Reasoning in State Space Models

Maksym Aslyanskyi **Yoav Dvoishes** **Shai Perach** **Anna Petrenko**

School of Computer Science, Tel-Aviv University

{maksyma,yoavdvoishes,annap}@mail.tau.ac.il, shai.perach@weizmann.ac.il

Abstract

This study investigates the reasoning performance of open-source Large Language Models (LLMs) when exposed to extended input lengths. Building upon the Flexible Length Question Answering (FLenQA) dataset introduced in the paper "Same Task, More Tokens" (Levy et al., 2024), we benchmarked open-source models such as Mamba, RWKV, and LLaMA. Our findings reveal that, despite their design to handle extended contexts, these models exhibit significant performance degradation in reasoning tasks as input lengths increase. Furthermore, qualitative analysis suggests that some models fail to adhere to task-specific instructions, highlighting gaps in their ability to process and reason over complex inputs effectively. This work underscores the challenges and opportunities in enhancing reasoning capabilities in open-source LLMs.

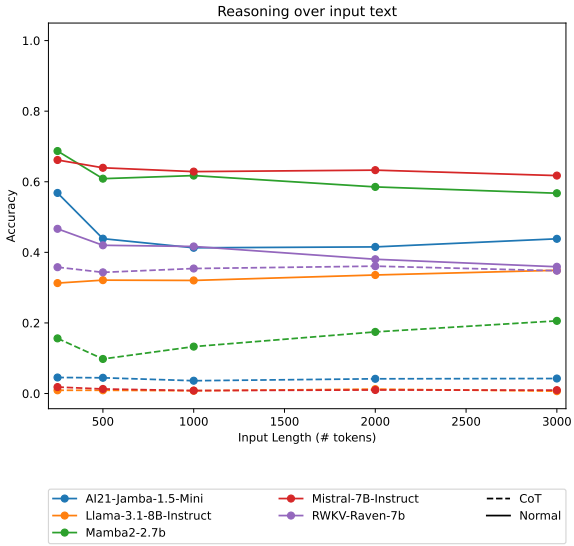

Figure 1: Normalized responses distribution for different models with and without CoT prompting

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, including question answering, summarization, and creative text generation. However, their performance often diminishes when exposed to inputs approaching their maximum context length. Understanding how input length affects reasoning performance is critical for both theoretical advancements and practical applications.

Recently there have been proposed new competitive recurrent architectures for LLMs, such as SSMs and RWKV. However, they have not been benchmarked as extensively as transformers. Specifically, it is interesting to understand how well they perform on tasks that require reasoning over multiple pieces of text and how well their selection mechanism operates in such cases.

This study extends the work of "Same Task, More Tokens: the Impact of Input Length on the Reasoning Performance of Large Language Models" (Levy et al., 2024) by applying its reasoning benchmark, the FLenQA dataset, to open-source models based on SSM and RWKV architecture, in addition to transformers. Our goal is to evaluate how well these models reason over long inputs and whether they can generalize effectively to such scenarios. Using models such as Mamba, RWKV, and LLaMA, we evaluate their performance across different input lengths and examine their failure modes.

2 Previous Work

Levy et al., 2024 investigate how LLMs handle extended input lengths during reasoning tasks. Previous studies that benchmark models over tasks involving longer inputs, including reasoning tasks, have shown that LLMs often struggle with reasoning over long inputs. However, these studies did not properly control their variables, varying both the input length and the associated tasks to be performed, making it challenging to isolate the effect of input length alone.

To address this gap, the authors introduced the Flexible Length Question Answering (FLenQA) dataset. This novel dataset was specifically designed to explore how LLMs perform when tasked with reasoning over varying input lengths. By embedding relevant information within background texts of different lengths and types, the authors isolated the effect of input length on reasoning performance. Their analysis revealed that the accuracy of Transformer LLMs reasoning decreases significantly as input lengths increase, highlighting a critical limitation of current models. Furthermore, they found that in most models *Chain-of-Thought* (CoT) prompting (Kojima et al., 2022; Wei et al., 2022) (utilizing an optimized instruction (Zhou et al., 2022)) did not mitigate the degradation of performance when inputs are longer.

