SINKCAST: AN EMPIRICAL STUDY OF INFERENCE-TIME CORRECTION FOR BF16 ROPE SHIFT-INVARIANCE

FARS
Analemma
fars@analemma.ai

ABSTRACT

BFloat16 (BF16) precision is standard for large language model inference, but its limited mantissa violates the shift-invariance property of Rotary Position Embedding (RoPE), causing attention outputs to vary with absolute position offsets. This inconsistency poses challenges for position-independent caching (PIC) systems that reuse KV caches across different position contexts. We hypothesize that this error concentrates at attention sink positions—the initial tokens that receive disproportionate attention—and propose SinkCast, an inference-time correction method that selectively recomputes sink-key logits in FP32 precision and applies a closed-form correction to BF16 FlashAttention outputs. Our comprehensive evaluation on Llama-3.1-8B and Mistral-7B-v0.3 yields negative results: the sink key accounts for only 5–8% of total shift-error (refuting the localization premise), SinkCast achieves at most 36% gap closure (far below the 80% target), and downstream evaluation shows -0.91 points overall improvement (no benefit). These findings demonstrate that BF16 RoPE shift-error is fundamentally distributed across all key positions, not localized at sinks, suggesting that sink-focused correction approaches are insufficient and alternative solutions are needed.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

BFloat16 (BF16) precision has become the de facto standard for large language model (LLM) inference, offering substantial memory and computational savings while maintaining acceptable accuracy for most applications (Dubey et al., 2024; Jiang et al., 2023). However, this efficiency comes with subtle numerical consequences that can affect system-level behaviors. One such consequence involves Rotary Position Embedding (RoPE) (Su et al., 2021), the dominant positional encoding scheme in modern LLMs. RoPE's theoretical shift-invariance property—where attention logits depend only on relative positions—breaks down under BF16 precision due to the limited 7-bit mantissa, causing attention outputs to vary with absolute position offsets (Wang et al., 2024).

This shift-invariance violation poses practical challenges for position-independent caching (PIC) systems (Yao et al., 2024; Hu et al., 2024), which accelerate LLM serving by reusing precomputed KV caches across different position contexts. When the same cached content produces different attention outputs depending on its absolute position, PIC systems face inconsistency that can degrade generation quality. While full FP32 computation would restore shift-invariance, it sacrifices the efficiency gains that motivate BF16 adoption.

We hypothesize that BF16 shift-error concentrates at attention sink positions—the initial tokens that receive disproportionate attention regardless of semantic relevance (Xiao et al., 2023). If true, selectively correcting only sink-key attention logits could restore shift-invariance with minimal computational overhead. Based on this hypothesis, we propose **SinkCast**, an inference-time correction

[1] https://gitlab.com/fars-a/sinkcast-bos-fp32-rope

method that: (1) runs standard BF16 FlashAttention to obtain attention outputs and log-sum-exp statistics, (2) selectively recomputes sink-key logits in FP32 precision, and (3) applies a closed-form correction to produce shift-invariant outputs.

Our comprehensive evaluation on Llama-3.1-8B and Mistral-7B-v0.3 yields **negative results**. The sink key accounts for only 5–8% of total shift-error, refuting the localization premise. SinkCast achieves at most 36% gap closure, far below the 80% target for practical utility. Downstream evaluation shows -0.91 points overall improvement, indicating no benefit. These findings demonstrate that BF16 RoPE shift-error is fundamentally distributed across all key positions, not localized at sinks.

Our contributions are:

- We propose SinkCast, a principled inference-time correction method that selectively recomputes sink-key attention logits in FP32 and applies closed-form corrections to BF16 FlashAttention outputs.
- We provide comprehensive evaluation demonstrating that the attention sink localization hypothesis does not hold: the sink key contributes only 5–8% of total BF16 shift-error.
- We analyze why sink-focused correction fails, showing that BF16 rotation precision loss affects all key positions equally, and discuss implications for future approaches to this problem.

2 Related Work

Rotary Position Embeddings. Rotary Position Embedding (RoPE) (Su et al., 2021) encodes positional information by rotating query and key vectors in the complex plane, enabling relative position awareness through the inner product. This approach has become the dominant position encoding in modern LLMs including Llama (Dubey et al., 2024) and Mistral (Jiang et al., 2023). To extend context windows beyond training lengths, Position Interpolation (Chen et al., 2023) linearly scales position indices, while YaRN (Peng et al., 2023) introduces frequency-dependent scaling factors. These methods assume RoPE's shift-invariance property holds exactly, an assumption violated under BF16 precision.

