International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Extending Context Windows in Large Language Models: A Survey of Techniques, Architectures, and Evaluation Methods

Mohammed Sule

Department of Computer Science, Hemchandracharya North Gujarat University
Email: *mohammed.sule922015[at]gmail.com*

Abstract: ***Since the inception of Transformers, models have dramatically evolved in their ability to model long and short-term memory. From the original Transformer with 512 tokens to Gemini 2.5 Pro at 2 million tokens-4,000× increase. This survey examines how position encoding methods such as RoPE scaling and ALiBi and architectural alternatives like state space models and hybrids allow for context window increases. A finding from most models is that few utilize an effective context length near what they boast. In reality, most exist within 10-50% of their effective context windows-RULER claims those boasting 128K effectively only use 32K without decay. The "lost in the middle" effect suggests systematic failure to retrieve whatever exists in the middle regardless of whether it exists in 0-128K or 0-32K of context. Thus, this survey compiles these findings across positional encoding methods, sparse attention methods, memory-augmented ones, state space and retrieval-augmented generation and computational optimization. Findings are taken from peer-reviewed literature across NeurIPS, ICML, ICLR, ACL and EMNLP to compile a useful overview for practitioners working with long context LLMs.***

Keywords: Large Language Models, Context Length, Positional Encoding, RoPE, ALiBi, Sparse Attention, State Space Models, Mamba, Retrieval-Augmented Generation, FlashAttention, KV Cache

1. Introduction

The ability to consume long sequence contexts powers many real-world applications for large language models (LLMs) from document summarization to code generation to multi-turn dialogue to retrieval-augmented question answering. When attention was first introduced with the original Transformer [1], it was set at $O(n^2)$ and with a reasonable context of 512 tokens. Yet in the seven years since its introduction, many techniques have been derived to exponentially increase this allowance when the latest models boast well over 2 million tokens [2].

Yet while it's theoretically possible to work with long contexts across implementations, it's not always a straightforward decision when developing LLMs. For one, computational complexity induced from naive implementation is $O(n^2)$ from the start. In addition, positional encodings trained at shorter lengths struggle with generalizability over LLMs for lengthened applies contexts. Most importantly, however, research shows that a vast majority of LLMs seldom use their potential context windows, anyway. The "lost in the middle" phenomenon [3] shows a systematic failure to retrieve information from the middle, even though it's clear information included in a set of 0–128K or 0-32K initial lengths.

This survey looks at the developments made possible relative to exponentially higher contexts and is separated into six overarching sections based on the following thematic elements:

1) Position Encoding Methods: RoPE, ALiBi and their derivatives for scaled RoPE and hybrid contexts as variations.
2) Sparse Attention Mechanisms: Longformer, BigBird and sliding windows plus memory independent options.
3) Memory-Augmented Architectures: Memorizing Transformers, MemGPT and Infini-attention methods that implement memory cross contexts.
4) State Space Models: S4, Mamba and RWKV hybrid architectures that take the place of standard attention implementations with state space attentions.
5) Retrieval-Augmented Generation: RAGs, RETROs, self-RAGs and hybrid methods which bring pre-existing data into play and outside resources for processes.
6) Computational Optimization: FlashAttention, KV Cache compression and distributed inference for ease of management/training speed that don't impair quality.

2. Position Encoding Innovations

The original Transformer utilized sinusoidal positional encoding at a maximum fixed context of 512 tokens [1]. The next generation followed with RoPE [4], which encodes absolute position using rotation matrices while simultaneously generating relative position using an attention mechanism itself. RoPE has been adopted by almost all new open source LLMs that have been developed like LLaMA [5], Mistral [6], Qwen [7], etc.

2.1 RoPE Scaling Methods

With RoPE came attempts to extend RoPE scaling beyond trained lengths as researchers employed positional downscaling opportunities. Position Interpolation [8] allows for linear down-scaling of position indices which extends the LLaMA 7B from 2K to 16K contexts with a mere 640 A100 GPUhours suggesting that based on interpolation within the trained space learned from a smaller scale is more useful than extrapolation beyond it.

