Fractional Rotation, Full Potential?
Investigating Performance and Convergence of Partial RoPE

Anonymous ACL submission

Abstract

Rotary Positional Embedding (RoPE) is widely used in transformers to encode relative positional information, yet the effect of applying RoPE to only a subset of hidden dimensions remains underexplored. This design choice can substantially reduce memory usage, especially at long context lengths. We show that partial RoPE achieves up to 10× memory savings over standard RoPE cache while maintaining comparable final loss. We systematically study its impact on training dynamics and convergence across architectures, model sizes, sequence lengths, and datasets. Our results reveal that: (1) applying RoPE to only a small fraction of dimensions (around 10%) matches the convergence of full RoPE; (2) these trends remain consistent across settings; and (3) NoPE (No Positional Encoding) can produce loss spikes, which are mitigated by minimal RoPE usage or QK-Norm application. Overall, our findings highlight partial RoPE as an effective trade-off between efficiency and stability.

1 Introduction

Transformers use positional encodings to capture the order of tokens within a sequence, since the other parts of the model themselves are permutation-invariant. Rotary Positional Embedding (RoPE) (Su et al., 2023) has emerged as a leading method for encoding relative positions directly in the query-key interactions in self-attention (Vaswani et al., 2017) computations. RoPE is favored in modern decoder-only language models for its strong empirical performance and ability to generalize to sequences longer than those seen during training. It is widely adopted across architectures, either as the sole positional encoding or alongside alternatives such as No Positional Encoding (NoPE) (Kazemnejad et al., 2023).

Despite the widespread adoption of RoPE, a fundamental design choice remains largely unexplored. What is the minimum fraction of hidden dimensions within each attention head that needs to undergo the rotary transformation? This parameter varies considerably across major model families, highlighting a lack of consensus on best practices. Early implementations in models like GPT-J (Wang and Komatsuzaki, 2021) and GPT-NeoX (Black et al., 2022), and later Pythia (Biderman et al., 2023), adopted a partial approach, applying RoPE to only 25% of the dimensions. In contrast, the LLaMA series (Touvron et al., 2023a,b; Grattafiori et al., 2024; Meta AI, 2025) and most of the Qwen series (Bai et al., 2023; Yang et al., 2024; Qwen et al., 2025; Yang et al., 2025a) applied the transformation to all dimensions. Interestingly, Qwen3-Next model moved to a 25% application, a change claimed to improve extrapolation to longer sequences (Qwen Team, 2025). Other models have explored intermediate values; Nemotron-4-340B (Nvidia et al., 2024) uses 50% while Phi-2 (Javaheripi et al., 2023) uses 40%.

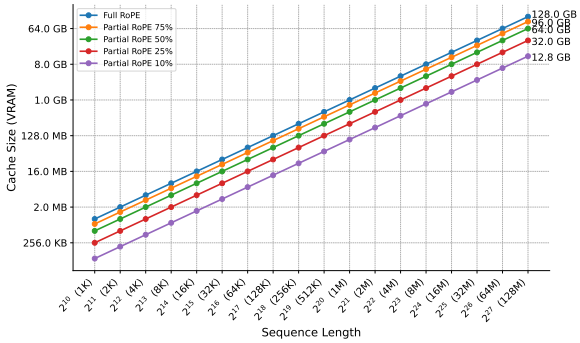


Figure 1: Estimated RoPE cache memory usage as a function of sequence length. Applying RoPE to only a fraction of hidden dimensions substantially reduces cache size, yielding significant savings at long context lengths and in resource-constrained settings such as edge devices. More details are described in Appendix D.

This wide variation in implementation highlights a gap in the literature, as no prior work has systematically studied how partial RoPE affects convergence, training dynamics, or efficiency. Unlike other arch. choices, it is inconsistently applied, rarely ablated and not well supported in most frame-

works. This is particularly relevant for long-context settings, where applying RoPE to a small fraction (e.g., 10%) can reduce cache memory by an order of magnitude (Fig. 1). A systematic study of this parameter can therefore improve understanding and enable more memory-efficient designs.

2 Related Work

RoPE and Partial RoPE. RoPE (Su et al., 2023) is widely used in transformer models due to its simplicity, strong length generalization, and compatibility with modern attention variants. Prior work has studied several aspects of its design, including hybrid use with sliding-window attention or selective omission across layers to improve efficiency while maintaining performance (Kazemnejad et al., 2023; Cohere et al., 2025; Meta AI, 2025). Other studies analyze rotary base and scaling choices and their effects on long-context extrapolation and convergence (Men et al., 2024; Yang et al., 2025b), as well as underlying geometric properties (Barbero et al., 2025) and numerical precision issues in long-context regimes (Wang et al., 2024).

However, the question of how many hidden dimensions should be rotated remains largely underexplored. The only explicit discussion of partial RoPE appears in (Black et al., 2022), which, based on small-scale experiments[1], suggests that rotating 25% of dimensions offers a good performance–efficiency trade-off. This heuristic has since influenced several model designs.

Long Context Models. There has been rapid progress in extending transformer context lengths, exemplified by Gemini 1.5 (Google, 2024) and Llama 4 Scout (Meta AI, 2025), which report contexts up to 10M tokens, as well as early systems reaching 100M tokens (Magic, 2024). At these scales, implementation details become critical: the RoPE cache, typically negligible, grows linearly with sequence length and can become a significant memory bottleneck. An extended related work section is provided in Appendix B.

3 Experimental Setup

We pretrain several models from scratch to examine how the fraction of each attention head's hidden state receiving rotary positional embeddings affects model performance. Specifically, we apply RoPE to the first $K\%$ of dimensions within each head and evaluate fractions 0% (NoPE), 10%, 25%, 50%, 75%, and 100% (full RoPE), to observe their effect on loss convergence. Additionally, we include experiments with the minimal feasible application, corresponding to just two channels per head, which represents approximately 1% for Pythia-1B (head dimension 256) and 4% for LLaMA-1B (head dimension 64). Further details are outlined in App. A.

We evaluate two transformer architectures: (1) a sequential attention design, implemented using Llama-3.2-1B and Llama-3.1-8B (Meta, 2024; Grattafiori et al., 2024), and (2) a parallel attention design, implemented using Pythia-1B (Biderman et al., 2023). The primary training is conducted on the FineWeb dataset (Penedo et al., 2024), with additional experiments on FineWeb-Edu (Lozhkov et al., 2024) to assess effect of dataset quality. We use the officially released 100B-token subsets and process all inputs with the Pythia tokenizer.