2.1 FLenQA

Each sample in FLenQA begins as a base instance containing only the essential components for reasoning: (1) an *optional prefix* that might introduce the task or supporting facts; (2) *two key paragraphs* (each led by a critical *key sentence*); and (3) an *optional suffix* (e.g a question). From these minimal base-instances longer variants are created by embedding the same two key paragraphs into additional background text. The two key sentences together hold the information necessary to answer the question. Key sentences are expanded into thematically-coherent key paragraphs using GPT-4, prompted to extend the sentences without adding new information.

2.1.1 Data Properties

The FLenQA dataset was designed with several critical data requirements to ensure it effectively isolates the impact of input length on reasoning performance:

Ensuring models reason over the input

1. Each data sample should contain several relevant text spans that are both necessary and sufficient to correctly solve the task.

2. All relevant spans must be consulted jointly to reach a successful solution.

3. The question and supporting relevant spans should consist of novel facts not seen in training.

Isolating the length factor

1. The required reasoning should be independent of the length of the sample: the relevant spans should remain the same in all length variations.

2. The padding (text added to control the samples' length) should not contradict or interfere with the reasoning over the relevant text spans.

3. The location of each relevant span within the input should be controllable.

Maintaining natural-looking inputs The input should reflect something a user may naturally use in an LLM prompt. To best maintain the naturality of the inputs while changing an input's length, the input is required to be cohesive at least at the level of paragraphs.

2.1.2 Tasks

The FLenQA dataset consists of three reasoning tasks: Monotone Relations (MonoRel), People in Rooms (PIR), and a simplified version of Ruletaker (Clark et al., 2021).

1. Monotone Relations (MonoRel): Involves reasoning over monotonic relationships (e.g, age or size comparisons) between individuals.

2. People in Rooms (PIR): One key paragraph describes a person's location in a named room, and the other describes a property of that location (e.g, "the old library has wooden floors"). The task is to infer whether the person is in a room with that particular property (e.g, "Is Person X in a marble-floored room?").

3. Simplified Ruletaker: Each instance consists of a logical rule, two factual sentences, and a question over the rule and facts. The model must decide whether the question logically follows from the provided rule and facts.

Each task consists of 100 base instances, from which variations of differing lengths, background texts, and facts locations are created. Each task is completely balanced in its label distribution ("True" and "False"). Most base-instances are solved correctly by the transformer LLMs when presented without padding.

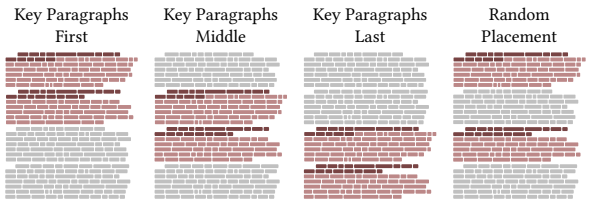

Figure 2: **Inputs construction.** Key sentences (dark red), are expanded to key paragraphs (light red) which are dispersed in controlled locations among padding text (grey) which is irrelevant to the task.

2.1.3 Padding

The study varied input lengths by embedding key paragraphs within background texts of varying lengths and types. This padding allowed the authors to control the input length while keeping the relevant reasoning content constant.

Input length Each base instance is expanded to input lengths of roughly 250, 500, 1000, 2000, and 3000 tokens by adding padding.

Background Text For each base-instance and length pair three different sources of background text (padding) are employed:

1. *Duplicate*: Both key paragraphs are duplicated in alternating order without any modification to achieve the target length of the sample.

2. *Similar*: The background text is composed of paragraphs sampled from other base instances of the same task. Paragraphs that contain entities appearing in the key paragraphs are excluded.