Inductive NTK-aware scaling [9] suggests the opposite where high frequencies should be scaled down relative to lower

frequencies to accommodate sustained scaling posttraining application development. NTK-aware scaling works best for non-fine-tuned models attempting to develop since it scales differently than those already fine-tuned. NTK-aware scaling assesses the notion that within rotary embeddings there are multiple frequencies present with varying resolution at which information can be drawn from each direction.

In addition, YaRN (Yet another RoPE extensioN) [10] combines NTK-by-parts with attention temperature scaling extending Llama 2 7B up to 128K context with only 400 training steps–10× fewer tokens than Position Interpolation requires– the current state-of-the-art for extending RoPE available at ICLR 2024 at this time not yet compared against any other contenders as YaRN was accepted on its own at ICLR 2024 while currently state-of-the-art for RoPE extension.

2.2 ALiBi: Attention with Linear Biases

ALiBi [11] took a much different track by eliminating position embeddings altogether but instead adding static nonlearned bias that's strictly linear to each attention score that penalizes keys being queried if too far away; when steep enough as in the case of head m at positions i, j, it computes that:

$$\text{bias}_m(i,j) = -m \cdot |i - j| \quad (1)$$

Where m is a head specific slope conditioned on linearized strings ≤2K, ALiBi appropriately generalizes upwards without re-training and suggests proper memory use through lighter trained associated with implementations [12] being 11% faster with only 11% less memory utilization as such. Thus, BLOOM [12] and MPT [13] use ALiBi within their systems without needing retraining after trained context only need 1,024 tokens for those over 2K+.

In Table X we can see how different positions stack up according to their effectiveness thus far.

Table 1: Comparison of Position Encoding Methods

Method	Extrap.	Training	Models
Sinusoidal	Poor	Fixed	BERT, GPT-2
ALiBi	Good	None	BLOOM, MPT
RoPE	Moderate	Required	LLaMA, Mistral
YaRN	Excellent	Minimal	Extended LLaMA

3. Sparse and Memory-Augmented Attention

Standard attention grows at $O(n^2)$ computational complexity making access of long contexts daunting without sparse attention patterns and memory-augmented alternatives required instead.

3.1 Sparse Attention Patterns

Longformer [14] combines sliding window attention patterns with dilated patterns and $O(n)$ complexity supporting up to 4k tokens; with sliding windows providing local context while global-attention style tokens spread across the entire strided sequence allowing token movement through information referenced alternatives.

BigBird [15] adds random attention connections to windows and global patterns proving theoretically sparse attention across a universal approximator equivalent transformed function-to-sequence; this is important because it shows that big bird does not overly limit theoretical means model expressiveness although empirically challenged by cost/benefit analysis in practicality of need [16].

Those who implemented Mistral [6] utilized it as a sliding window attention working over 4k tokens but recognizes contextual allowance of up to 32k; it's critical because if each layer/token can only pay attention to previous 4096 tokens then there must be significance attributed to various layers interpenetrating multiplications through the total number of tokens suggesting a significantly greater maximum than which is found on any one token/layer basis alone +16M parameters suggested additional gains found independently per layer scaled appropriately if cumulative effective threshold was too good or bad to be true along with negative effective dimensions [+–], but [22]. [23].

Memory-Augmented Transformers [16] implement a *k*nearest neighbor (*k*NN) lookup into externalized memory (key,value); thus creating an extensive MM that goes beyond what standard attention could ever handle never mind previously set near-over-the-top loci in representation of analytically accurate perplexity up until 262K memory length as long as information flow exists so long as one accounts for various necessary handling operations crossing operations between different systems instead of acquiring their own across spatial bounds [64].

Augments standard attentional approach via retrieval from external diff'nable memory like so:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + kNN(Q,M)\right)V \quad (2)$$

Beyond low thresholded expectations like MemGPT [17] replicating operating system virtual memory paging [(1). paging allows certain pages/layers of specific parts of information/context are temporarily filled/unloaded which means user does not need constant access/layer needs not constant usage unless specifically helpful in any particular moment; (2). there may exist template-page like contoured areas deemed too troublesome as piecemeal substitutes during any moment even if useful enough; (3). there may be instances where now RYA must remove itself into "distraction" meaning that its knowledge becomes diminished or non-existent through shifted focus], allowing LLMs to operate within their own boundaries through function calls is extremely helpful relative to specific noting allowing for external storage options while certain pages go back "in" [20] between RAM/chip concerns making it possible for practically unbounded moments without context managed entirely autonomously as long as they can operate best value-free which is not often denoted best options saying machines fail at best value placement [21].

