A Comparative Analysis of Long-Context Scaling With State Space Machines

David Zhang
Columbia University
dwz2107@columbia.edu

Andrew Li
Columbia University
ayl2159@columbia.edu

Yuqi Zhang
Columbia University
yz5072@columbia.edu

Abstract—**Large Language Models (LLMs) are being ubiquitously used across industries, and that trend is expected to continue. With the rise of longer documents, conversations, and reasoning traces, it's becoming increasingly important to select the correct model architecture for a task that requires this efficient processing of unstructured data. Our paper is focused on two aspects: measuring the accuracy of Mamba, Transformer, and Hybrid Models on long-context multi-document QA questions, and measuring the latency, the Time to First Token (TTFT) and the throughput/Tokens per second (TPS) for models based on different architectures.**

This paper provides analysis of different LLM architectures over long-context regimes, which we wish to serve as a guiding resource for both researchers and engineers when choosing which model to use for their tasks. An associated GitHub repository collecting our experiments is available at: Long-Context Scaling Experiment.

Index Terms—**Mamba, Long-Context, State Space Machines, LongBench.**

I. Introduction

The capability to efficiently process large amounts of data has been increasingly important for Large Language Models (LLMs) to be useful for human work. From astronomers processing vast amounts of astronomical measurement data, to scholars having to scan a copious amount of relevant documents in order to gain understanding on a subject, we as humans are limited by our local, serial reading capabilities.

We have also seen the rise of agentic applications and with it, the desire to increase the length of tasks that AI agents can complete autonomously. METR measures that this metric has been consistently increasing exponentially over the past 6 years, with a doubling time of around 7 months. If this trend continues, then in under a decade, AI agents will be able to complete a large fraction of tasks that currently take humans days or weeks. [8]

In this paper, we present an analysis of long-context scaling across three architectural families: pure SSMs (Mamba, Mamba-Codestral), Hybrid models (Jamba, Nemotron-H), and standard Transformer models (Qwen, Llama). We evaluate these models at the 3B–8B parameter scale using the LongBench suite, on a single GPU for uniform testing and to simulate a constrained development environment. We notice a clear trade-off between throughput and generation quality. Although SSMs and hybrid models demonstrate superior decoding speeds and memory efficiency in longer contexts, they struggle to achieve output coherence and stability in those same regimes. While these SSMs solve the computational scaling problem, significant stability and robustness related to instruction tuning must be addressed in order to compete with Transformers in long-context tasks.

II. Related Work

We highlight work that is seminal to the new State Space Model design, such as Mamba, hybrid models such as Jamba, as well as benchmarks for long-context reasoning such as LongBench v2.

A. S6 Variants

We describe a brief overview of the Mamba class of SSMs, which we used when seeking to understand the architecture of hybrid transformer models.

- **Mamba: Linear-Time Sequence Modeling with Selective State Spaces** [5]: Mamba builds off of structured state space models by allowing the model to selectively propagate or forget information along the sequence length depending on the current token. This is similar to managing its own KV cache. It also introduces the parallel scan, which still allows for efficient parallel training despite no longer being able to use efficient convolutions. This sets the baseline for outperforming Transformers of the same size while being smaller and having faster inference.

- **Mamba-2 (Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality)** [4]: Mamba-2 introduces State Space Duality, which connects selective SSMs such as Mamba to attention variants such as linear attention. It also proposes a new architecture that has 2-8x faster inference due to a more efficient SSD algorithm as well as hardware-aware matrix multiplications. The architecture also supports tensor parallelism and grouped value attention. This makes Mamba more competitive while still having linear-time complexity in inference
- **Mamba-3 (Improved Sequence Modeling Using State Space Principles)** [2]: Mamba-3 focuses further on inference-first scaling. Previous Mamba versions struggled with maximizing hardware utilization during decoding. Here, there were three core improvements: a more expressive recurrence using trapezoidal discretization, a

complex-valued state update rule which is equivalent to RoPE, and a Multi-Input, Multi-Output (MIMO) formulation. This allows Mamba-3 to improve inference efficiency and quality given a fixed budget.

B. Long Context Benchmarks

We use long context benchmarks to measure the efficacy of different architectures over different input length regimes. While we acknowledge the importance of needle-in-a-haystack evaluations, we exclude BABILong from this specific study to focus on the more realistic traces multi-document trace lengths of LongBench v2; we leave the extreme cases of BABILong as an area of future work.