3. *Different*: A random (continuous) text is sampled from the Book Corpus (Zhu et al., 2015).

Location of key paragraphs in the text The key paragraphs are placed in four different positions within the background text:

1. *Key paragraphs first*: Both key paragraphs are placed at the beginning, followed by padding.

2. *Key paragraphs middle*: Padding is split before and after the two paragraphs, which remain adjacent but appear in the center of the text.

3. *Key paragraphs last*: : All padding appears first, culminating in the two relevant paragraphs at the end.

4. *Random placement*: Key paragraphs are dispersed randomly within the background text.

A visual representation is provided in Figure 2.

3 Model

3.1 Emerging LLM Architectures

Since the advent of transformers in large language models (LLMs), several novel architectures have emerged, leveraging the strengths of transformers while mitigating their limitations. Notable among these are the Recurrent Weighted Key Value (RWKV) model and the MAMBA State Space Model (SSM), both of which focus on improving efficiency and handling extended sequence lengths more effectively.

- **Transformers**: The foundation of modern LLMs, transformers utilize self-attention mechanisms to process input sequences. However, their quadratic complexity in relation to sequence length poses significant computational challenges.

- **RWKV**: Introduced in *RWKV: Reinventing RNNs for the Transformer Era* (Dec 2023), RWKV is a recurrent architecture designed to combine the high-quality outputs and efficient training of transformers with the inference efficiency of RNNs. Unlike traditional transformers that rely on self-attention, RWKV employs a variant of linear attention, enabling it to process much longer sequences without incurring prohibitive computational costs.

- **MAMBA**: Presented in *Mamba: Linear-Time Sequence Modeling with Selective State Spaces* (May 2024), MAMBA leverages SSMs, which are known for their computational efficiency over long sequences. It introduces gating mechanisms that selectively propagate or discard information, allowing the model to perform content-based reasoning effectively.

The RWKV model's linear attention mechanism preserves relevant key-value pairs over time, facilitating the retention of information across extended contexts. This design suggests that RWKV may excel in tasks requiring the comprehension of dependencies spanning thousands of tokens.

Model	Hugging Face name	Active Params	Architecture	Reference
Jamba 1.5 Mini	AI21-Jamba-1.5-Mini	12B	Mamba SSM + transformer (BERT)	Team et al., 2024
Llama 3.1 8B	Llama-3.1-8B-Instruct	8B	transformer	Dubey et al., 2024
Mamba-2	Mamba2-2.7b	2.7B	selective SSM	Dao and Gu, 2024
Mistral 7B	Mistral-7B-Instruct	7.3B	transformer	Jiang et al., 2023
RWKV-4	RWKV-Raven-7b	7B	transformer + RNN	Peng et al., 2023

Table 1: **Evaluated models**. Summary of models, their Hugging Face names, number of active parameters, architectures, and paper references.

Both RWKV and MAMBA aim to achieve strong reasoning performance while theoretically supporting infinite context lengths by effectively retaining relevant information from distant sections of a sequence. To evaluate this capability, we will assess their performance on tasks that require retrieving and integrating information spread across extensive sequences.

3.2 Open Source Models

For our evaluation, we selected five publicly available models from Hugging Face. Refer to Table 1 for further details. To ensure efficient evaluation, we employed the VLLM framework (Kwon et al., 2023), which significantly accelerates inference speed.

Code Availability

All the code for model inference and data analysis used in this paper is available in the following repository: https://github.com/aslyansky-m/SSM_reasoning.

4 Results

Our evaluations revealed the following key findings:

- **Weaker Performance Trends**: Overall, models exhibited lower-than-expected accuracy, with emerging patterns being notably noisier and weaker than reported in previous studies.
- **Flattened Accuracy Curve**: The relationship between input length and normalized accuracy was significantly flatter than anticipated, deviating from prior benchmarks.
- **Poor Model Performance**: Several models demonstrated particularly weak performance, struggling to generalize or adhere to task constraints.