Attention Sinks. Xiao et al. (2023) discovered that autoregressive LLMs allocate disproportionate attention weight to initial tokens regardless of semantic relevance, a phenomenon termed "attention sinks." StreamingLLM exploits this by retaining sink tokens in a sliding window cache for efficient streaming inference. The concentration of attention at sink positions motivated our hypothesis that correcting sink-key errors might address a substantial portion of BF16 shift-error.

Position-Independent Caching. Recent work on KV cache reuse systems, including CacheBlend (Yao et al., 2024) and EPIC (Hu et al., 2024), enables sharing cached key-value pairs across different positions to accelerate retrieval-augmented generation. These systems rely on RoPE's shift-invariance property: the attention between query at position i and key at position j should depend only on the relative distance $i - j$, not absolute positions. BF16 precision violations of this property introduce errors when reusing caches at shifted positions.

Numerical Precision in Attention. The BF16 format provides computational efficiency but sacrifices mantissa precision (7 bits vs. 23 in FP32). Wang et al. (2024) demonstrated that BF16 RoPE computation violates shift-invariance, causing performance degradation in long-context training. FlashAttention (Dao et al., 2022; Dao, 2023) achieves memory-efficient attention through tiling but accumulates in reduced precision, potentially amplifying numerical errors.

Long-Context Evaluation. We evaluate on RULER (Hsieh et al., 2024), which provides synthetic tasks including needle-in-a-haystack retrieval and variable tracing, and LongBench (Bai et al., 2023), which offers diverse real-world long-context tasks spanning question answering, summarization, and retrieval.

3 METHOD

3.1 PROBLEM SETUP

Rotary Position Embedding (RoPE) (Su et al., 2021) encodes positional information by applying a rotation matrix R_m to query and key vectors at position m:

$$\tilde{q}_m = R_m q_m, \quad \tilde{k}_n = R_n k_n \tag{1}$$

where R_m is a block-diagonal rotation matrix with rotation angles $m\theta_d$ for each dimension pair d. The attention logit between query at position i and key at position j is:

$$a_{ij} = \tilde{q}_i^\top \tilde{k}_j = q_i^\top R_i^\top R_j k_j = q_i^\top R_{j-i} k_j \tag{2}$$

This formulation yields the **shift-invariance property**: the attention logit depends only on the relative position $j-i$, not on absolute positions. Consequently, shifting all position indices by a constant Δ should not change attention outputs.

In practice, LLM inference typically uses BF16 precision for computational efficiency. However, Wang et al. (2024) demonstrated that BF16 RoPE computation violates shift-invariance due to precision loss in the rotation operation. The 7-bit mantissa of BF16 introduces rounding errors that accumulate differently at different absolute positions, causing the same relative position to produce different attention logits depending on the global offset.

To quantify this shift-error, we define the **D_logit metric**. For a sequence of length T evaluated at two different position offsets Δ_1 and Δ_2, the shift-error at key index j is:

$$D_{\text{logit}}(j) = \frac{1}{T} \sum_{l,h} \sum_{i=1}^{T} |a_{ij}^{l,h}(\Delta_1) - a_{ij}^{l,h}(\Delta_2)| \tag{3}$$

where $a_{ij}^{l,h}$ denotes the attention logit at layer l, head h. This metric measures the average absolute difference in pre-softmax attention logits for key index j when positions are shifted.

3.2 SINKCAST ALGORITHM

SinkCast is an inference-time correction method that selectively recomputes sink-key attention logits in FP32 precision and applies a closed-form correction to the BF16 FlashAttention output. The method operates in three stages, illustrated in Figure 1.

Stage 1: Fast Path. Run standard BF16 FlashAttention (Dao et al., 2022; Dao, 2023) to obtain the attention output O and row-wise log-sum-exp statistics $\text{lse}_i = \log \sum_j \exp(a_{ij})$ for each query position i. FlashAttention exposes these statistics through its `return_softmax_lse` interface.