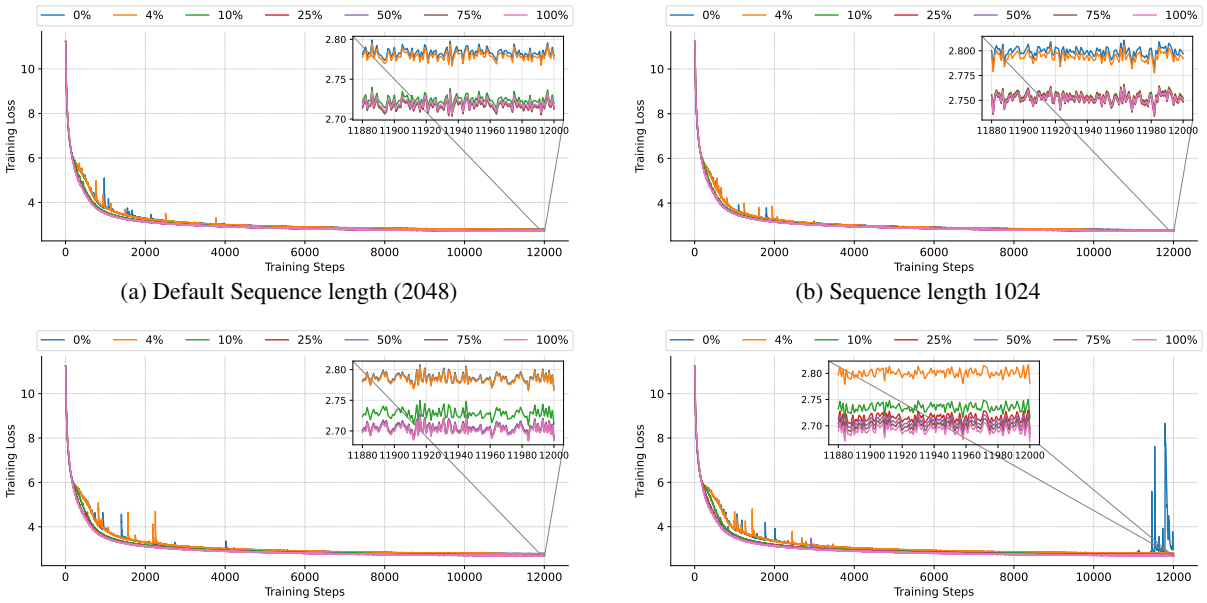
In addition to loss, we evaluate models using EleutherAI's LM Evaluation Harness (Gao et al., 2023). We adopt the same benchmarks used for the Pythia suite (Biderman et al., 2023), as they are well suited for non-instruction-tuned models, cover diverse task types, and remain informative at the 100B-token training scale (Wei et al., 2026). This set is further supplemented with PubMedQA (Jin et al., 2019). Additional training and evaluation details are presented in Appendix C.

4 Partial RoPE Analysis

RQ1: How does the fraction of hidden dimensions receiving RoPE influence model training dynamics? As shown in Fig. 2a, configurations using 10% or more RoPE exhibit nearly identical convergence behavior. By the end of training, two distinct convergence groups emerge: models with NoPE or with RoPE applied to only 2 channels (4%) converge to higher final losses, while those with 10% or more RoPE achieve similar and lower final losses. This indicates that applying RoPE to even a modest fraction of hidden dimensions is sufficient to match the convergence of full RoPE.

RQ2: How does the quality of pre-training data affect the optimal partial RoPE configuration? To explore this, we repeated the experiments using FineWeb-Edu, a higher quality dataset. FineWeb-Edu is derived from FineWeb by applying a classifier that filters for content with higher educational quality. The impact of data quality is

[1] We infer from public logs that GPT2-Small (124M parameters) trained on 82M tokens was used, whereas we evaluate 1B–8B models trained on over 100B tokens.



(a) Default Sequence length (2048)

(b) Sequence length 1024

(c) Sequence length 4096

(d) Sequence length 8192

Figure 2: Training loss trajectories on the FineWeb dataset for sequential models with varying RoPE configurations.

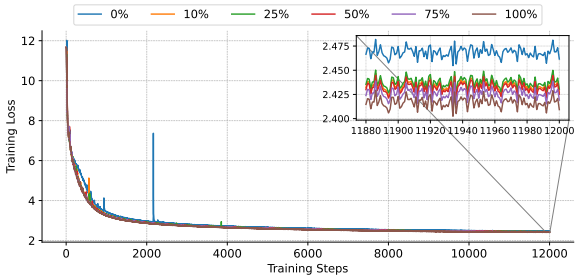


Figure 3: Training loss trajectories for the 8B model with varying Partial RoPE configurations.

evident in the higher final loss values across runs (see Figs. 2a and 5a). Nonetheless, similar convergence patterns are observed across both datasets.

RQ3: How does the training sequence length influence the optimal Partial RoPE configuration? To evaluate the effect of sequence length on Partial RoPE behavior, we repeat the experiments from RQ1 using sequence lengths of 1024, 4096, and 8192 tokens, which correspond to commonly used pretraining context window sizes. As shown in Fig. 2b, 2c, and 2d, the models exhibit similar convergence bands across configurations, indicating that the observed trends are largely consistent across different sequence lengths.

We observe a single notable exception in which a loss spike appears for the NoPE run at a sequence length of 8192; strategies to mitigate this behavior are discussed later. Additionally, as the sequence length increases, the 10% run begins to diverge slightly from the 25% and higher settings. However, this separation is smaller than the differences observed for the NoPE and 4% runs.

RQ4: How consistent are the effects of partial RoPE across different transformer block designs (sequential vs. parallel attention)? To investigate this, we trained models following the Pythia-1B architecture with parallel transformer blocks and observed two notable trends. The NoPE configuration (0%) fails to converge (Fig. 5b), a phenomenon we analyze further later. Excluding the NoPE run, two distinct convergence bands emerge, similar to those seen in sequential attention models, indicating that the trends hold across transformer block designs.

RQ5: How do the effects of partial RoPE change with model scale? To study scaling, we train a Llama-3-8B-like model on 100B FineWeb tokens. We again observe distinct convergence bands: NoPE forms a separate higher-loss cluster, while RoPE variants cluster together (Fig. 3). Relative to the 1B model at 2048 sequence length, RoPE configurations in the 8B model are slightly more dispersed, but overall trends remain consistent.

RQ6: Do benchmark evaluation results corroborate the loss-based analysis? Across 9 out of 10 benchmarks, all RoPE variants exhibit broadly similar multiple-choice question (MCQ) performance, with WSC as the only exception, where no single configuration consistently outperforms the others. In contrast, LAMBADA perplexity drops substantially when transitioning from $< 10\%$ to $\geq 10\%$ RoPE, while remaining relatively stable across higher-fraction variants (Fig. 4). Additional results are provided in Table 3 and 4. This is inline with the observations of Heineman et al. (2025),

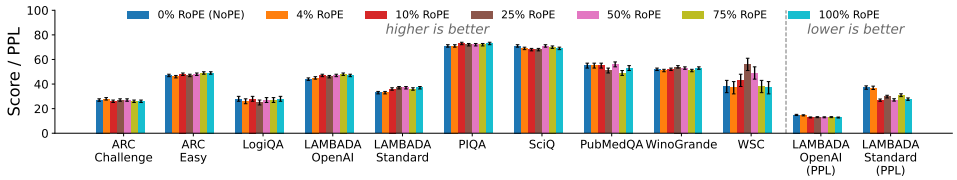


Figure 4: Benchmark performance across tasks for the Sequential 1B model trained on FineWeb (Seq. Len. 2048)

where perplexity was found to offer a stronger signal as well as our findings of distinct loss bands.