Ultimately Infini-attention [18] allows for compression embedded within vanilla attention without covering reliable none helpful reliability–IE∃ external memory processes both

suggests what's helpful from what inventory units suggest should be required instead but locked into memories' value [] without complex pipelines disrupted managed externally– Memory compression is special; There exists compressed external memory M where at transformation everything infinitely retains where each new X_k finds its value worth accessing however at every step M is updated which includes deletion.[...] $M_s = M_{s-1} + \sigma(K_s)^T V_s$ (3)

A 1B parameter model was able to steal the passkey back with 1 million tokens although it was only trained on 5k length inputs, a 114× compression compared to Memorizing Transformers.

4. Distributed Attention: Ring Attention

Ring Attention [19] is a way of achieving truly huge contexts through distributed processing.

By setting devices up in a ring topology and overlaps KV block passing for communication with attention calculation, it achieves near-linear scaling for as many devices as there are. The input sequence is divided across the devices and each device only needs to calculate the attention of its local query block with all key-value blocks. Key-value blocks are passed around the ring and attention calculations are achieved simultaneously which hides the latency of communication.

Ring Attention can manage 16 million token contexts on 512 A100 GPUs. Meta in 2024 achieved 1M context prefill for Llama3 405B in 77 seconds [20], suggesting practical implementation at such scale.

5. State Space Models

State space models are another approach entirely with $O(n)$ linear complexity and therefore, they are attractive for long contexts.

5.1 Structured State Spaces (S4)

S4 [21] was the first to introduce structured state spaces for sequence modeling, achieving $O(n)$ complexity by using a continuous time differential equation. The model is defined by the following state equation:

$$h'(t) = Ah(t) + Bx(t), \quad y(t) = Ch(t) + Dx(t) \qquad (4)$$

where A is set to the HiPPO matrix for ideal long-range dependency creation. S4 was the first model to complete the Path-X task (16,384 tokens sequences) in the Long Range Arena benchmark.

5.2 Mamba: Selective State Spaces

Mamba [22] extended S4 by filling in the major gap of not being able to do content-based reasoning via selective state spaces where the parameters become input dependent. Instead of static A, B, C matrices, Mamba uses:

5.3 RWKV and RetNet

RWKV [24] is a linear attention trained as an RNN or Transformer, bridging the gap between RNN efficiency and Transformer parallelism. RWKV is trained to 14B parameters (the largest dense RNN trained to date) with constant $O(1)$ inference cost and theoretically infinite context.

RetNet [25] provides the same advantages with its retention mechanism, 8.4× faster decoding than Transformers with 70% reduced memory usage. The retention mechanism can be computed recurrently (inference) and in parallel (training).

5.4 Hybrid Architectures

Thus far, hybrid models marrying attention and SSMs seem most promising. Jamba [26] uses a 1:7 blend of Transformer and Mamba layers with Mixture of Experts (MoE) to facilitate 256K effective context without KV cache that's only 8× smaller than vanilla Transformers.

Griffin [27] uses gated linear recurrences and local attention that, respectively, work together to meet Llama-2 performance with 6× less tokens trained on. The short-range local attention provides more accurate modeling while the recurrence efficiently manages long-range considerations.

Table 2: Architecture Comparison

Model	Complexity	Memory	Context
Transformer	$O(n^2)$	$O(n)$	Limited
Mamba	$O(n)$	$O(1)$	Unlimited
RWKV	$O(n)$	$O(1)$	Unlimited
Jamba	$O(n)$	$O(n/8)$	256K
Griffin	$O(n)$	$O(n)$ local	Long

6. Benchmarks and Evaluation

Extensively benchmarking and evaluating long-context abilities shows significant discrepancies between what's promised and what's delivered.

6.1 RULER Benchmark

The RULER benchmark [28] revealed some shocking results; only 50% of the models that claimed they could support 32K+ context continued to perform well up to 32K tokens. While every model performed with almost 100% accuracy in the needle-in-a-haystack retrieval task, every evaluated model experienced significant decline as context lengthened.

