequence lengths. We average the probability scores (attention scores post soft-max) over all heads and over a set of validation batches. We randomize the token order in order to remove the effect of the correlation structure present in natural language.

Figure 3a shows the average attention maps of the 6th layer in the NoPE model. For sequences longer than the training length (green), the model places roughly equal attention on all 256 tokens preceding the target. In contrast, for sequences within training range (orange and blue),it preferentially attends to the tokens closest to the target token. A model that properly extrapolates to longer sequences should maintain similar attention patterns for tokens close to the target token, regardless of sequence length. In contrast, Figure 3b shows the average attention maps of the 20th layer (6th NoPE layer) in our SWAN model. Unlike the pure NoPE model, SWAN's attention maps exhibit consistent patterns across sequences both within and beyond the training length.

These analyses support our hypothesis that interleaving SWA-RoPE layers fundamentally alters how NoPE layers process positional information. The use of positional embeddings in the SWA-RoPE layers appears to stabilize the representations in the NoPE layers, making them more robust to sequence length extrapolation. This suggests that SWAN's superior length extrapolation capability stems from the emergent properties of the interleaved architecture.

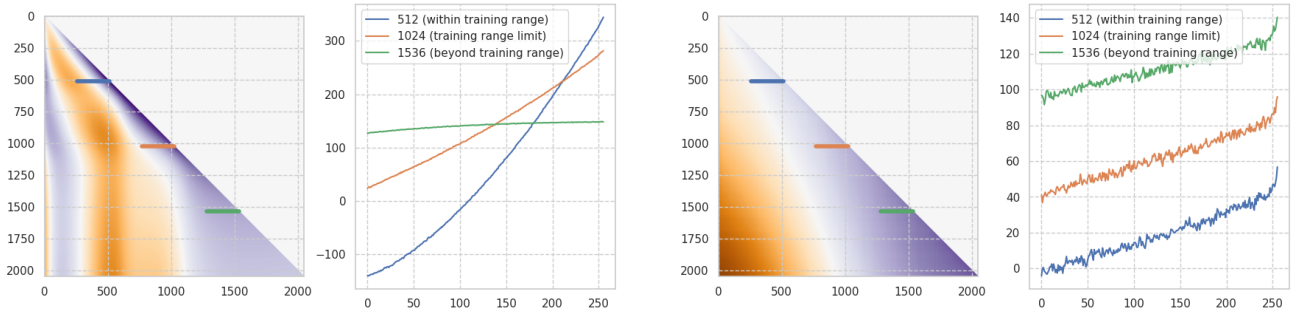
E. Dynamic Attention Scaling for Extended Context Processing

While our architecture demonstrates inherent sequence length extrapolation, we find that further performance improvements can be achieved through proper scaling of attention logits during inference. This scaling is particularly important for the global NoPE layers, which must effectively integrate information across arbitrary distances as sequence length increases.

Prior work has shown that RoPE-based models improve their performance on extended context lengths when the temperature of the attention logits is properly adjusted (Peng et al., 2023b). The SWA-RoPE layers in our SWAN architecture inherently handle longer sequences due to their local attention window. However, we hypothesize that the global attention NoPE layers may still require scaling to maintain performance at extended lengths.

For this analysis, we sampled 200 documents from the model's training distribution (each with at least 32K tokens) to maintain a consistent semantic distribution while extending context length beyond the original 1K tokens used during training. We partitioned each 32K-token context into 128-token windows and estimated a single optimal scaling factor per window by minimizing its perplexity over all 200 documents.

Figure 4 shows the empirically determined optimal scaling factors (black dots) across different positions in the 32K context. We find that a logarithmic scaling function $\log_a(a+n)$ (green line) provides an excellent fit to the empirical data. This function captures two key properties we observe – a natural growth rate that matches the data's progression, and a base scaling factor that never falls below 1.0, which is important for maintaining model stability at early positions. Interestingly, while prior work found that the YaRN scaling



(a) NoPE model (6th layer)

(b) SWAN model (20th layer)

Figure 3: Each panel shows attention averaged over all heads and validation records (left) and cross sections for sequence lengths of 512, 1024 (training range limit), and 1536 (extrapolation regime). The attention patterns of leading 256 tokens differ significantly: NoPE shows inconsistent patterns when extrapolating beyond training range, while SWAN maintains consistent decay patterns for sequences both within and beyond the training range.