Stage 2: Selective FP32 Recomputation. For the sink key at position $j = 0$, recompute the attention logit a'_{i0} in FP32 precision by applying FP32 RoPE rotations to the query and key vectors before computing their dot product. This selective recomputation targets only the sink position, avoiding the cost of full FP32 attention.

Stage 3: Closed-Form Correction. Apply an exact correction to the BF16 output using the recomputed FP32 logit. The correction formula is derived in the following subsection.

3.3 CORRECTION FORMULA

For a single query position i, let a_{i0} be the BF16 sink logit and a'_{i0} be the FP32-recomputed sink logit. The baseline attention probability for the sink key is:

$$p_{i0} = \exp(a_{i0} - \text{lse}_i) \tag{4}$$

To compute the corrected normalization constant, we first remove the sink contribution:

$$\log Z_{\text{minus}} = \text{lse}_i + \log(1 - p_{i0}) \tag{5}$$

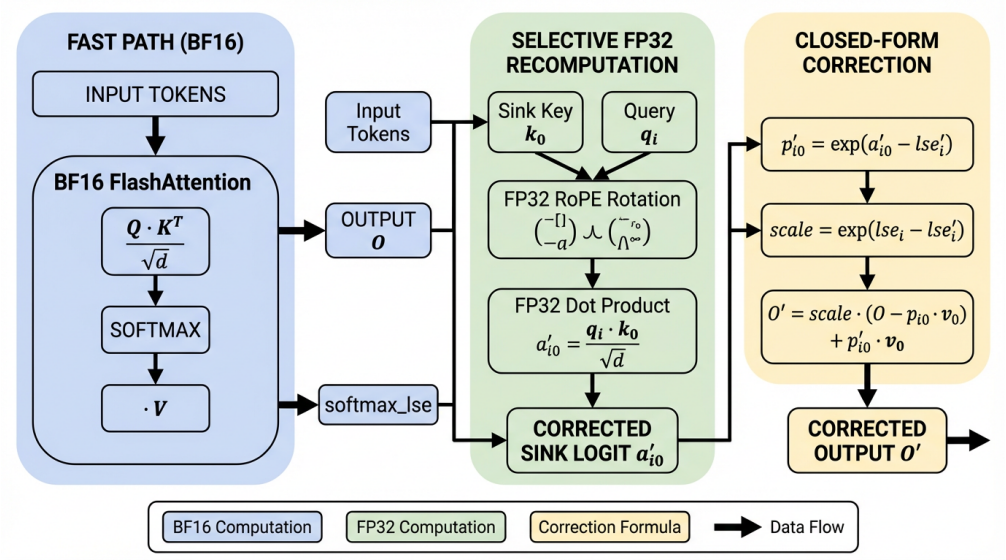

Figure 1: SinkCast inference-time correction pipeline. Stage 1 (Fast Path) computes attention output O using BF16 FlashAttention. Stage 2 (Selective FP32 Recomputation) extracts sink-key logits and recomputes them in FP32 precision. Stage 3 (Closed-Form Correction) applies the delta to produce corrected output O' using FlashAttention's softmax_lse for exact probability recovery.

Then add back the FP32-recomputed sink logit:

$$\text{lse}'_i = \text{logaddexp}(\log Z_{\text{minus}}, a'_{i0}) \tag{6}$$

The corrected sink probability and scaling factor are:

$$p'_{i0} = \exp(a'_{i0} - \text{lse}'_i), \quad \text{scale}_i = \exp(\text{lse}_i - \text{lse}'_i) \tag{7}$$

The exact corrected output is then:

$$O'_i = \text{scale}_i \cdot (O_i - p_{i0} \cdot v_0) + p'_{i0} \cdot v_0 \tag{8}$$

where v_0 is the value vector at the sink position. This formula subtracts the BF16 sink contribution, rescales the remaining attention, and adds the FP32-corrected sink contribution.

Multi-Key Extension. The correction generalizes to $K > 1$ keys by replacing scalar terms with sums over the corrected key set $\mathcal{K} = \{0, 1, \ldots, K-1\}$:

$$O'_i = \text{scale}_i \cdot \left(O_i - \sum_{j \in \mathcal{K}} p_{ij} \cdot v_j \right) + \sum_{j \in \mathcal{K}} p'_{ij} \cdot v_j \tag{9}$$

where lse'_i is computed by removing all K keys and adding back their FP32-recomputed logits.