6.2 LongBench [29] is the first bilingual multi-task benchmark including 21 datasets where researchers found that effective retrieval for context compression aids weaker long-context models but not effectively enough to beat natively strong long context models. The tasks occur in six categories: single- document QA, multi-document QA, summarization, few-shot learning, synthetic and code completion tasks.

6.3 LongBench: Mamba-2 [23] demonstrated theoretical equivalence of SSMs and linear attention via the "structured state space duality" literature which allows 2-8× faster training than Mamba-1 for the purposes of tensor core operations.

Results include 5× higher inference throughput than Transformers, $O(1)$ per-step complexity (no KV cache, it's all

learned sequentially), and Transformer performance with 2× the size of Mamba.

Table 3: RULER Benchmark Results

Model	4K	32K	128K	Effective
GPT-4 Yi-34B	96.3%	93.2%	81.20%	128K
Mixtral))	94.7%	85.6%	-	32K
	94.90%	84.70%	-	32K
Command-R	92.40%	84.30%	-	32K

6.3 Lost in the Middle Phenomenon

The "lost in the middle" phenomenon [3] provided a Ushaped performance curve: when relevant information was located in the middle (or other middle-like positions) of the context, the model performed poorly, even to the most obvious long-context models' accessibility. This has serious implications for the design of RAG systems—order matters.

Liu et al. found a 20% performance drop when relevant information resided in the middle versus the beginning and end of context. This was true across smaller to larger model sizes (7B to 70B parameters).

6.4 BABILong

BABILong [30] found that popular LLMs access only 1020% of their context effectively, with degradation occurring after 10% of access and with GPT-4, in particular. The tasks extend to long-context versions of the classic bAbI tasks through implanted reasoning facts within distractor text.

7. Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) is an effective workaround to generalize an entire knowledge-base worth of context without increasing model parameters; it is an alternative to—and/or a companion to—long context windows.

7.1 Foundational RAG Approaches

RAG [31] was introduced by Lewis et al. at NeurIPS 2020 as a retriever (DPR) plus generator (BART), marginalized over retrieved documents, which proved successful on knowledge-intensive tasks without knowledge needing to be stored in model parameters.

RETRO [32] found that retrieval over 2 trillion tokens with a 7.5B model matched performance results of a GPT-3 (175B)-trained model—achieving comparable capability with 25× fewer parameters. RETRO inserts retrieved chunks into the Transformer computation through cross-attention layers.

Atlas [33] found performance results better than the PaLM (540B parameters) using 50× fewer parameters, proven successful for knowledge-intensive questions and answers.

7.2 RAG vs. Long Context

Where RAG versus long context is concerned, nuanced approaches provide insight into both claims. Self-Route [34] found that RAG consistently loses on average when long context has enough resources to be effective—but RAG is 1,250× cheaper per query ($0.00008 vs $0.10).

Order-Preserve RAG found that Llama3.1-70B achieved 44.43 F1 with 16K versus only 34.32 F1 with a full 128K context window; therefore, sometimes it's better to retrieve than stuff context.

7.3 Advanced RAG Methods

Self-RAG [35] measures generates reflection tokens that teach LLMs how to effectively engage with adaptive retrieval; this approach revealed that 7B models can outperform ChatGPT with open-domain QA using self-RAG's assessments about whether or not retrieval is best and self-critiquing retrieved passages/results.

RAPTOR [36] grows tree structures recursively summated over hierarchically positioned passages; its use with GPT4 achieves an absolute accuracy improvement of 20% over QuALITY since it captures low-level details and high-level themes, accessing them in meaningful tree-form structures.

8. Computational Optimization

Truly applying long-context models requires computational efficiency; training and inference optimizations present amazing opportunities for these models to hold up effectively.

8.1 FlashAttention

FlashAttention [37] revolutionized the study of attention with IO-aware algorithmic tiling where researchers no longer compute $O(n^2)$ attention matrix but instead compute attention on tiles for efficiency across SRAM processes instead of churning out larger pieces on slower HBM storage.

FlashAttention-3 [38] achieved 75% utilization of H100 theoretical max (v2 had only 35%); the solution achieves 740 TFLOPS with FP16 and 1.2 PFLOPS with FP8 due to asynchronous exploitation of tensor cores and low-precision capacity.