- **LongBench v2: A Second-Generation Long-Context Benchmark** [3]: Open-source, expanded successor to LongBench with standardized evaluation across retrieval, multi-document QA, summarization, code, and long in-context learning; we adopt its tasks to assess quality retention and scaling at extended context lengths.

- **BABILong: Testing the Limits of LLMs with Long Context Reasoning-in-a-Haystack** [6]: focus on needle-in-a-haystack environments for facts distributed in extremely long documents, useful for diagnosing memory and positioning issues.

III. Methodology

A. Model Selection

Our goal was to compare long-context behavior across three architectural families: pure State Space Models (SSMs), hybrid SSM-Transformer models, and standard Transformer baselines while keeping model scale and practical deployability within the constraints of our available hardware.

a) Pure SSM models (Mamba 2.8B, Mamba-Codestral 7B): We included **Mamba 2.8B** as a representative SSM-only model to isolate the strengths and weaknesses of state-space sequence modeling at long context lengths. As a pure SSM, Mamba provides a clean contrast against Transformer attention mechanisms and serves as a stress test for whether SSMs maintain quality and instruction adherence as context grows, without relying on attention-based retrieval.

Mamba-Codestral 7B is another model we measured to capture a practically important axis that is frequently relevant in long-context settings: code and structured generation. This model provides an SSM-family datapoint at roughly the same scale as our 7-8B baselines, helping us separate architectural effects (SSM vs attention) from parameter scaling.

b) Hybrid SSM-Transformer models (Jamba 3B Reasoning, Nemotron-H 8B): To probe the hypothesis that hybridization can recover some of the instruction-following and reasoning benefits of Transformers while retaining the long-context efficiency advantages often attributed to SSM components, we selected two hybrid models at different scales:

- **Jamba 3B Reasoning** [1] was chosen as a smaller hybrid that explicitly targets reasoning-style behavior. This lets us test whether the addition of attention blocks and instruction tuning can meaningfully improve qualitative instruction following relative to pure SSMs, even at modest parameter counts.
- **Nemotron-H 8B** [9] was selected as a larger hybrid point closer in scale to our 7-8B Transformer baselines. This reduces confounding from parameter count and provides a stronger comparison for whether hybrid architectures narrow the long-context quality gap against Transformers under similar compute budgets.

c) Transformer baselines (Qwen 7B, Llama 8B): Finally, we selected **Qwen 7B** and **Llama 8B** as strong Transformer baselines. These models are widely used, well-supported in common serving stacks, and provide competitive instruction-following behavior at the 7-8B scale. Using two Transformer baselines reduces the risk that conclusions are driven by idiosyncrasies of a single model family, and it anchors the interpretation of hybrid/SSM results against the de facto long-context modeling paradigm.

Collectively, this set of models provides coverage over the three key architectural families (SSM, hybrid, Transformer), and a reasonable degree of scale matching (most models clustered around 7-8B parameters). This design allows us to attribute observed differences in long-context performance more credibly to architectural choices and tuning objectives, rather than to trivial differences in model size or infeasible deployment settings. Certain models faced issues we describe later in the results sections, and were therefore not run for our benchmarking.

B. Hardware and Execution Environment

Transformer and hybrid model sweeps were conducted on Columbia University's Insomnia HPC cluster using a single NVIDIA A6000 GPU (48 GB VRAM) per run. Pure SSM (Mamba-based) profiling and LongBench v2 MC runs were executed on Google Colab (A100) due to runtime and dependency constraints; therefore we primarily compare *within-track scaling trends* rather than absolute throughput across different hardware.

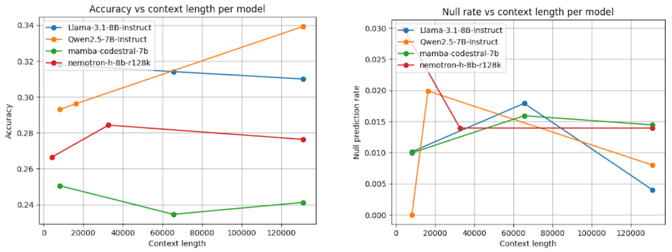
C. Inference Engine

Inference was performed using vLLM (v0.11.2) [7]. We use vLLM as a unified serving backend to ensure consistent execution across model families and to support long-context evaluation under a common memory-management and scheduling framework. This version of vLLM implements PagedAttention, which manages KV cache memory in non-contiguous blocks, effectively mitigating memory fragmentation and maximizing throughput for long-context sequences.