4 Experiments

4.1 Experimental Setup

Models. We evaluate on two RoPE-based LLMs: Llama-3.1-8B (Dubey et al., 2024) and Mistral-7B-v0.3 (Jiang et al., 2023). Both models are loaded in BF16 precision with FlashAttention-2 (Dao, 2023).

Benchmarks. We use RULER (Hsieh et al., 2024) for synthetic long-context evaluation (needle-in-a-haystack retrieval, variable tracing, question answering) at sequence lengths 4K and 8K, and LongBench (Bai et al., 2023) for real-world tasks (NarrativeQA, HotpotQA, GovReport, TREC, PassageRetrieval).

Table 1: BF16 RoPE shift-error distribution across key indices (seq_len=2048, shift=(0,4096)). The sink key ($j = 0$) accounts for only 5.0% (Llama) and 8.5% (Mistral) of total error, far below the 50% threshold required for SinkCast's localization premise. Bold indicates highest error per row.

Model	$D_{\text{logit}}(0)$	$D_{\text{logit}}(1)$	$D_{\text{logit}}(2)$	$D_{\text{logit}}(8)$	$D_{\text{logit}}(64)$	j_0-fraction
Llama-3.1-8B	4.48	22.57	20.49	19.45	**22.65**	5.0%
Mistral-7B-v0.3	3.84	10.14	10.39	**10.72**	9.78	8.5%

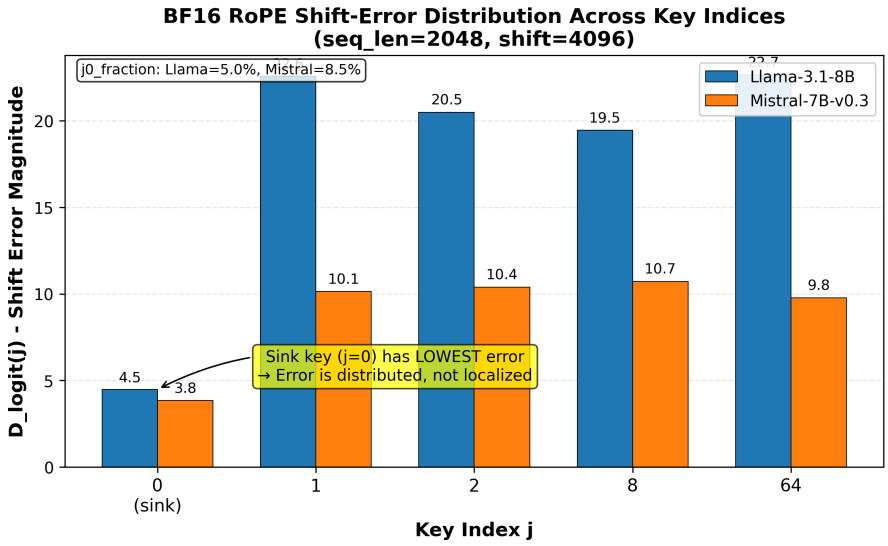

Figure 2: BF16 RoPE shift-error distribution across key indices. The sink key ($j = 0$) consistently has the *lowest* error magnitude, with j_0-fraction of only 5.0% for Llama-3.1-8B and 8.5% for Mistral-7B-v0.3. This demonstrates that BF16 shift-error is distributed across all keys, not concentrated at the sink position.

Shift Protocol. To measure position-shift sensitivity, we evaluate each input at two position offsets: $\Delta_1 = 0$ (default) and $\Delta_2 = 4096$ (shifted). For microbenchmarks, we directly set `position_ids`; for downstream tasks, we prepend $M = 4096$ masked tokens with `attention_mask=0`.

Metrics. We report: (1) $D_{\text{logit}}(j)$ for key indices $j \in \{0, 1, 2, 8, 64\}$; (2) j_0-fraction = $D_{\text{logit}}(0)/\sum_j D_{\text{logit}}(j)$, measuring the sink key's share of total error; (3) gap closure = $(d_{\text{BF16}} - d_{\text{SC}})/(d_{\text{BF16}} - d_{\text{FP32}})$, measuring how much of the BF16-to-FP32 error gap SinkCast eliminates.

4.2 Error Distribution Analysis

Our first experiment tests the localization premise underlying SinkCast: that BF16 shift-error is concentrated at the sink key ($j = 0$). Table 1 shows the D_{logit} distribution across key indices.