8.2 KV Cache Optimization

The KV cache grows linearly with context length since it holds key and value tensors from previous tokens; Llama 3 70B on 128K context has a KV cache alone requiring up to 40GB which approaches the size of model weights themselves and eat away at finite resources.

PagedAttention [39] in vLLM grows like an OS virtual memory; this helps reduce memory wastage from 60-80% down to <4% and increases throughput by 2-4×. This is possible since fragmented memory from previous iterations is no longer necessary; non-contiguous allocation is the best option going forward.

Grouped Query Attention (GQA) [40] is an important optimization for entry into KV cache—Llama-2-70B uses 8 KV heads shared across query heads whereby reduce cache size by 87.5% without sacrificing quality through multi-head attention.

Through quantization techniques like KIVI [41], 2-bit quantized KV cache can occur without loss of quality—in fact, it suggests support for up to 10M contexts due to reductions in memory requirements drastically.

8.3 Computational Scaling

Standard attention earns $O(n^2)$ complexity and FLOPs per context show drastic scaling: a 7B model requires about 8 TFLOPs at 4K context but requires about 500 PFLOPs at 1M context—62,500× more effort. This inspires sub-quadratic solutions that must be sought out.

Table 4: Optimization Techniques Impact

Technique	Speedup	Memory Red.
FlashAttention-3	2-4×	20×
PagedAttention	2-4×	55-65%
GQA (8 heads)	—	87.50%
KV Quantization	—	4-8×

9. Training-Free Context Extension

As per recent literature, context extension without additional costly continual pre-training is possible.

9.1. Self-Extend

Self-Extend [42] boasts only 4 lines of code change for 100% passkey recovery, utilizing bi-level attention that distinguishes between grouped (long-range) and neighbor (local) attention. It was accepted as an ICML 2024 Spotlight paper.

They essentially note that LLMs can have long-context capabilities "unlocked" instead of retrained from scratch. By employing a different position encoding for nearby vs. far away tokens, Self-Extend allows models trained on 4K context to essentially learn how to be applied to 16K+ without any fine-tuning.

9.2 LongLoRA

LongLoRA [43] uses shifted sparse attention to extend LLaMA 2 7B to 100K context on a single 8×A100 machine. LongLoRA optimizes and combines efficient attention patterns with parameter-efficient fine-tuning (LoRA) to reduce as much as possible the amount of computational resources needed.

9.3 Extending with Minimal Data

ExtendLLM [44] suggests that researchers overestimate the amount of training data needed for context windows to successfully extend; using only 100 samples, researchers extended context windows from 4K to 16K, challenging the conventional assumptions. They focus on well-curated longcontext examples that are of high-quality over large amounts of mediocre ones.

10. Current Production Landscape

Thus, the production landscape spans multi-tiered access to various lengths of context windows.

Leading the pack is Gemini 1.5/2.0 Pro with 2 million tokens and >99.7% recall on a needle-in-a-haystack at 1M tokens through MoE and sparse attention.[2] Claude 3/4 now offers 200K (1M in beta) without problems with extensive document analysis capabilities [45]. GPT-4 Turbo offers 128K although degradation exists after approximately 73K due to non-devilry usage patterns.[46] LLaMA 3.1 democratizes 128K context in open-source model reworks by using RoPE scaling [47]. Finally, Jamba 1.5 uniquely possesses an effective length of 256K through its hybrid Mamba Transformer architecture. [26]

Table 5: Production Model Context Lengths

Model	Claimed	Effective
Gemini 2.5 Pro	2M	~1M
Claude 3.5	200K	~150K
GPT-4 Turbo	128K	~73K
LLaMA 3.1	128K	~64K
Jamba 1.5	256K	256K

11. Quality Degradation Problems

Quality degradation remains an issue. Recent research [48] reports 13.9-85% quality degradation for input lengths from 512 to 2048, 4096 and up—for models that still have the active ability to generate every response with a factually correct understanding of all retrieves. Why? "Left-skewed position frequency distribution" during pretraining: position indices of long distance were under-trained to a highly catastrophic level.

This means two things: one, increasing context windows is not enough—models trained upon generated contexts with different position frequency distributions can better harness longer contexts; two, the pretraining corpus utilized is mostly filled with short documents meaning that long distance positions will always be under-trained.