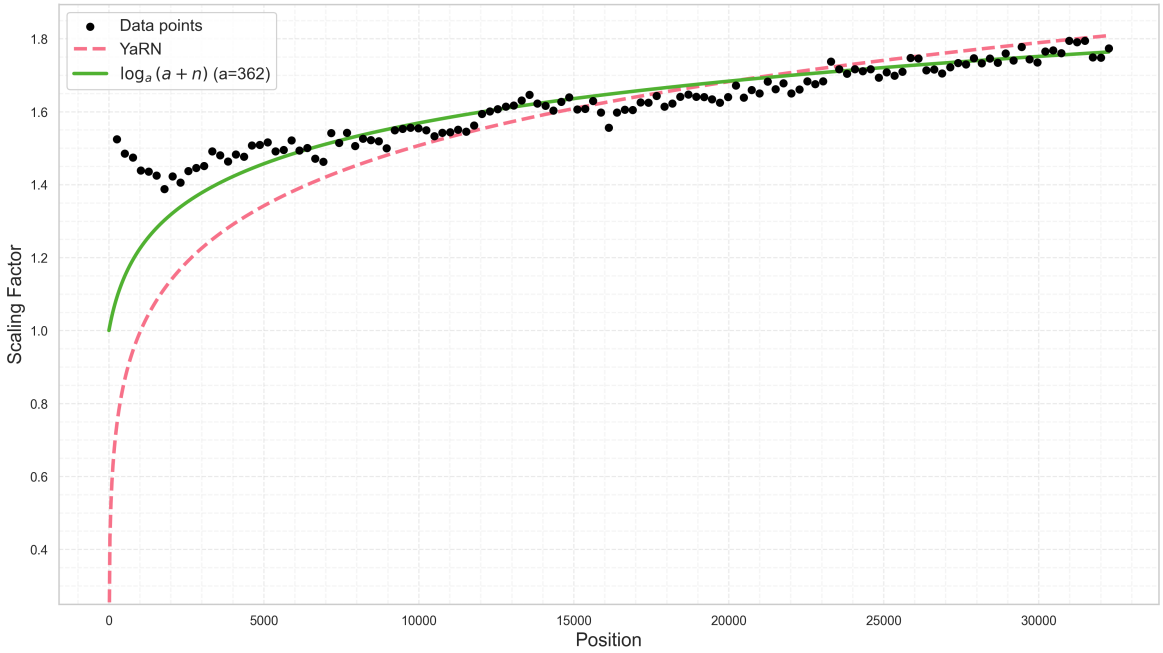


Figure 4: Estimates of optimal scaling factors (black) comparing the fit of our logarithmic scaling function vs. YaRN scaling. We find that YaRN scaling doesn't work as well for NoPE layers.

function (Peng et al., 2023b) works well for RoPE-based models, we observe that it (dashed pink line) fits poorly for the NoPE layers in our SWAN architecture, particularly in early positions where it significantly under-estimates the required scaling.

To validate our empirically determined scaling function, we compute perplexity on held-out documents from the PG19 dataset, using the same procedure described above. Figure 5 plots the perplexity at each location within the 32K token context, with and without our scaling function applied. Without scaling (blue), we observe a clear degradation in model performance on longer contexts. In contrast, our scaling (green points) allows the model to maintain better performance as measured by a lower and more stable perplexity value for the entire context length up to contexts 32 times longer than the training length ($1K$ tokens). This improved performance with scaling is further validated by our NIAH evaluation results in Table 5 in Appendix A, where we demonstrate that scaling improves NIAH scores from 0.171 to 0.957 at 8K context length and from 0.005 to 0.907 at 16K context length.