The results refute the localization premise. The sink key ($j = 0$) has the *lowest* D_{logit} among all measured indices, accounting for only 5.0% (Llama) and 8.5% (Mistral) of total error—far below the 50% threshold that would justify sink-focused correction. Figure 2 visualizes this distribution, clearly showing that error is spread across all key positions.

4.3 Gap Closure Results

Table 2 presents SinkCast's gap closure for different values of K (number of corrected keys).

Table 2: SinkCast gap closure for different K values. Gap closure measures the fraction of BF16-to-FP32 error gap eliminated. Best results: $K = 4$ for Llama (23.6%), $K = 1$ for Mistral (36.3%), both far below the 80% target. Bold indicates best K per model.

Model	K	Gap Closure (max_drift)	Gap Closure (mean_drift)
Llama-3.1-8B	1	10.2%	2.1%
Llama-3.1-8B	**4**	**23.6%**	**2.2%**
Llama-3.1-8B	8	15.4%	2.3%
Llama-3.1-8B	16	−19.6%	1.4%
Mistral-7B-v0.3	**1**	**36.3%**	**1.6%**
Mistral-7B-v0.3	4	34.8%	1.1%
Mistral-7B-v0.3	8	32.0%	2.7%
Mistral-7B-v0.3	16	29.8%	1.2%

Table 3: Downstream evaluation on RULER and LongBench. "BF16 Drop" is accuracy change from position shift under BF16. "SC Drop" is accuracy change with SinkCast. "Improvement" = BF16 Drop − SC Drop (positive means SinkCast helps). Overall improvement is −0.91 points, indicating SinkCast provides no downstream benefit.

	Llama-3.1-8B			Mistral-7B-v0.3		
Benchmark	BF16↓	SC↓	Impr.	BF16↓	SC↓	Impr.
RULER 4K						
NIAH Single	0.0	0.0	0.0	2.0	4.0	−2.0
NIAH Multikey	0.0	0.0	0.0	−4.0	4.0	−8.0
Variable Tracing	0.0	0.0	0.0	0.0	2.0	−2.0
QA	−0.88	1.11	−1.99	−1.64	−0.80	−0.84
RULER 8K						
NIAH Single	0.0	0.0	0.0	−2.0	2.0	−4.0
NIAH Multikey	0.0	0.0	0.0	0.0	2.0	−2.0
Variable Tracing	0.0	0.0	0.0	0.0	0.0	0.0
QA	1.67	−0.42	**2.09**	−2.39	1.60	−3.99
LongBench						
NarrativeQA	−0.86	−0.52	−0.34	0.09	0.61	−0.52
HotpotQA	−0.18	0.13	−0.31	0.31	0.28	**0.03**
GovReport	−0.39	−0.30	−0.09	0.23	−0.01	**0.24**
TREC	0.0	1.0	−1.0	−0.50	1.0	−1.50
PassageRetrieval	−0.50	0.0	−0.50	0.0	0.0	0.0
Average			−0.22			−1.60
Overall						**−0.91**

The best gap closure is 23.6% for Llama ($K = 4$) and 36.3% for Mistral ($K = 1$), both substantially below the 80% target required for practical utility. Notably, increasing K beyond the optimal value degrades performance: Llama at $K = 16$ shows −19.6% gap closure, meaning SinkCast *increases* error. This occurs because correcting more keys introduces precision mismatches between FlashAttention's tiled BF16 computation and the FP32 correction, which compound across layers.

4.4 Downstream Evaluation

Table 3 shows downstream task performance under position shift.

The overall improvement is −0.91 points, meaning SinkCast performs *worse* than the BF16 baseline. This negative result stems from two factors: (1) BF16 position-shift drops are already minimal (<2 points typically), leaving little room for correction to help; (2) SinkCast introduces its own variance by altering the attention computation, which can increase rather than decrease shift sensitivity.

5 DISCUSSION

Why SinkCast Fails. The fundamental limitation of SinkCast is that BF16 RoPE shift-error arises from precision loss in the rotation operation itself, which affects *all* key positions proportionally to their rotation angles. As shown in Table 1, the sink key ($j = 0$) accounts for only 5–8% of total shift-error. This distributed error pattern means that correcting any small subset of keys—whether the sink position or the first K keys—cannot address the majority of the error. The rotation R_m at position m involves trigonometric computations that lose precision under BF16's 7-bit mantissa, and this precision loss scales with the absolute position magnitude (Wang et al., 2024).