Takeaways. Our experiments reveal clear trends for partial RoPE. Applying RoPE to even a small fraction of hidden dimensions (10% or more) is enough to replicate the convergence behavior and final-loss performance of full RoPE. These patterns hold across datasets of varying quality, sequence lengths, and transformer block architectures, and remain consistent as model size scales from 1B to 8B parameters, though larger models exhibit slightly more variability across RoPE configurations. Evaluations beyond loss largely support these results: RoPE variants perform similarly on most MCQ benchmarks, while perplexity analyses show that configurations with 10% or more RoPE behave comparably and outperform lower-percentage variants. Overall, partial RoPE appears to be a promising optimization strategy.

Analyzing Loss Spikes with NoPE

Having shown that partial RoPE attains comparable convergence, we next analyze the two unrecoverable loss spikes observed in the NoPE configurations. Such loss spikes were not reported by Kazemnejad et al. (2023), who primarily studied smaller synthetic tasks with substantially lower model and dataset scales and did not consider parallel architectures. In contrast, our experiments show that these loss spikes emerge only at large training scales. They appear after tens of billions of tokens in parallel architectures, and only under similarly large training budgets combined with long context lengths in sequential architectures. To identify the potential root causes and mitigate these spikes, we systematically investigate the following for the parallel attention model.

Effect of Randomness: We first considered that the spikes could result from an unlucky ordering of training data or model initialization. However, repeating the experiments with multiple random seeds (which change initialization and data order) consistently produced similar spikes (Fig. 6a).

Effect of Learning Rate (LR): We next examine whether LR explains the behavior. We tested learning rates an order of magnitude higher and lower than the base (4×10^{-3} & 4×10^{-5}). With a lower LR, training converged to a higher loss, while a higher LR led to early spikes (Fig. 6b).

Effect of QK-Norm: Finally, we applied QK-Norm (Henry et al., 2020), a normalization technique known to stabilize training. More details about the implementation are outlined in Appendix C. With QK-Norm, the loss spikes disappear (Fig. 7a), suggesting that the underlying cause may be excessively large or spiky gradients and parameter magnitudes (Taylor, 2024; OLMo et al., 2025), both of which this normalization mitigates. Following its application, the NoPE configuration converged to a higher loss band, consistent with patterns observed in other RQs. Similar trends are also observed in the sequential attention model as outlined in Fig. 7b in Appendix F. *Based on this, we recommend that model trainers adopt QK-Norm as a precautionary measure against loss spikes or use partial RoPE to prevent the same.*

5 Conclusion

We presented a systematic empirical study of partial application of Rotary Positional Embeddings (RoPE) across hidden dimensions in large-scale transformer models. Our results show that even small RoPE fractions (as low as 10% of dimensions) can match full RoPE in convergence and final loss, while extremely low fractions or NoPE configurations slow training and can introduce pronounced loss spikes. These patterns remain consistent across datasets of varying quality, model sizes, and both sequential and parallel transformer architectures. We also find that while stabilization methods such as QK-Norm can reduce instability under NoPE, partial RoPE is a more effective and direct solution. Overall, partial RoPE offers a simple and effective way to reduce memory overhead (particularly for long-context and resource constrained settings) without sacrificing training stability or performance. Our findings provide practical guidance for transformer design and highlight partial RoPE as an underexplored but impactful direction for efficient positional encoding.

Limitations

As with any study involving pretraining, the design space is vast and each experiment incurs substantial computational cost. While it is infeasible to explore all possible combinations of architectures, model sizes, and datasets, we carefully select configurations that follow established best practices and evaluate them at scales consistent with prior work, allowing us to make claims we expect to generalize to larger settings.

We leave several directions for future work such as combining partial RoPE with NoPE for additional efficiency gains, studying scaling laws for partial RoPE, and exploring the interaction of partial RoPE with length extrapolation methods due to the substantial computational resources and training time each of these investigations would require.

References

Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, and 2 others. 2023. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch. https://www.github.com/eleutherai/gpt-neox. Version 2.0.0.

Anthropic. 2026. Introducing claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6. Accessed: 2026-05-24.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. 2025. Round and round we go! what makes rotary positional encodings useful? *Preprint*, arXiv:2410.06205.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

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

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Team Cohere, :, Aakanksha, Arash Ahmadian, Marwan Ahmed, Jay Alammar, Milad Alizadeh, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, Raphaël Avalos, Zahara Aviv, Sammie Bae, Saurabh Baji, Alexandre Barbet, Max Bartolo, Björn Bebensee, and 211 others. 2025. Command a: An enterprise-ready large language model. *Preprint*, arXiv:2504.00698.

Yufeng Du, Phillip Harris, Minyang Tian, Eliu A Huerta, Srikanth Ronanki, Subendhu Rongali, Aram Galstyan, and Hao Peng. 2026. Rope distinguishes neither positions nor tokens in long contexts, provably. *Preprint*, arXiv:2605.15514.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2023. A framework for few-shot language model evaluation.

Google DeepMind Google, Alphabet. 2024. Our next-generation model: Gemini 1.5. https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

David Heineman, Valentin Hofmann, Ian Magnusson, Yuling Gu, Noah A. Smith, Hannaneh Hajishirzi, Kyle Lo, and Jesse Dodge. 2025. Signal and noise: A framework for reducing uncertainty in language model evaluation. *Preprint*, arXiv:2508.13144.

Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. 2020. Query-key normalization for transformers. *Preprint*, arXiv:2010.04245.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro

Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, and 1 others. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Matt Gardner Johannes Welbl, Nelson F. Liu. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. *Preprint*, arXiv:2305.19466.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012*, Proceedings of the International Conference on Knowledge Representation and Reasoning, pages 552–561. Institute of Electrical and Electronics Engineers Inc. 13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012 ; Conference date: 10-06-2012 Through 14-06-2012.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Ilya Loshchilov and Frank Hutter. 2017. Sgdr: Stochastic gradient descent with warm restarts. *Preprint*, arXiv:1608.03983.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu: the finest collection of educational content.

Magic. 2024. 100m token context windows. https://magic.dev/blog/100m-token-context-windows.

Xin Men, Mingyu Xu, Bingning Wang, Qingyu Zhang, Hongyu Lin, Xianpei Han, and Weipeng Chen. 2024. Base of rope bounds context length. *Preprint*, arXiv:2405.14591.

Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

Meta AI. 2025. Llama 4: Multimodal intelligence. https://ai.meta.com/blog/llama-4-multimodal-intelligence/. Accessed: 2025-10-03.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. *Preprint*, arXiv:1710.03740.

Nvidia, :, Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, and 64 others. 2024. Nemotron-4 340b technical report. *Preprint*, arXiv:2406.11704.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. 2 olmo 2 furious. *Preprint*, arXiv:2501.00656.

OpenAI. 2026. Gpt-5.5 models documentation. https://developers.openai.com/api/docs/models/gpt-5.5. Accessed: 2026-05-24.