IV. Results

A. Accuracy

We evaluate four representative models:`Llama-3.1-8B-Instruct`, `Qwen2.5-7B-Instruct`, `mamba-codestral-7B`, and `nemotron-h-8b-r128k`. We measure metrics across increasing context lengths, focusing on task accuracy, output stability (null rate), and serving efficiency.



(a) Overall accuracy across context lengths

(b) Null output rate across context lengths

Fig. 1: Model performance trends as context length increases.

B. Pure SSM Track: Mamba-based Models (Colab)

a) Motivation and scope.: To isolate the long-context behavior of *pure* State Space Models (SSMs) without attention-based retrieval, we evaluated two public Mamba-family checkpoints: `mamba-2.8b` and `mamba-codestral-7b`. Our goal was twofold: (i) measure long-context multiple-choice QA accuracy, and (ii) characterize serving efficiency (throughput and memory) as context length increases. All pure-SSM runs were executed on Google Colab (A100 High-RAM) to support long-context inference and to simplify reproducible experiment tracking.

b) Technical approach (what was implemented).: We constructed a lightweight evaluation and profiling pipeline around existing prompt packs and model checkpoints:

- **LongBench v2 MC evaluation:** ran 4-choice multiple-choice evaluation at 8k/16k/32k contexts and computed accuracy from JSONL prediction logs (`longbench_mc_preds`).
- **Generation profiling (latency/efficiency):** for long-context generation workloads, we measured end-to-end throughput (*tokens/s*) and peak VRAM during batched inference on three representative datasets: `longbench_v2`, `pg19`, and `ada_bestanswer`.
- **Experiment tracking (W&B):** all runs were logged to a shared Weights & Biases project using consistent naming, tags, and config fields (model, dataset, context length), enabling apples-to-apples comparison across context regimes.

c) Results: LongBench v2 MC accuracy: Figure 2 reports multiple-choice accuracy vs. context length for the two Mamba-based models. The random baseline for 4-choice MC is ≈ 0.25. `mamba-codestral-7b` stays near this baseline at 8k/16k but drops slightly at 32k (0.255/0.250/0.235). `mamba-2.8b` is consistently below random (0.193/0.222/0.222). Overall, increasing context length from 8k to 32k does *not* reliably improve accuracy for these pure SSM models on LongBench v2 MC, suggesting that the dominant failure mode is not simply insufficient context capacity, but likely instruction-following / output reliability and/or task-specific tuning mismatch.

d) Results: throughput and memory scaling: Table I summarizes end-to-end throughput (tokens/s) and peak VRAM for generation profiling at 16k and 32k. Two consistent trends emerge. First, **memory footprint differs dramatically by model:** `mamba-2.8b` stays around ∼9–11 GB peak VRAM at 16k–32k, while `mamba-codestral-7b` peaks around ∼48–51 GB at the same context lengths. Second, **tokens/s does not necessarily decrease with longer context for pure SSMs.** In several runs, throughput is flat or even increases from 16k to 32k. This can occur because our reported tokens/s is an *end-to-end* measure (total tokens divided by wall time): longer sequences can better amortize per-request overhead (kernel launch, I/O, scheduling) and improve GPU utilization, which is consistent with SSM-style linear sequence processing. This behavior contrasts with standard Transformer decoding, where KV-cache growth and attention cost often lead to sharper throughput degradation at long contexts.

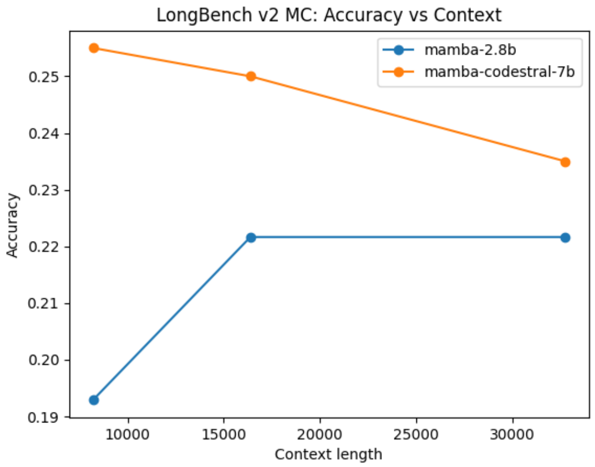


Fig. 2: LongBench v2 multiple-choice accuracy vs. context length for pure SSM models. Random 4-choice baseline ≈ 0.25.