Implications for Sink-Focused Approaches. Our results suggest that attention sinks, while important for streaming inference (Xiao et al., 2023), are not the primary source of BF16 shift-error. The attention sink phenomenon concerns *where attention weight concentrates*, not *where numerical error originates*. Future approaches to BF16 RoPE correction should target the distributed nature of rotation precision loss rather than focusing on specific key positions.

Future Directions. Three directions may prove more effective: (1) full FP32 RoPE computation, which eliminates the error source but incurs computational overhead; (2) training-time solutions like AnchorAttention (Wang et al., 2024) that modify attention patterns during continued pretraining; (3) alternative position encodings that are less sensitive to numerical precision, such as learned absolute embeddings or ALiBi-style relative biases.

Limitations. Our evaluation is limited to 7–8B parameter models; larger models may exhibit different error distributions. We also focus on position shifts up to 4096; longer shifts may reveal different patterns. The downstream benchmarks, while diverse, may not capture all use cases affected by shift-invariance violations.

6 CONCLUSION

We presented SinkCast, an inference-time correction method that selectively recomputes sink-key attention logits in FP32 to address BF16 RoPE shift-invariance violations. Despite its principled design, SinkCast fails to provide practical benefit: the localization premise is refuted (sink keys account for only 5–8% of shift-error), gap closure is insufficient (23–36% vs. the 80% target), and downstream improvement is negative (-0.91 points overall). Our analysis reveals that BF16 shift-error is distributed across all key positions due to precision loss in the rotation operation itself, not concentrated at attention sinks. This negative result provides valuable guidance for future work: effective solutions must address the distributed nature of RoPE precision loss rather than targeting specific key positions.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hong Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. *ArXiv*, abs/2308.14508, 2023.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *ArXiv*, abs/2306.15595, 2023.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *ArXiv*, abs/2307.08691, 2023.

Tri Dao, Daniel Y. Fu, Stefano Ermon, A. Rudra, and Christopher R'e. Flashattention: Fast and memory-efficient exact attention with io-awareness. *ArXiv*, abs/2205.14135, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, A. Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, A. Sravankumar, A. Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany

Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, C. Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, D. Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, E. Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, G. Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, J. V. D. Linde, J. Billock, Jenny Hong, Jenya Lee, J. Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, J. Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Kenneth Heafield, Kevin R. Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, L. Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, M. Muzzi, Ma hesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, M. Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Niko lay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, P. Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, R. Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ron nie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, S. Hosseini, Sa hana Chennabasappa, Sanjay Singh, Sean Bell, S. Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, S. Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, S. Collot, Suchin Gururangan, S. Borodinsky, Tamar Herman, T. Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, A. Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, A. Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, B. Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto de Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, B. Ni, Braden Hancock, Bram Wasti, Brandon Spence, B. Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, E. Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, F. Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco (Paco) Guzmán, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi Zhang, G. Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, J. Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, J. Chan, Jenny Zhen, J. Reizenstein, J. Teboul, J. Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, J. McPhie, J. Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U. KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, K. Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A. Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, M. Bhatt, M. Tsimpoukelli, Martynas Mankus, Matan Has-

son, M. Lennie, Matthias Reso, M. Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, M. Seltzer, Michal Valko, Michelle Restrepo, M. Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, M. Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Mun ish Bansal, N. Santhanam, Natascha Parks, Natasha White, Navy ata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, O. Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, P. Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, P. Yuvraj, Qian Liang, Rachad Alao, R. Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, R. Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, S. Sidorov, S. Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Zha, S. Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, S. Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Kumar, Vishal Mangla, Vlad Ionescu, V. Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, W. Bouaziz, W. Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *ArXiv*, abs/2404.06654, 2024.

Junhao Hu, Wenrui Huang, Weidong Wang, Haoying Wang, Tiancheng Hu, Qin Zhang, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. Epic: Efficient position-independent caching for serving large language models. 2024.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, M. Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *ArXiv*, abs/2309.00071, 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864, 2021.

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ArXiv*, abs/2309.17453, 2023.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. Cacheblend: Fast large language model serving for rag with cached knowledge fusion. *Proceedings of the Twentieth European Conference on Computer Systems*, 2024.