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

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qwen Team. 2025. Qwen3-next: Towards ultimate training & inference efficiency. https://qwen.ai/blog?from=research.latest-advancements-list&id=4074cca80393150c248e508aa62983f9cb7d27cd. Accessed: 2025-10-05.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Ross Taylor. 2024. Qk norm and the curious case of logit drift. https://rossjtaylor.com/blog/qk-norm-and-the-curious-case-of-logit-drift/. Accessed: 2025-10-03.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

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

Johnny Wei, Ameya Godbole, Mohammad Aflah Khan, Ryan Yixiang Wang, Xiaoyuan Zhu, James Flemings, Nitya Kashyap, Krishna P. Gummadi, Willie Neiswanger, and Robin Jia. 2026. Hubble: a model suite to advance the study of LLM memorization. In *The Fourteenth International Conference on Learning Representations*.

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

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Bowen Yang, Bharat Venkitesh, Dwarak Talupuru, Hangyu Lin, David Cairuz, Phil Blunsom, and Acyr Locatelli. 2025b. Rope to nope and back again: A new hybrid attention strategy. *Preprint*, arXiv:2501.18795.

Overview of Appendix Sections

- Appendix A: Rotary Dimension Allocation Details
- Appendix B: Extended Related Work
- Appendix C: Training and Evaluation Details
  - Training Details
  - Evaluation Details
- Appendix D: Computation of RoPE Cache Size
- Appendix E: References for Partial RoPE Usage
- Appendix F: Additional Results
- Appendix G: LLM Usage
- Appendix H: Artifact Release/Usage

A Rotary Dimension Allocation Details

We report the exact number of hidden dimensions to which RoPE is applied for each configuration. Because RoPE rotates hidden states in pairs, the number of rotated dimensions must be even. Accordingly, percentage-based specifications are rounded to the nearest valid even number of dimensions.

This adjustment only occurs for **Pythia**, which has a head dimension of 256. A nominal 10% corresponds to 25.6 dimensions and would be rounded to 25 by the training framework. Since RoPE requires pairwise rotation, we instead use 26 dimensions (10.2%) to ensure a valid pair count.

The resulting rotated dimensions for each model are listed in Tables 1.

Model (Head Dim)	0%	1%	4%	10% / 10.2%	25%	50%	75%	100%
Pythia-1B (256)	0	2	–	26	64	128	192	256
Llama-1B (64)	0	–	2	6	16	32	48	64
Llama-8B (128)	0	–	–	12	32	64	96	128

Table 1: Number of rotated dimensions used for each RoPE percentage configuration. Values are adjusted to ensure an even number of dimensions since RoPE rotates hidden states pairwise.

B Extended Related Work

RoPE and Partial RoPE. RoPE (Su et al., 2023) is widely used in transformer models because of its simplicity, strong length generalization, and compatibility with diverse attention mechanisms. Prior work has explored multiple aspects of RoPE design and integration. Several studies, as well as frontier models, omit RoPE in selected layers or combine it with sliding-window attention to improve efficiency while preserving performance (Kazemnejad et al., 2023; Cohere et al., 2025; Meta AI, 2025). Other work has examined how the rotary base and scaling strategies influence convergence and long-context extrapolation (Men et al., 2024; Yang et al., 2025b). Additional studies analyze the underlying mechanics of RoPE (Barbero et al., 2025), explore guarantees on RoPE's distinguishing ability between tokens (Du et al., 2026) and highlight precision-related issues that arise in long-context settings (Wang et al., 2024). Despite these efforts, the specific question of how many dimensions of the hidden state should undergo rotation has received little attention. To our knowledge, the only discussion on partial RoPE application appears in (Black et al., 2022), which suggested, based on small-scale experiments[2], that a 25% application offered a good trade-off between performance and efficiency. These findings have influenced design choices in multiple subsequent models.

Long Context Models. There has been sustained momentum toward training models with ever-larger context windows. Notable examples include Google's Gemini 1.5 family (Google, 2024) and Meta's Llama 4 Scout (Meta AI, 2025), both of which report context windows of up to 10 million tokens, as well as preliminary work by Magic demonstrating 100 million token contexts (Magic, 2024). Highly used frontier models also now boast context windows of a million tokens such as Opus 4.6 (Anthropic, 2026) and GPT-5.5 (OpenAI, 2026). At these extreme scales, previously negligible implementation details become critical. For instance, RoPE cache, which typically consumes minimal VRAM, can require substantial chunks of memory usage due to linear scaling with sequence length. This challenge is compounded in multi-GPU setups, where either replicating the cache on each device consumes redundant memory, or sharding it introduces communication overhead.

[2]On inspecting the public logs we were able to infer the authors used the GPT2-Small architecture (124M parameters) and trained on 82M tokens, in contrast we test multiple architectures at the 1B/8B scale and over 100B tokens

C Training and Evaluation Details

C.1 Training Details

All experiments are conducted on a cluster of 4 nodes, each equipped with 8 H200 GPUs.

For most runs we use a per-GPU micro-batch size of 64 with 2 gradient accumulation steps, resulting in an effective global batch size of 4096 sequences, each containing 2048 tokens. For RQ3 experiments with a sequence length of 1024, we double the micro-batch size to 128 in order to keep the number of training steps consistent across runs. Conversely, for sequence lengths of 4096 and 8192, we halve and quarter the micro-batch size, respectively, to maintain the same training step count. For our 8B model runs we use a micro batch size of 4 with 32 gradient accumulation steps to attain the same effective global batch size of 4096 sequences.

Training is implemented using the GPT-NeoX framework (Andonian et al., 2023), selected for its strong support for partial RoPE and our existing familiarity with its ecosystem. Optimization is performed using AdamW (Loshchilov and Hutter, 2019) with an initial learning rate of 4×10^{-4} (unless otherwise specified), a 5% warmup phase, and cosine learning rate decay to 10% of the original learning rate (Loshchilov and Hutter, 2017), without any dropout. All models are trained in mixed precision (Micikevicius et al., 2018) following common conventions, where forward and backward computations are performed in BF16 while gradient accumulation and inter-GPU reductions

are carried out in FP32. The training setup employs pure data parallelism across all GPUs, with no additional forms of model or pipeline parallelism. For 8B models, we follow Wei et al. (2026) by adding extra layers, resulting in a total of 36 layers compared to the 32 layers in Llama-3.1-8B. We also employ BF16 gradient accumulation with FP32 reductions to reduce memory usage.

We apply QK-Norm only over the hidden dimension of each attention head, rather than across both the attention heads and the head dimension. The latter is the default behavior in GPT-NeoX, which differs from the original QK-Norm formulation (Henry et al., 2020). The original method also works with models that employ grouped query attention (GQA), such as the Llama-3.2-1B-based sequential models used in our experiments.

For the 1B-parameter range, runs with parallel attention completed in 12–14 hours, while runs with sequential attention took 24–27 hours, with the longer sequence-length runs corresponding to the higher end of this range. Each 8B run took approximately 120 hours.