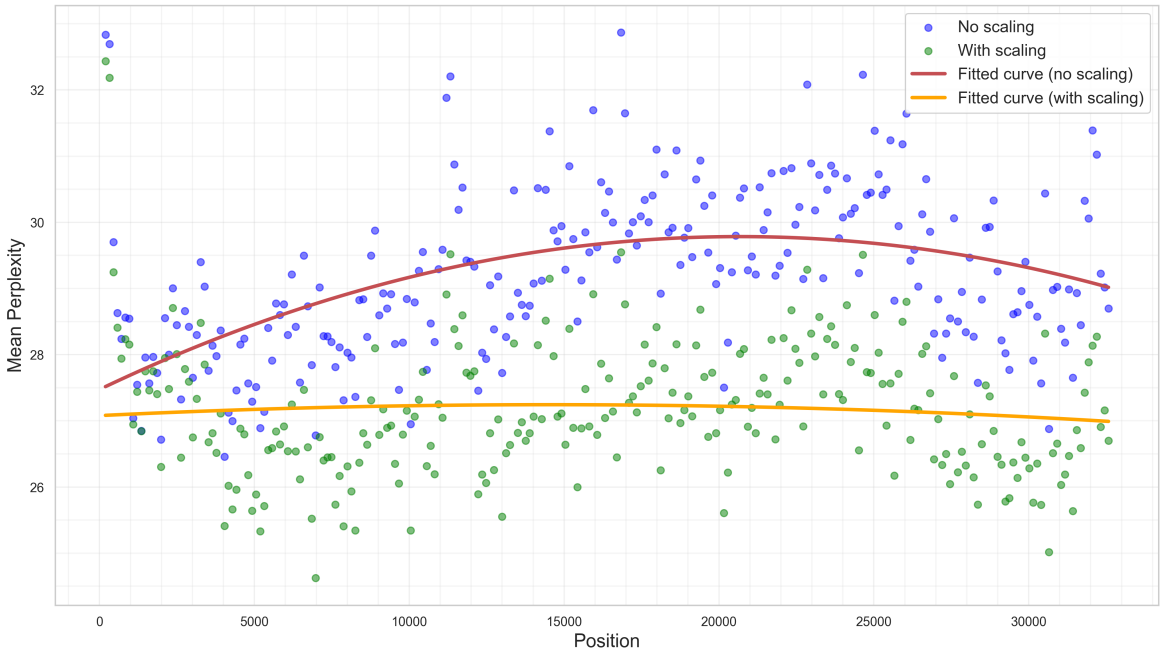


Figure 5: Held-out perplexity, with (green) and without (blue) logarithmic scaling. Without scaling, we see that perplexity scores degrade on longer contexts, whereas with scaling the performance is more stable.

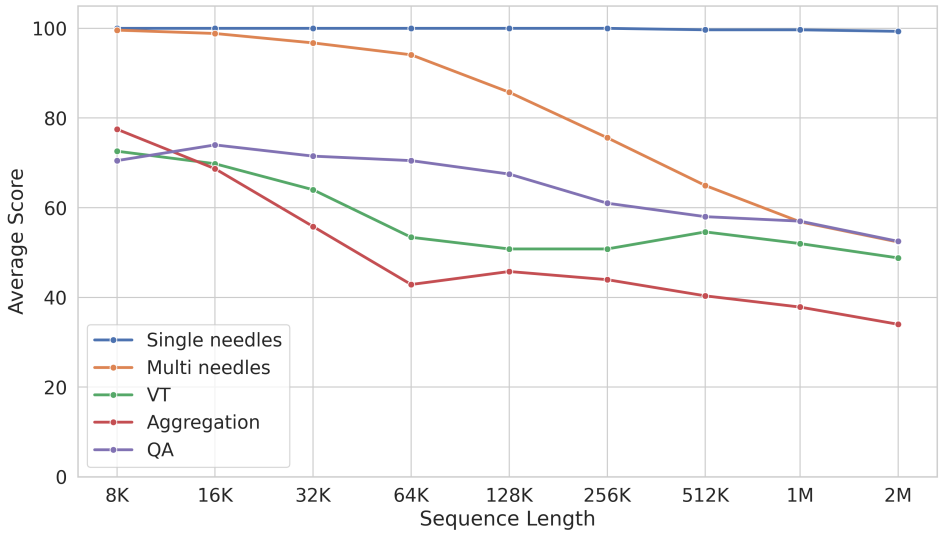


Figure 6: RULER scores split by task type for SWAN-8B model. VT:Variable Tracking, QA: question-Answering. SWAN-8B model shows near-perfect recall for single needle tasks upto sequence length of 2M (64 times of training sequence length).

F. Additional RULER Results

Figure 6 shows RULER scores split by task type (single needles, multi-needles, Variable Tracking (VT), Aggregation and Question-Answering (QA)) for SWAN-8B model upto 2M sequence length. SWAN model shows near-perfect recall for single needle tasks. Figure 7 shows passkey retrieval task performance of SWAN-8B model.