12. Open Questions and Future Work

For long-context LLMs, the following questions remain open:

1) Effective Vs. Advertised Context: How to get and maintain what's advertised; there's no correlational study through RAG-like architecture that helps train.
2) Position Bias: The "lost in the middle" factor; either acknowledge there's nothing you can do or increase and make positional arc aware.
3) Computational Access: Linear path to linear attention where processors can truly manage no holds barred.
4) Evaluation Standards: There's not much beyond needlein-a-haystack awareness. We need more practical assessments.
5) Hybrid Mechanisms: Is retrieval better than attention for short contexts and reverse for long? Should they be combined? Is SSM preferred?

13. Conclusion

This survey provides an intricate overview of how LLMs have allowed longer and longer contexts to be engaged from the original approach of 512 tokens to new models boasting 2 million tokens max. Literature on the matter compiles to the following practical takeaways.

1) Never believe whatever a model will tell you about effective context. Prepare for 50% effective reliance upon sophisticated evaluators like RULER.
2) Position means everything: don't put what's most relied upon in the middle because it won't be seen (before logical conclusion)—it should be at the beginning or the end.
3) Hybrid solutions come out on top as time proceeds—from Self-RAG and RAPTOR for retrieval and long context to Jamba and Griffin for SSM/attention hybrids and speculative decoding/kV compression.
4) For models already in production that work best together, FlashAttention-3, GQA, Paged Attention and quantization through kV cache have proven best performance dividends to date from experience.
5) If context requires compression, Lingua LLM compresses by up to 20× with minor performance degradation.
6) RAG vs long context comes down to purpose—if knowledge frequently changes, RAG is better (and therefore cheaper) but if coherent documentation is required, long context is preferred. However, ordering considerations must be paid to full contextualization instead of spreading it out since fragmentation diminishes awareness. Relative to the near future, by 2024-2025 distinctions between types will fade as SSMs accomplish linear attention as effective trained-free extended approaches boast similar results compared to expensive yet constantly pre-trained hybrid models showing that adding together mechanisms may yield better success than singular assignments.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.
[2] Google DeepMind, "Gemini 1.5: Unlocking Multimodal Understanding Across Millions of Tokens of Context," *Technical Report*, 2024.
[3] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the Middle: How Language Models Use Long Contexts," *Trans. Assoc. Comput. Linguistics*, vol. 12, pp. 157–173, 2024.
[4] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "RoFormer: Enhanced Transformer with Rotary Position Embedding," *Neurocomputing*, vol. 568, 2024.
[5] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.
[6] A. Q. Jiang et al., "Mistral 7B," *arXiv preprint arXiv:2310.06825*, 2023.
[7] J. Bai et al., "Qwen Technical Report," *arXiv preprint arXiv:2309.16609*, 2023.
[8] S. Chen, S. Wong, L. Chen, and Y. Tian, "Extending Context Window of Large Language Models via Positional Interpolation," *arXiv preprint arXiv:2306.15595*, 2023.
[9] Reddit user bloc97, "NTK-Aware Scaled RoPE," *Reddit post*, 2023.
[10] B. Peng, J. Quesnelle, H. Fan, and E. Shippole, "YaRN: Efficient Context Window Extension of Large Language Models," in *Proc. ICLR*, 2024.
[11] O. Press, N. A. Smith, and M. Lewis, "Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation," in *Proc. ICLR*, 2022.
[12] T. L. Scao et al., "BLOOM: A 176B-Parameter OpenAccess Multilingual Language Model," *arXiv preprint arXiv:2211.05100*, 2022.
[13] MosaicML, "Introducing MPT-7B: A New Standard for OpenSource, Commercially Usable LLMs," *Blog Post*, 2023.
[14] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The LongDocument Transformer," *arXiv preprint arXiv:2004.05150*, 2020.
[15] M. Zaheer et al., "Big Bird: Transformers for Longer Sequences," in *Proc. NeurIPS*, 2020, pp. 17283–17297.
[16] Y. Wu et al., "Memorizing Transformers," in *Proc. ICLR*, 2022.
[17] C. Packer, S. Wooders, K. Lin, V. Fang, S. G. Patil, I. Stoica, and J. E. Gonzalez, "MemGPT: Towards LLMs as Operating Systems," *arXiv preprint arXiv:2310.08560*, 2023.
[18] T. Munkhdalai, M. Faruqui, and S. Gopal, "Leave No Context Behind: Efficient Infinite Context Transformers with Infiniattention," *arXiv preprint arXiv:2404.07143*, 2024.
[19] H. Liu, M. Zaharia, and P. Abbeel, "Ring Attention with Blockwise Transformers for Near-Infinite Context," *arXiv preprint arXiv:2310.01889*, 2023.
[20] Meta AI, "Context Parallelism for Scalable Million-Token Inference," *arXiv preprint arXiv:2411.01783*, 2024.
[21] A. Gu, K. Goel, and C. Ré, "Efficiently Modeling Long Sequences with Structured State Spaces," in *Proc. ICLR*, 2022.
[22] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," in *Proc. ICLR*, 2024.
[23] T. Dao and A. Gu, "Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality," in *Proc. ICML*, 2024.
[24] B. Peng et al., "RWKV: Reinventing RNNs for the Transformer Era," in *Proc. EMNLP Findings*, 2023.
[25] Y. Sun et al., "Retentive Network: A Successor to Transformer for Large Language Models," *arXiv preprint arXiv:2307.08621*, 2023.
[26] O. Lieber et al., "Jamba: A Hybrid Transformer-Mamba Language Model," *arXiv preprint arXiv:2403.19887*, 2024.
[27] S. De et al., "Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models," *arXiv preprint arXiv:2402.19427*, 2024.
[28] C.-P. Hsieh et al., "RULER: What's the Real Context Size of Your Long-Context Language Models?" *arXiv preprint arXiv:2404.06654*, 2024.
[29] Y. Bai et al., "LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding," in *Proc. ACL*, 2024.