C.2 Evaluation Details

The benchmarks that we borrow from the Pythia Suite include ARC (Clark et al., 2018), LogiQA (Liu et al., 2020), LAMBADA (Radford et al., 2019; Paperno et al., 2016), PIQA (Bisk et al., 2020), SciQ (Johannes Welbl, 2017), WinoGrande (Sakaguchi et al., 2019), and WSC (Levesque et al., 2012).

For our MCQ evaluations, we primarily use byte-length normalized accuracy. However, for certain benchmarks, specifically Winogrande, PubMedQA, and WSC, LMEvalHarness only reports unnormalized accuracy, so we adopt that measure to remain consistent with prior work. Additionally, for LAMBADA, we consider perplexity alongside accuracy.

D Computation of RoPE Cache Size

Rotary Positional Embeddings (RoPE) enhance transformer models by encoding positional information without learnable parameters. Instead of storing explicit embedding vectors for each position, RoPE applies a rotational transformation to query and key vectors on-the-fly. To optimize this process, the sine and cosine values required for these rotations are pre-computed and stored in a cache upon model initialization.

The VRAM required to store the RoPE cache is a direct function of three key parameters: the maximum supported sequence length, the dimensionality of each attention head, and the numerical precision used for storage. The total size can be calculated using the following formula:

$$S_{cache} = L_{max} \times D_{head} \times P_{bytes} \qquad (1)$$

Where:

- S_{cache} is the total cache size in bytes.
- L_{max} represents the maximum sequence length (or context window) the model is configured to handle.
- D_{head} is the dimensionality of the vector for a single attention head.
- P_{bytes} represents the number of bytes needed to store a single numerical value, determined by the chosen precision (e.g., 4 bytes for 32-bit floating point, FP32). We use FP32 in our calculations, as prior work has shown that lower precisions often lead to instability and divergence during long-context training, issues that are mitigated when using FP32 (Wang et al., 2024).

This linear relationship demonstrates that the cache size scales directly with both the maximum sequence length and the head dimension, a critical consideration when designing models for long-context applications.

For Fig. 1, we use an attention head dimension of 256, which corresponds to the head dimensionality of Pythia-1B. The memory requirements would increase further for larger head dimensions (in practice most current models seem to stick to 128). Our calculations also exclude any effects of memory fragmentation and consider only the raw storage needed for the RoPE sine/cosine cache.

E References for Partial RoPE Usage

Table 2 lists various models that utilize partial RoPE, along with links referencing this configuration in their model settings.

F Additional Results

Figures 5 showcase the training loss trajectories of sequential and parallel attention architectures with varying partial RoPE configurations.

Figures 6a & 6b showcase the persistence of loss spikes across seeds and learning rates.

Model	URL
EleutherAI/gpt-j-6b	www.hf.co/EleutherAI/gpt-j-6b/blob/main/config.json#L21
EleutherAI/gpt-neox-20b	www.hf.co/EleutherAI/gpt-neox-20b/blob/main/config.json#L19
EleutherAI/pythia-12b	www.hf.co/EleutherAI/pythia-12b/blob/main/config.json#L17
microsoft/phi-2	www.hf.co/microsoft/phi-2/blob/main/config.json#L20
nvidia/Nemotron-4-340B-Base	www.hf.co/nvidia/Nemotron-4-340B-Base/blob/main/model_config.yaml#L34
Qwen/Qwen3-Next-80B-A3B-Instruct	www.hf.co/Qwen/Qwen3-Next-80B-A3B-Instruct/blob/main/config.json#L31

Table 2: Examples of widely used open-weight/open-source models employing **Partial RoPE**, with references indicating this choice—parameter names vary across models due to differences in training frameworks. Note: For Pythia, we show one family member, though all members use the same 25% partial RoPE.

Figure 7a demonstrates that QK-Norm effectively eliminates the loss spike for the parallel attention architecture with NoPE.

Figure 7b demonstrates that QK-Norm effectively eliminates the loss spike, even for the sequential attention run with a sequence length of 8192.

Table 3 showcases the MCQ evaluation results for all models on all benchmarks.

Table 4 showcases the perplexity evaluation results for all models on LAMBADA.

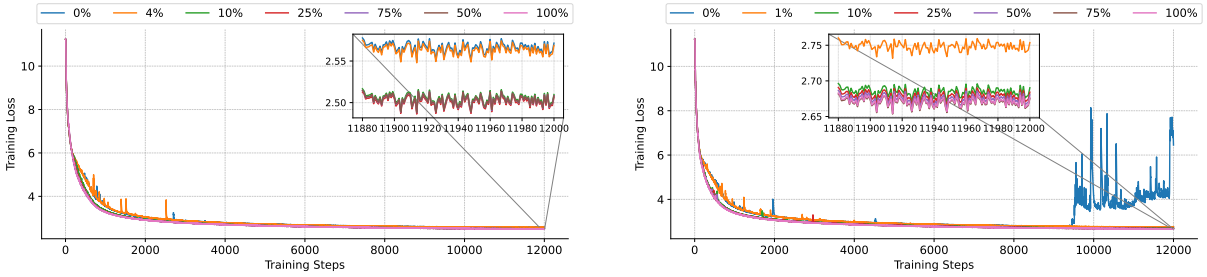
G LLM Usage

We use LLMs for help with grammatical corrections/writing of the paper and also as coding assistants. In both cases we independently verify the outputs.

H Artifact Release/Usage

We will publicly release all our artifacts (models, optimizer states, logged checkpoints, config files, etc.) upon acceptance.

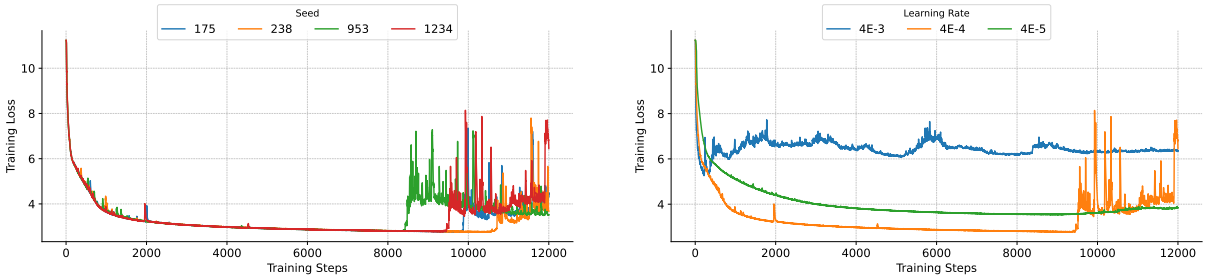
Our usage of artifacts such as the pretraining dataset and frameworks is consistent with their intended usage.



(a) Sequential attention model.

(b) Parallel attention model.