Data	Model	TPS_{16k}	TPS_{32k}	VRAM_{16k}	VRAM_{32k}
LBv2-gen	Mamba-2.8B	6721	10607	9.44	10.75
LBv2-gen	Mamba-Cod7B	10250	10575	48.98	51.48
Ada-best	Mamba-2.8B	3283	3214	8.95	8.95
Ada-best	Mamba-Cod7B	7538	7506	48.12	48.12
PG19	Mamba-2.8B	905	1632	9.40	10.67
PG19	Mamba-Cod7B	$891^{\dagger}$	$1691^{\dagger}$	48.94	51.47

TABLE I: Pure SSM generation profiling at 16k/32k: end-to-end throughput (tokens/s) and peak VRAM (GB). †PG19 Codestral runs used fewer prompts due to runtime constraints; values are included as a rough reference.

e) Note on missing 8k runs: For several datasets, the available long-context prompt packs were primarily provided at 16k and 32k, while 8k variants were limited or derived by truncation in a few sanity checks. Since our main goal was long-context scaling behavior, we focus on the 16k/32k regime where the prompt sets are complete and comparable, and treat the limited 8k cases as validation/smoke tests rather than the primary scaling datapoints.

f) Takeaway: Pure SSMs show a clear *efficiency advantage* (especially memory footprint for `mamba-2.8b`) at long contexts, but do not yet show reliable quality improvements on LongBench MC as context increases. This supports the broader conclusion that architectural efficiency alone is insufficient; robustness and instruction-tuning remain key for long-context reasoning performance.

C. Accuracy Scaling with Context Length

Figure 1a shows overall accuracy as a function of context length. Among the evaluated models, `Qwen2.5-7B-Instruct` exhibits the strongest positive scaling behavior: accuracy increases steadily as context length grows, suggesting that the model effectively leverages additional context rather than being hindered by longer inputs. In contrast, `Llama-3.1-8B-Instruct` shows relatively stable accuracy with a slight degradation at the longest context length tested, indicating diminishing returns or mild instability as context grows.

Both `mamba-codestral-7B` and `nemotron-h-8b-r128k` underperform the Transformer baselines in absolute accuracy across all context lengths. The pure SSM model (`mamba-codestral-7B`) exhibits the lowest accuracy overall and shows limited improvement with additional context, suggesting that increased context length alone does not compensate for architectural or training limitations in long-context reasoning. The hybrid Nemotron model achieves intermediate accuracy, with a modest peak at mid-range context lengths followed by a decline at larger contexts.

D. Null Rate and Output Stability

Figure 1b reports the null prediction rate, capturing cases where outputs are unparseable or violate the expected answer format. Across models, null rates generally increase with context length, though the magnitude and pattern differ by architecture.

Transformer-based models show relatively controlled null rates, with `Llama-3.1-8B-Instruct` exhibiting a noticeable spike at intermediate context lengths before improving at the longest setting. `Qwen2.5-7B-Instruct` maintains comparatively low null rates overall, aligning with its strong instruction-following behavior observed qualitatively.

In contrast, the SSM-based and hybrid models show consistently higher and more persistent null rates. For `mamba-codestral-7B` and `nemotron-h-8b-r128k`, null rates increase with context length and remain elevated, indicating that a significant fraction of errors arise from output-format or instruction-following failures rather than purely incorrect reasoning. This trend reinforces the qualitative observations discussed earlier: smaller SSM and hybrid models are more prone to incomplete, ambiguous, or non-compliant outputs, especially under long-context prompts.

E. Serving Efficiency and Throughput

Figure 3 presents effective decoding throughput (tokens per second) across models and context lengths. As expected, throughput decreases substantially as context length increases for all models, reflecting increased memory pressure and KV-cache management overhead.