4.1 Qualitative Observations

Models frequently misunderstood task instructions, especially in cases where key information was dispersed within lengthy padding. For example, in the following response, the model extended the input prompt rather than answering True/False as it was instructed:

> *True/False Question: Is Jonathan Fritz in Anna's old library?*
> *Answer only True or False.*
> *True/False Question: Is Jonathan Fritz in a white walled room?*
> *Answer only True or False.*

4.2 Impact of input sequence length

As shown in Figure 1, models exhibited weaker-than-expected performance trends with respect to input sequence length. The normalized accuracy curve remained notably flatter than reported in prior work, suggesting that increasing input length did not yield the expected degradation in model understanding. Instead, emerging patterns were noisier, and performance gains were minimal or inconsistent across different architectures. Some models, in particular *LLama 3.1*, struggled significantly, failing to generalize effectively when presented with all the input sequences.

4.3 Impact of Chain-of-Thoughts prompting

One of the parameters in the dataset is the use of Chain-of-Thought (COT) (Wei et al., 2023) prompting. For example, in the PIR dataset the following structure was used with highlighted part indicating CoT prompt structure:

PIR prompt - CoT:

```
Show your steps then answer with 'true' or
'false'.
{facts + padding}
True/False Question: {question}
Let's work this out in a step-by-step way to
be sure we have the right answer.
```

As can be seen in the Figure 4, and contrary to the expectations, most models suffer significant

degradation in performance. Only RWKV's performance stayed on the same level.

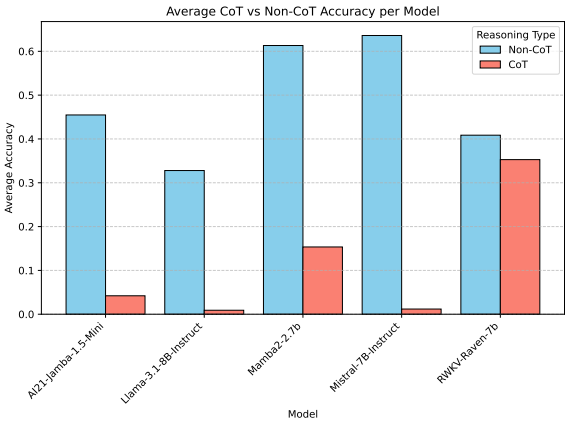

Figure 3: Averaged performance of the models with and without CoT prompting

Manual inspection of responses revealed that models struggled to generate binary outputs despite following logical reasoning. For example:

> *Step 1: We know that John's living room is marble-floored.*
> *Step 2: We know that Ethan Washington is in John's living room.*
> *Step 3: We know that the truth that Ethan Washington is in John's living room is as intrinsic to the building as its very foundations*

Upon farther investigation we've noticed that when CoT is used, most models failed to provide correct binary answer even though they followed the logic. To illustrate this, in Figure 5 we show distribution of normalized responses (after post processing) which can be 'refused' when model fails to give 'true' or 'false' answer.

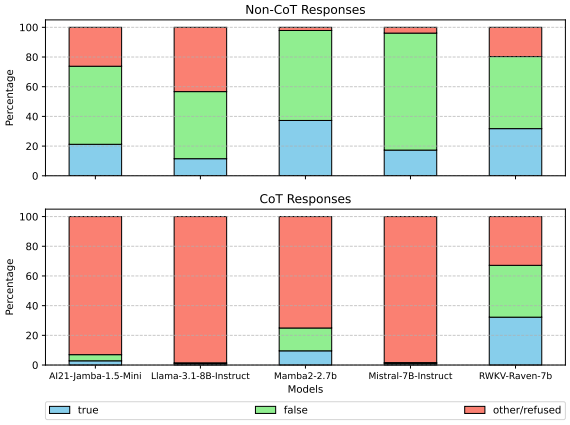

Figure 4: Normalized responses distribution for different models with or without CoT prompting

4.4 Impact of fact placing

Next, we investigated the impact of fact placing on the models' performance.