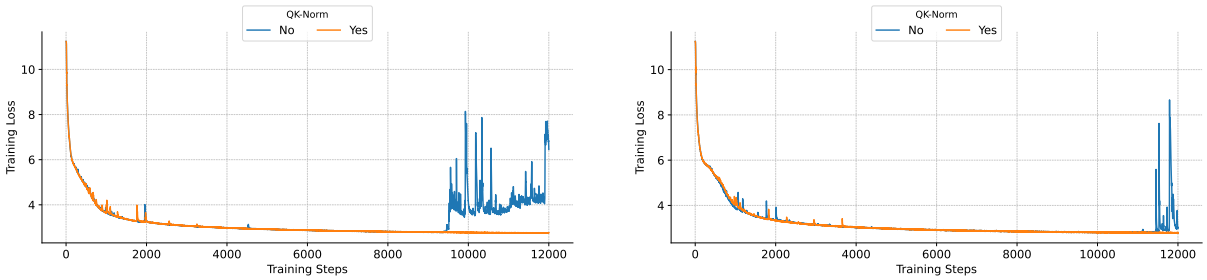
Figure 5: Training loss trajectories on the FineWeb-Edu dataset comparing sequential and parallel attention architectures under varying Partial RoPE configurations.



(a) Different random seeds.

(b) Different learning rates.

Figure 6: Training loss trajectories for parallel attention models trained on FineWeb-Edu under different experimental settings. The left panel varies random seeds, while the right panel varies learning rates.



(a) Pythia 1B (FineWeb-Edu): training loss trajectories with and without QK-Norm.

(b) LLaMA 1B (FineWeb, seq len 8192): QK-Norm stabilizes training and removes loss spike seen without normalization.

Figure 7: Comparison of training loss trajectories with and without QK-Norm across different 1B parameter models and training setups.

Table 3: **MCQ Benchmark Results.** We run evaluations through EleutherAI's Language Model Evaluation Harness (Gao et al., 2023) under Zero-Shot setting.