G. Related work

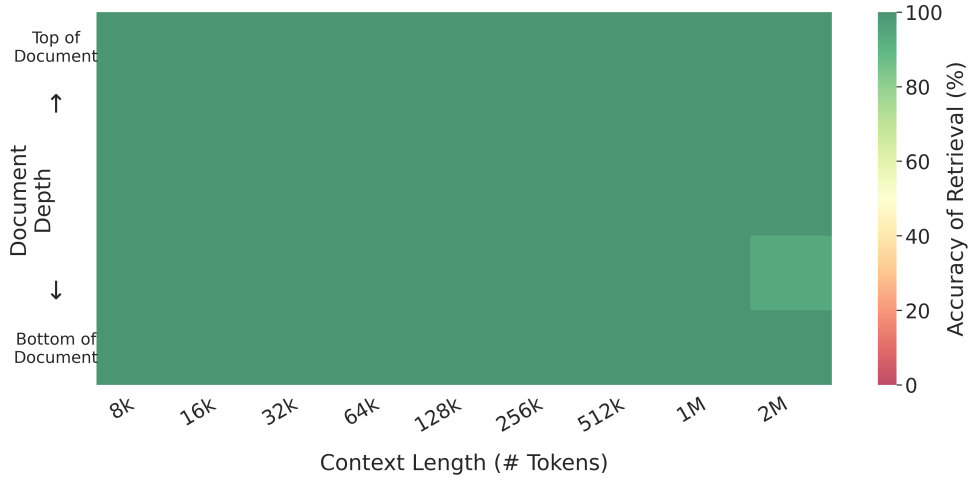


Figure 7: Passkey retrieval test on SWAN-8B model with documents upto 2M tokens. Results show that SWAN-8B model can accurately recall hidden numbers from documents upto 2M tokens.

Extending LLM context length to hundreds of thousands of tokens presents challenges across architecture, computation, and data quality (Lv et al., 2024; Gao et al., 2025; Liu et al., 2025). Our work addresses the architectural aspect through design choices enabling length extrapolation.

Several approaches extend context purely at inference time. For RoPE-based models, these include NTK-aware scaling (bloc97, 2023b;a) and Positional Interpolation (PI) (Chen et al., 2023), though these can degrade performance or require careful tuning (An et al., 2024). Recent methods modify attention mechanisms directly: ReRoPE (Su, 2023), SelfExtend (Jin et al., 2024), and Dual Chunk Attention (An et al., 2024). Others leverage attention patterns through windowing approaches like StreamingLLM (Xiao et al., 2024) and LM-Infinite (Han et al., 2024). SWAN-GPT differs by addressing fundamental architectural limitations through hybrid design rather than post-hoc modifications.

Training-based approaches include PI (Chen et al., 2023) and YaRN (Peng et al., 2023b), which work best after continued pre-training. While effective, CPT on longer sequences (Xiong et al., 2023) becomes prohibitively expensive for large models. Parameter-efficient methods like LongLoRA (Chen et al., 2024a) help but still require additional training. State-of-the-art models like Llama 3 (Grattafiori et al., 2024; Meta AI, 2024a;b) and Qwen2.5 (Yang et al., 2025a) achieve long-context capabilities through extensive pre-training with varied sequence lengths. In contrast, SWAN-GPT achieves length extrapolation without long-context specific training and can adapt existing models with minimal continued pre-training.

The quadratic complexity of self-attention poses efficiency bottlenecks for long contexts (Kwon et al., 2023; Fu, 2024; Liu et al., 2025). Sparse attention mechanisms in Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2021) address this by limiting attention patterns. Alternative architectures like Mamba (Gu & Dao, 2024) and RWKV (Peng et al., 2023a) achieve near-linear complexity but require training from scratch. SWAN-GPT's hybrid design improves efficiency: SWA-RoPE layers use efficient local attention, while global NoPE layers can benefit from techniques like Multi-head Latent Attention (DeepSeek-AI et al., 2024). Additional KV cache optimizations (Zhang et al., 2023; Xiao et al., 2024; Zhang et al., 2024; Hooper et al., 2024; Liu et al., 2025) can complement our approach.

The Gemma family (Team & et al., 2024; 2025) similarly uses sliding window and global attention but retains RoPE throughout, unlike our strategic omission of positional encodings in global layers. Concurrent work (Yang et al., 2025b) explores similar layer interleaving but lacks our dynamic attention scaling mechanism and mechanistic analysis. We uniquely demonstrate efficient adaptation of existing pre-trained models with minimal continued pre-training.