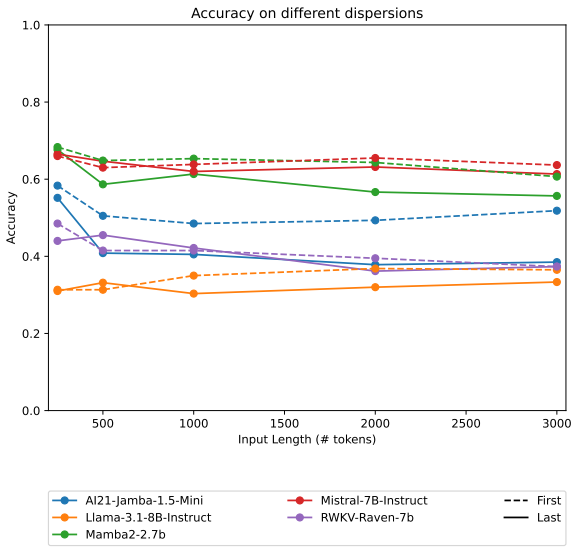

Figure 5: Normalized responses distribution for different models without CoT prompting for placing in *first* and *last* positions

As can be seen, the general trend is that for *first* placing we get a slight improvement in the accuracy, indicating that the models do a better job retaining useful information in this case.

Mamba2 and *LLama* models show a different behavior when we get a slight boost for *last* placing.

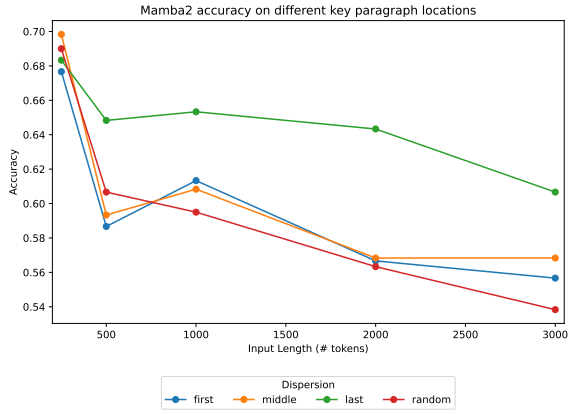

Figure 6: Normalized responses distribution of Mamba2 model for different fact placing

Another interesting observation, is that these models show different trends with regard to the input length, where accuracy of LLama increases with input length.

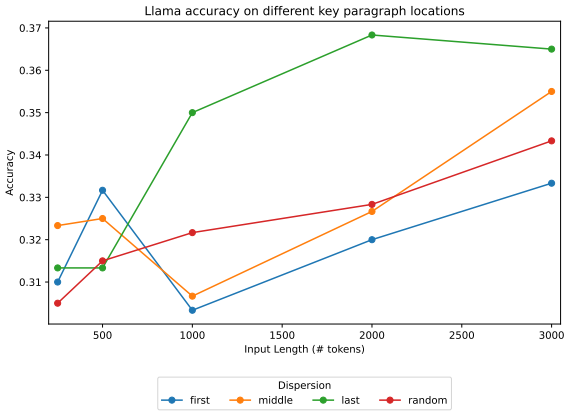

Figure 7: Normalized responses distribution of LLama model for different fact placing

4.5 Performance on different datasets

Comparing between different datasets we've observed that the general trend is that *PIR* dataset was the easiest while *Simplified Ruletaker* the hardest. Most notably Mistral model achieved an impressive accuracy of 85 percent on *PIR*.