Config	ARC Challenge	ARC Easy	LogiQA	LAMBADA (OpenAI)	LAMBADA (Standard)	PIQA	SciQ	PubMedQA	Wino -Grande	WSC
Llama-3.2-1B x FW-Edu x Seq. Len. 2048										
0% RoPE (NoPE)	0.31 (0.01)	0.57 (0.01)	0.28 (0.02)	0.36 (0.01)	0.27 (0.01)	0.70 (0.01)	0.75 (0.01)	0.56 (0.02)	0.51 (0.01)	0.55 (0.05)
4% RoPE	0.31 (0.01)	0.58 (0.01)	0.28 (0.02)	0.36 (0.01)	0.27 (0.01)	0.69 (0.01)	0.75 (0.01)	0.51 (0.02)	0.52 (0.01)	0.38 (0.05)
10% RoPE	0.32 (0.01)	0.60 (0.01)	0.28 (0.02)	0.38 (0.01)	0.28 (0.01)	0.71 (0.01)	0.75 (0.01)	0.57 (0.02)	0.54 (0.01)	0.44 (0.05)
25% RoPE	0.31 (0.01)	0.58 (0.01)	0.26 (0.02)	0.38 (0.01)	0.29 (0.01)	0.70 (0.01)	0.75 (0.01)	0.52 (0.02)	0.53 (0.01)	0.54 (0.05)
50% RoPE	0.32 (0.01)	0.59 (0.01)	0.26 (0.02)	0.37 (0.01)	0.29 (0.01)	0.71 (0.01)	0.76 (0.01)	0.55 (0.02)	0.53 (0.01)	0.63 (0.05)
75% RoPE	0.33 (0.01)	0.61 (0.01)	0.27 (0.02)	0.39 (0.01)	0.27 (0.01)	0.71 (0.01)	0.75 (0.01)	0.48 (0.02)	0.52 (0.01)	0.62 (0.05)
100% RoPE	0.33 (0.01)	0.59 (0.01)	0.27 (0.02)	0.39 (0.01)	0.30 (0.01)	0.71 (0.01)	0.73 (0.01)	0.50 (0.02)	0.52 (0.01)	0.37 (0.05)
Llama-3.2-1B x FW x Seq. Len. 1024										
0% RoPE (NoPE)	0.27 (0.01)	0.48 (0.01)	0.27 (0.02)	0.46 (0.01)	0.35 (0.01)	0.72 (0.01)	0.71 (0.01)	0.51 (0.02)	0.51 (0.01)	0.54 (0.05)
4% RoPE	0.27 (0.01)	0.50 (0.01)	0.28 (0.02)	0.46 (0.01)	0.33 (0.01)	0.71 (0.01)	0.70 (0.01)	0.50 (0.02)	0.53 (0.01)	0.48 (0.05)
10% RoPE	0.26 (0.01)	0.50 (0.01)	0.27 (0.02)	0.49 (0.01)	0.38 (0.01)	0.72 (0.01)	0.71 (0.01)	0.55 (0.02)	0.52 (0.01)	0.41 (0.05)
25% RoPE	0.26 (0.01)	0.49 (0.01)	0.28 (0.02)	0.48 (0.01)	0.38 (0.01)	0.72 (0.01)	0.72 (0.01)	0.55 (0.02)	0.55 (0.01)	0.37 (0.05)
50% RoPE	0.25 (0.01)	0.49 (0.01)	0.27 (0.02)	0.48 (0.01)	0.38 (0.01)	0.72 (0.01)	0.71 (0.01)	0.55 (0.02)	0.52 (0.01)	0.49 (0.05)
75% RoPE	0.26 (0.01)	0.49 (0.01)	0.27 (0.02)	0.47 (0.01)	0.38 (0.01)	0.73 (0.01)	0.71 (0.01)	0.56 (0.02)	0.53 (0.01)	0.62 (0.05)
100% RoPE	0.27 (0.01)	0.50 (0.01)	0.28 (0.02)	0.48 (0.01)	0.37 (0.01)	0.73 (0.01)	0.68 (0.01)	0.58 (0.02)	0.53 (0.01)	0.37 (0.05)
Llama-3.2-1B x FW x Seq. Len. 2048										
0% RoPE (NoPE)	0.27 (0.01)	0.47 (0.01)	0.28 (0.02)	0.44 (0.01)	0.33 (0.01)	0.71 (0.01)	0.71 (0.01)	0.55 (0.02)	0.52 (0.01)	0.38 (0.05)
4% RoPE	0.28 (0.01)	0.46 (0.01)	0.26 (0.02)	0.45 (0.01)	0.33 (0.01)	0.71 (0.01)	0.69 (0.01)	0.55 (0.02)	0.51 (0.01)	0.37 (0.05)
10% RoPE	0.26 (0.01)	0.48 (0.01)	0.28 (0.02)	0.47 (0.01)	0.36 (0.01)	0.73 (0.01)	0.68 (0.01)	0.55 (0.02)	0.52 (0.01)	0.43 (0.05)
25% RoPE	0.27 (0.01)	0.47 (0.01)	0.25 (0.02)	0.46 (0.01)	0.37 (0.01)	0.72 (0.01)	0.68 (0.01)	0.51 (0.02)	0.54 (0.01)	0.56 (0.05)
50% RoPE	0.27 (0.01)	0.48 (0.01)	0.27 (0.02)	0.47 (0.01)	0.37 (0.01)	0.72 (0.01)	0.71 (0.01)	0.56 (0.02)	0.53 (0.01)	0.49 (0.05)
75% RoPE	0.26 (0.01)	0.49 (0.01)	0.27 (0.02)	0.48 (0.01)	0.36 (0.01)	0.72 (0.01)	0.70 (0.01)	0.49 (0.02)	0.51 (0.01)	0.38 (0.05)
100% RoPE	0.26 (0.01)	0.49 (0.01)	0.28 (0.02)	0.47 (0.01)	0.37 (0.01)	0.73 (0.01)	0.69 (0.01)	0.53 (0.02)	0.53 (0.01)	0.37 (0.05)
Llama-3.2-1B x FW x Seq. Len. 4096										
0% RoPE (NoPE)	0.26 (0.01)	0.45 (0.01)	0.27 (0.02)	0.43 (0.01)	0.30 (0.01)	0.70 (0.01)	0.68 (0.01)	0.56 (0.02)	0.52 (0.01)	0.60 (0.05)
4% RoPE	0.25 (0.01)	0.46 (0.01)	0.26 (0.02)	0.42 (0.01)	0.30 (0.01)	0.70 (0.01)	0.69 (0.01)	0.56 (0.02)	0.52 (0.01)	0.37 (0.05)
10% RoPE	0.27 (0.01)	0.47 (0.01)	0.29 (0.02)	0.46 (0.01)	0.35 (0.01)	0.72 (0.01)	0.69 (0.01)	0.56 (0.02)	0.52 (0.01)	0.47 (0.05)
25% RoPE	0.26 (0.01)	0.47 (0.01)	0.27 (0.02)	0.47 (0.01)	0.34 (0.01)	0.72 (0.01)	0.69 (0.01)	0.46 (0.02)	0.53 (0.01)	0.42 (0.05)
50% RoPE	0.26 (0.01)	0.47 (0.01)	0.29 (0.02)	0.46 (0.01)	0.36 (0.01)	0.72 (0.01)	0.68 (0.01)	0.54 (0.02)	0.52 (0.01)	0.63 (0.05)
75% RoPE	0.27 (0.01)	0.48 (0.01)	0.27 (0.02)	0.46 (0.01)	0.36 (0.01)	0.71 (0.01)	0.73 (0.01)	0.56 (0.02)	0.53 (0.01)	0.64 (0.05)
100% RoPE	0.27 (0.01)	0.48 (0.01)	0.28 (0.02)	0.48 (0.01)	0.37 (0.01)	0.72 (0.01)	0.68 (0.01)	0.58 (0.02)	0.52 (0.01)	0.40 (0.05)
Llama-3.2-1B x FW x Seq. Len. 8192										
0% RoPE (NoPE)	0.23 (0.01)	0.39 (0.01)	0.26 (0.02)	0.23 (0.01)	0.16 (0.01)	0.64 (0.01)	0.61 (0.02)	0.52 (0.02)	0.52 (0.01)	0.37 (0.05)
0% RoPE (NoPE) + QK-Norm	0.26 (0.01)	0.45 (0.01)	0.27 (0.02)	0.40 (0.01)	0.30 (0.01)	0.70 (0.01)	0.68 (0.01)	0.55 (0.02)	0.50 (0.01)	0.38 (0.05)
4% RoPE	0.25 (0.01)	0.45 (0.01)	0.26 (0.02)	0.41 (0.01)	0.30 (0.01)	0.71 (0.01)	0.68 (0.01)	0.55 (0.02)	0.53 (0.01)	0.66 (0.05)
10% RoPE	0.25 (0.01)	0.46 (0.01)	0.27 (0.02)	0.45 (0.01)	0.35 (0.01)	0.71 (0.01)	0.69 (0.01)	0.50 (0.02)	0.51 (0.01)	0.37 (0.05)
25% RoPE	0.26 (0.01)	0.47 (0.01)	0.26 (0.02)	0.45 (0.01)	0.34 (0.01)	0.71 (0.01)	0.71 (0.01)	0.55 (0.02)	0.52 (0.01)	0.37 (0.05)
50% RoPE	0.27 (0.01)	0.47 (0.01)	0.28 (0.02)	0.47 (0.01)	0.33 (0.01)	0.71 (0.01)	0.69 (0.01)	0.46 (0.02)	0.52 (0.01)	0.47 (0.05)
75% RoPE	0.25 (0.01)	0.46 (0.01)	0.29 (0.02)	0.47 (0.01)	0.37 (0.01)	0.71 (0.01)	0.69 (0.01)	0.55 (0.02)	0.53 (0.01)	0.38 (0.05)
100% RoPE	0.25 (0.01)	0.47 (0.01)	0.28 (0.02)	0.47 (0.01)	0.35 (0.01)	0.72 (0.01)	0.69 (0.01)	0.53 (0.02)	0.53 (0.01)	0.63 (0.05)
Pythia-1B x FW-Edu x Seq. Len. 2048										
0% RoPE (NoPE)	0.24 (0.01)	0.30 (0.01)	0.24 (0.02)	0.00 (0.00)	0.00 (0.00)	0.53 (0.01)	0.28 (0.01)	0.34 (0.02)	0.50 (0.01)	0.63 (0.05)
0% RoPE (NoPE) + QK-Norm	0.28 (0.01)	0.52 (0.01)	0.29 (0.02)	0.28 (0.01)	0.22 (0.01)	0.67 (0.01)	0.71 (0.01)	0.53 (0.02)	0.52 (0.01)	0.37 (0.05)
1% RoPE	0.28 (0.01)	0.50 (0.01)	0.25 (0.02)	0.30 (0.01)	0.23 (0.01)	0.66 (0.01)	0.72 (0.01)	0.54 (0.02)	0.51 (0.01)	0.37 (0.05)
10% RoPE	0.29 (0.01)	0.53 (0.01)	0.27 (0.02)	0.34 (0.01)	0.24 (0.01)	0.68 (0.01)	0.73 (0.01)	0.48 (0.02)	0.51 (0.01)	0.37 (0.05)
25% RoPE	0.30 (0.01)	0.55 (0.01)	0.27 (0.02)	0.34 (0.01)	0.23 (0.01)	0.68 (0.01)	0.72 (0.01)	0.45 (0.02)	0.51 (0.01)	0.55 (0.05)
50% RoPE	0.29 (0.01)	0.54 (0.01)	0.27 (0.02)	0.33 (0.01)	0.23 (0.01)	0.67 (0.01)	0.73 (0.01)	0.48 (0.02)	0.51 (0.01)	0.41 (0.05)
75% RoPE	0.30 (0.01)	0.55 (0.01)	0.27 (0.02)	0.35 (0.01)	0.24 (0.01)	0.67 (0.01)	0.72 (0.01)	0.52 (0.02)	0.51 (0.01)	0.37 (0.05)
100% RoPE	0.28 (0.01)	0.54 (0.01)	0.27 (0.02)	0.34 (0.01)	0.24 (0.01)	0.69 (0.01)	0.72 (0.01)	0.55 (0.02)	0.51 (0.01)	0.37 (0.05)
Llama-3.1-8B x FW x Seq. Len. 2048										
0% RoPE (NoPE)	0.30 (0.01)	0.57 (0.01)	0.28 (0.02)	0.58 (0.01)	0.49 (0.01)	0.77 (0.01)	0.80 (0.01)	0.60 (0.02)	0.57 (0.01)	0.42 (0.05)
10% RoPE	0.32 (0.01)	0.60 (0.01)	0.27 (0.02)	0.60 (0.01)	0.50 (0.01)	0.77 (0.01)	0.81 (0.01)	0.64 (0.02)	0.59 (0.01)	0.55 (0.05)
25% RoPE	0.33 (0.01)	0.58 (0.01)	0.27 (0.02)	0.59 (0.01)	0.52 (0.01)	0.77 (0.01)	0.82 (0.01)	0.60 (0.02)	0.59 (0.01)	0.59 (0.05)
50% RoPE	0.31 (0.01)	0.58 (0.01)	0.29 (0.02)	0.60 (0.01)	0.51 (0.01)	0.77 (0.01)	0.81 (0.01)	0.57 (0.02)	0.59 (0.01)	0.41 (0.05)
75% RoPE	0.33 (0.01)	0.58 (0.01)	0.29 (0.02)	0.60 (0.01)	0.52 (0.01)	0.77 (0.01)	0.79 (0.01)	0.59 (0.02)	0.58 (0.01)	0.44 (0.05)
100% RoPE	0.33 (0.01)	0.59 (0.01)	0.28 (0.02)	0.60 (0.01)	0.52 (0.01)	0.77 (0.01)	0.81 (0.01)	0.62 (0.02)	0.59 (0.01)	0.49 (0.05)