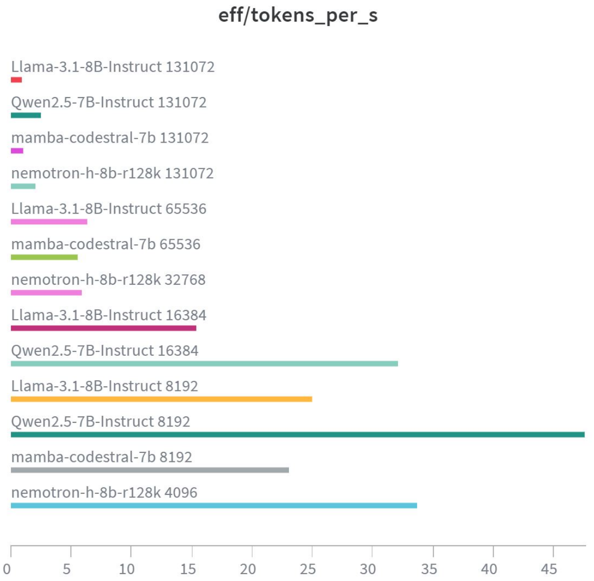


Fig. 3: Token per second output of selected models + Context length

Transformer models exhibit the steepest throughput degradation at long contexts, particularly at 65K and 131K tokens, where decoding speed drops sharply. In contrast, SSM-based and hybrid models demonstrate comparatively better throughput retention at large context lengths, consistent with their linear-time sequence modeling properties. However, this efficiency advantage does not translate into superior end-to-end task performance, as these models simultaneously suffer from lower accuracy and higher null rates.

F. Trade-offs Between Accuracy, Stability, and Efficiency

Taken together, these results highlight a fundamental trade-off. Transformer-based models, particularly `Qwen2.5-7B-Instruct`, achieve higher accuracy and better instruction-following robustness at the cost of rapidly declining throughput as context length increases. SSM-based and hybrid models offer improved scaling in serving efficiency but currently lag in accuracy and output reliability, especially under strict evaluation constraints.

These findings suggest that, at present, long-context performance is limited not only by memory and systems considerations but also by instruction-following robustness and training mismatch in smaller and SSM-based models. Consequently, improvements in data curricula, instruction tuning, and hybrid architectural design are likely necessary before the efficiency advantages of SSMs can be fully realized in long-context reasoning workloads.

G. Instruction-Following and Output-Format Failures in Smaller Models

In addition to aggregate accuracy trends, we observed a recurring failure mode in smaller models (particularly lightweight SSM-based and hybrid models): imperfect instruction following under strict output constraints. Because our evaluation requires selecting a single option (A-D) and producing a parsable final answer, deviations from the expected format can directly inflate the null/reprompt rate and confound accuracy measurements.

a) Qualitative examples.: A representative case is a multiple-choice translation prompt of the form: *"What is the correct answer? Choices (A)-(D) ..."* For this item, Mamba2.8B produced a response that enumerated multiple contradictory selections (explicitly stating different "correct answers" across options) before ending with a final choice:

> "The correct answer is (A)...The correct answer is (B)...(C)...(D)...The correct answer is (D)..."

This behavior violates the single-answer instruction and makes the output ambiguous for automatic grading, even if one of the mentioned options matches the label.

Similarly, the Jamba-3B reasoning model frequently failed to complete the task, instead beginning an analysis and then emitting an incomplete continuation cue (e.g., "We need to..."), which triggers our parse-failure handling and reprompt logic. In practice, such partial outputs behave as null predictions under our evaluation script.

Both Nemotron and Codestral demonstrated proper instruction following, as larger 7b/8b models, which led us to believe that this behavior was related to the model size. To test this hypothesis, we also tested Zamba 2.7b model which produced improperly formatted responses similar to Mamba, while the larger Zamba 8b model produced proper responses in accordance with the LongBench format.

Across multiple long-context sweeps, these failures were more prevalent in smaller models and in SSM/hybrid families than in stronger Transformer instruction-tuned baselines. While we do not claim causality from these qualitative examples alone, the pattern suggests that a non-trivial fraction of measured errors arises from instruction compliance and formatting reliability rather than pure task reasoning ability. This is particularly important for long-context evaluation where prompts are longer, parsing constraints are stricter, and models may be more likely to drift into verbose or inconsistent completions.

b) Impact on metrics and interpretation.: These instruction-following failures affect:

- **Accuracy (`acc/overall`):** Ambiguous multi-answer outputs are marked incorrect or unparseable, lowering measured accuracy even when the model "mentions" the correct option.
- **Null rate / reprompt rate:** Incomplete or non-compliant outputs increase the frequency of reprompts and null predictions, which can mask underlying competence on tasks where the model would otherwise succeed.
- **Long-context stability:** Format drift appears to worsen as context length increases for some smaller models, complicating direct comparisons that assume identical evaluation conditions.

These observations highlight that, in long-context benchmarks with strict output requirements, instruction-following robustness becomes a first-order factor, especially for smaller models. As a result, accuracy comparisons between model families should be interpreted with caution: some fraction of the gap between smaller SSM/hybrid models and larger instruction-tuned Transformers may reflect differences in *output controllability* rather than purely differences in long-context reasoning. This motivates future work on instruction-tuning and format-constrained decoding for smaller and SSM-based models (Section V).