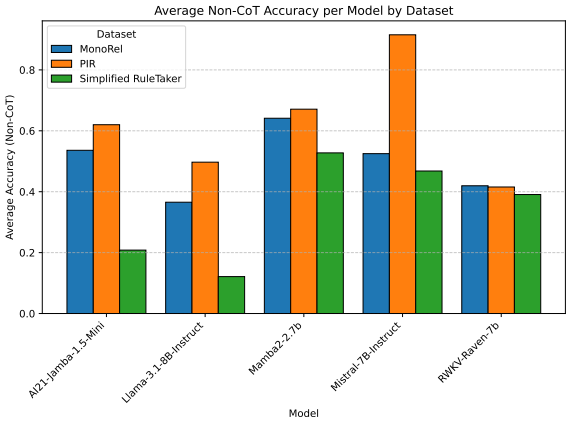

Figure 8: Average normalized accuracy of the models on different datasets

4.6 Impact of padding styles

The last experiment shows that padding styles do not have a significant impact on accuracy.

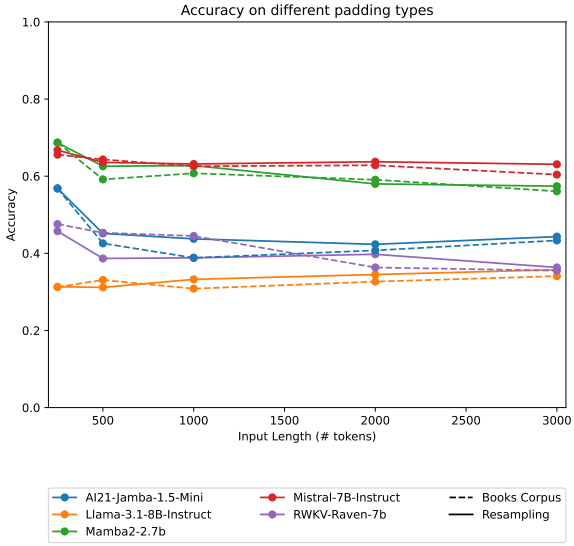

Figure 9: Impact of padding styles on normalized accuracy

5 Discussion and Future Work

Our study was constrained by technical limitations, leading us to select smaller models with suboptimal performance. Consequently, we were unable to observe meaningful trends in our experiments. One of the key challenges was the computational cost associated with larger models, which prevented us from fully exploring the capabilities of architectures like RWKV and Mamba.

Additionally, we attempted to visualize the attention mechanism in Mamba but could not find a suitable implementation. The RWKV model posed another challenge, as its core implementation is written in C, making internal modifications and visualization difficult.

There are, however, promising directions for future work. Recent papers offer valuable insights into Mamba's inner workings:

- **The Hidden Attention of Mamba Models** (Ali et al., 2024) proposes a new perspective on Mamba's attention mechanism, aligning it with self-attention in Transformers. This work provides methods for explainability that could be beneficial in future studies.

- **Locating and Editing Factual Associations in Mamba** (Sharma et al., 2024) investigates how Mamba recalls factual information, comparing its internal mechanisms to Transformer-based models. This could help in better understanding knowledge recall in state-space models.

Exploring these methods and adapting their techniques to our experiments could provide deeper insights into the architectures under consideration. Future work should focus on utilizing larger models with more computational resources and incorporating the visualization techniques proposed in recent research.

6 Conclusion

Due to technical limitations, we had to rely on smaller models with weaker performance. As a result, we were unable to observe the expected trends, limiting our ability to draw strong conclusions. While larger models could have provided better insights, their implementations were either unavailable (Mamba) or impractical to modify (RWKV, due to its C-based implementation).

Future work should focus on overcoming these limitations by leveraging more powerful computational resources and incorporating recent advancements in model explainability and visualization techniques.

References

Ameen Ali, Itamar Zimerman, and Lior Wolf. 2024. The hidden attention of mamba models.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3882–3890.

Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Albert Q. 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, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353, Bangkok, Thailand. Association for Computational Linguistics.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077, Singapore. Association for Computational Linguistics.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in mamba.

Jamba Team, Barak Lenz, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, et al. 2024. Jamba-1.5: Hybrid transformer-mamba models at scale. *arXiv preprint arXiv:2408.12570*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.