Table 4: **Perplexity results for LAMBADA.** We run evaluations through EleutherAI's Language Model Evaluation Harness (Gao et al., 2023) under Zero-Shot setting.

Config	LAMBADA (OpenAI)	LAMBADA (Standard)	LAMBADA
Llama-3.2-1B x FW-Edu x 2048			
0% RoPE (NoPE)	26.72 (0.96)	69.87 (2.90)	48.30 (11.00)
4% RoPE	27.71 (1.00)	68.78 (2.78)	48.25 (10.48)
10% RoPE	23.25 (0.82)	64.50 (2.63)	43.87 (10.50)
25% RoPE	23.51 (0.83)	64.78 (2.75)	44.14 (10.52)
50% RoPE	23.89 (0.85)	64.87 (2.75)	44.38 (10.45)
75% RoPE	22.97 (0.81)	76.38 (3.26)	49.67 (13.56)
100% RoPE	22.99 (0.82)	57.82 (2.35)	40.41 (8.88)
Llama-3.2-1B x FW x Seq. Len. 1024			
0% RoPE (NoPE)	13.49 (0.41)	31.95 (1.15)	22.72 (4.70)
4% RoPE	13.41 (0.41)	32.38 (1.17)	22.90 (4.82)
10% RoPE	11.80 (0.36)	25.88 (0.92)	18.84 (3.59)
25% RoPE	12.35 (0.37)	27.11 (0.94)	19.73 (3.76)
50% RoPE	12.10 (0.37)	25.64 (0.90)	18.87 (3.45)
75% RoPE	12.35 (0.37)	24.07 (0.83)	18.21 (3.00)
100% RoPE	12.33 (0.37)	26.86 (0.95)	19.59 (3.70)
Llama-3.2-1B x FW x Seq. Len. 2048			
0% RoPE (NoPE)	14.90 (0.46)	37.15 (1.38)	26.03 (5.66)
4% RoPE	14.55 (0.45)	36.95 (1.36)	25.75 (5.69)
10% RoPE	12.86 (0.39)	26.91 (0.94)	19.89 (3.59)
25% RoPE	13.07 (0.40)	29.79 (1.09)	21.43 (4.26)
50% RoPE	12.96 (0.40)	27.11 (0.97)	20.03 (3.62)
75% RoPE	13.09 (0.40)	30.93 (1.14)	22.01 (4.54)
100% RoPE	12.78 (0.39)	27.73 (0.98)	20.26 (3.81)
Llama-3.2-1B x FW x Seq. Len. 4096			
0% RoPE (NoPE)	16.03 (0.51)	43.96 (1.64)	30.00 (7.09)
4% RoPE	16.61 (0.53)	43.40 (1.63)	30.01 (6.80)
10% RoPE	13.39 (0.41)	30.37 (1.09)	21.88 (4.33)
25% RoPE	13.48 (0.42)	33.71 (1.24)	23.60 (5.14)
50% RoPE	13.43 (0.41)	30.68 (1.12)	22.05 (4.40)
75% RoPE	13.22 (0.40)	31.09 (1.14)	22.16 (4.55)
100% RoPE	12.93 (0.40)	27.90 (1.01)	20.41 (3.82)
Llama-3.2-1B x FW x Seq. Len. 8192			
0% RoPE (NoPE)	79.69 (3.37)	396.00 (20.03)	237.85 (80.38)
0% RoPE (NoPE) + QK-Norm	19.09 (0.62)	47.90 (1.85)	33.49 (7.33)
4% RoPE	18.83 (0.61)	49.06 (1.91)	33.95 (7.69)
10% RoPE	14.39 (0.45)	32.69 (1.20)	23.54 (4.66)
25% RoPE	14.64 (0.46)	38.92 (1.53)	26.78 (6.18)
50% RoPE	14.15 (0.45)	40.83 (1.57)	27.49 (6.77)
75% RoPE	13.76 (0.43)	32.29 (1.21)	23.02 (4.72)
100% RoPE	13.55 (0.42)	34.64 (1.31)	24.10 (5.36)
Pythia-1B x FW-Edu x Seq. Len. 2048			
0% RoPE (NoPE)	340933.20 (25150.68)	3717492.05 (300190.33)	2029212.62 (870637.70)
0% RoPE (NoPE) + QK-Norm	51.74 (2.04)	136.88 (5.85)	94.31 (21.73)
1% RoPE	43.13 (1.69)	144.46 (6.33)	93.79 (25.75)
10% RoPE	33.49 (1.27)	115.78 (5.06)	74.64 (20.90)
25% RoPE	35.18 (1.34)	140.81 (6.30)	88.00 (26.80)
50% RoPE	33.35 (1.26)	131.37 (5.78)	82.36 (24.86)
75% RoPE	32.52 (1.23)	114.03 (4.93)	73.27 (20.69)
100% RoPE	32.88 (1.24)	122.90 (5.35)	77.89 (22.84)
Llama-3.1-8B x FW x Seq. Len. 2048			
0% RoPE (NoPE)	6.80 (0.18)	11.01 (0.32)	8.91 (1.09)
10% RoPE	6.42 (0.17)	10.08 (0.29)	8.25 (0.94)
25% RoPE	6.37 (0.16)	9.66 (0.27)	8.01 (0.85)
50% RoPE	6.18 (0.16)	10.05 (0.29)	8.12 (1.00)
75% RoPE	6.07 (0.15)	9.34 (0.26)	7.71 (0.85)
100% RoPE	6.11 (0.15)	9.03 (0.25)	7.57 (0.76)