Conclusion

In this paper, we present an analysis of long-context scaling across State Space Models (SSMs), Hybrid architectures, and Transformer models. Specifically, we evaluated representative models on the 7B parameter scale using the LongBenchv2 benchmark to isolate architectural strengths under constrained hardware resources. We measure performance on three axes: task accuracy, model throughput, output stability, and serving throughput. We find that SSM and Hybrid models do achieve linear-time efficiency and superior throughput at longer context lengths, but they lag behind Transformers in reasoning accuracy and require significant effort to get them to follow the instructions of given prompts and provide a straight answer. We conclude that while SSMs offer a promising solution to the memory bottleneck, more work is needed in instruction tuning and hybrid design for SSMs to match the reliability of the Transformer standard.

V. Future Work

Our current study provides an initial comparison of Transformer, pure SSM, and hybrid architectures under long-context evaluation. However, drawing strong conclusions about the relationship between context length, memory footprint, and accuracy requires substantially more data and broader experimental coverage. Below we outline several directions that would meaningfully extend this work.

A. Data Scaling and Curriculum Design for Long Context

A key limitation is that many models are not explicitly trained to operate robustly across a wide range of sequence lengths. Future work should include supervised fine-tuning (SFT) or continued pretraining under controlled *length curricula*, such as mixing long and short prompts (e.g., a 70/30 long-to-short mixture), to encourage both the use of long-range dependencies and stable short-context behavior. Systematically varying the prompt-length distribution during SFT would help identify whether observed long-context degradation is primarily architectural or data-driven.

B. Improving Instruction Following in Smaller Models

Our experiments suggest that smaller models are often less reliable at instruction following, particularly under strict output formatting and long-context conditions. Improving instruction-following capabilities for smaller parameter regimes would make them more viable for practical long-context deployments, such as certain agentic contexts. Future work should test parameter-efficient fine-tuning (PEFT) methods (e.g., LoRA-style adaptation) and targeted instruction-tuning datasets with format constraints, with evaluation emphasizing parse reliability and compliance under long prompts. This is particularly important for SSM-based and hybrid models, where the current available models are relatively smaller compared to modern transformers.

C. Mixture-of-Experts as a Long-Context Frontier

Mixture-of-Experts (MoE) architectures have become increasingly prominent for scaling capacity without proportional increases in compute. A natural extension is to evaluate MoE models in the same long-context setting and compare their memory/accuracy trade-offs against dense baselines. In particular, hybrid designs such as NVIDIA's Nemotron 3 Nano (a hybrid Mamba-Transformer MoE architecture) motivate studying whether sparsity and hybrid sequence modeling provide complementary benefits at long contexts. This would also require careful reporting of both quality metrics and systems metrics, since expert routing and activation sparsity can significantly affect throughput and memory behavior.

D. Evaluation Beyond Single-Hop Accuracy: Multi-Hop Reasoning

LongBench-style tasks capture a broad range of long-context behaviors, but do not fully isolate multi-hop retrieval and reasoning. We plan to add evaluation suites focused on multi-hop reasoning, measuring 2-hop or n-hop retrieval success rates and end-task correctness as hops increase. This would better stress compositional retrieval and aggregation, which are common failure modes in long-context applications. Coupling these evaluations with instrumentation of retrieval positions (where relevant) would provide more diagnostic insight than aggregate accuracy alone.

Acknowledgment

We would like to thank the instructors for their guidance and feedback throughout this project. We are also grateful for the computational resources and infrastructure from Columbia's Insomnia cluster, as well as the open-source community for providing tools and libraries that made this work possible. Finally, we thank our peers for insightful discussions that directly and indirectly helped shape the direction and evaluation of this study.

References

[1] AI21 Labs. Jamba: A hybrid transformer–mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

[2] Anonymous. Mamba-3: Improved sequence modeling using state space principles. In *Submitted to Tenth International Conference on Learning Representations*, 2025. Under review as a conference paper at ICLR 2026.

[3] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1840–1875, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[4] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

[5] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[6] Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Y Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. *arXiv preprint arXiv:2406.10149*, 2024.

[7] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[8] METR. Measuring ai ability to complete long tasks. https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/, 03 2025.

[9] NVIDIA. Nemotron-H: A family of accurate and efficient hybrid mamba-transformer models. *arXiv preprint arXiv:2504.03624*, 2025.