[30] Y. Kuratov et al., "BABILong: Testing the Limits of LLMs with Long Context Reasoning-in-a-Haystack," in *Proc. NeurIPS*, 2024.
[31] P. Lewis et al., "Retrieval-Augmented Generation for KnowledgeIntensive NLP Tasks," in *Proc. NeurIPS*, 2020.
[32] S. Borgeaud et al., "Improving Language Models by Retrieving from Trillions of Tokens," in *Proc. ICML*, 2022.
[33] G. Izacard et al., "Atlas: Few-shot Learning with Retrieval Augmented Language Models," *J. Mach. Learn. Res.*, vol. 24, no. 251, pp. 1–43, 2023.
[34] P. Li et al., "Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach," *arXiv preprint arXiv:2407.16833*, 2024.
[35] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "SelfRAG: Learning to Retrieve, Generate, and Critique through SelfReflection," in *Proc. ICLR*, 2024.
[36] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "RAPTOR: Recursive Abstractive Processing for TreeOrganized Retrieval," in *Proc. ICLR*, 2024.
[37] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness," in *Proc. NeurIPS*, 2022.
[38] T. Dao, "FlashAttention-3: Fast and Accurate Attention with Asynchrony and Low-precision," *arXiv preprint arXiv:2407.08608*, 2024.
[39] W. Kwon et al., "Efficient Memory Management for Large Language Model Serving with PagedAttention," in *Proc. SOSP*, 2023.
[40] J. Ainslie et al., "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints," in *Proc. EMNLP*, 2023.
[41] Z. Liu et al., "KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache," *arXiv preprint arXiv:2402.02750*, 2024.
[42] H. Jin et al., "LLM Maybe LongLM: Self-Extend LLM Context Window Without Tuning," in *Proc. ICML*, 2024.
[43] Y. Chen et al., "LongLoRA: Efficient Fine-tuning of LongContext Large Language Models," in *Proc. ICLR*, 2024.
[44] W. Luo et al., "Extending LLMs' Context Window with 100 Samples," *arXiv preprint arXiv:2401.07004*, 2024.
[45] Anthropic, "The Claude 3 Model Family: A New Standard for Models," *Technical Report*, 2024.
[46] Intelligence," *Technical Report*, 2024.
[47] OpenAI, "GPT-4 Technical Report," *arXiv preprint* Despite Perfect Retrieval," *arXiv preprint arXiv:2510.05381*, *arXiv:2303.08774*, 2023. 2025.
[48] Meta AI, "Llama 3.1: The Next Generation of Open Foundation
[49] Y. Zhang et al., "Context Length Alone Hurts LLM Performance