REVISITING ASSOCIATIVE RECALL IN MODERN RECURRENT MODELS

Destiny Okpekpe, Antonio Orvieto
Max Planck Institute for Intelligent Systems & ELLIS Institute Tuebingen, Tuebingen AI Center

ABSTRACT

Despite the advantageous subquadratic complexity of modern recurrent deep learning models – such as state-space models (SSMs) – recent studies have highlighted their potential shortcomings compared to transformers on reasoning and memorization tasks. In this paper, we dive deeper into one of such benchmarks: associative recall (AR), which has been shown to correlate well with language modeling performance, and inspect in detail the effects of scaling and optimization issues in recently proposed token mixing strategies. We first demonstrate that, unlike standard transformers, the choice of learning rate plays a critical role in the performance of modern recurrent models: an issue that can severely affect reported performance in previous works and suggests further research is needed to stabilize training. Next, we show that recurrent and attention-based models exhibit contrasting benefits when scaling in width as opposed to depth, with attention being notably unable to solve AR when limited to a single layer. Finally, by further inspection of 1-layer transformers, we reveal that despite their poor performance, their training dynamics surprisingly resemble the formation of induction heads, a phenomenon previously observed only in their 2-layer counterparts.

1 INTRODUCTION

Associative Recall. There is growing interest in benchmarking basic properties and capabilities of new language models (Gu & Dao, 2024; Nguyen et al., 2024) that use recurrent mechanisms as opposed to softmax attention (Vaswani et al., 2017). One key aspect of language modeling is the ability to recall previously encountered information. Following Arora et al. (2023), given the input

*"**Hakuna Matata** means **no worries** for the rest of your days. **Hakuna Matata** means ...",*

a well-performing model should predict ***"no worries"*** with high likelihood. Building on this idea, the synthetic associative recall (AR) task was introduced (Graves et al., 2014; Ba et al., 2016; Olsson et al., 2022), and gained popularity as an efficient reasoning benchmark to assess promising model architectures at a relatively low cost. Specifically, given a fixed vocabulary V, each sample consists of a sequence of tokens sampled from V representing alternating key-value pairs. Given such a sequence and a key that appeared earlier in such sequence, the model must correctly infer its corresponding value: For example, given the input sequence and the key:

$$A\ 6\ I\ 9\ \mathbf{C}\ \mathbf{7}\ P\ 1\ S\ 4\ D\ 2\ \mathbf{C} \rightarrow\ ?$$

the model should predict **7**. A crucial aspect of this task is that the tokens serve interchangeably as keys and values—they are drawn from the same vocabulary rather than separate sets. Consequently, the model cannot rely on preassigned roles for tokens. Moreover, since token roles and positions vary, the model cannot memorize a fixed mapping but must instead infer the correct associations dynamically in-context. Since performing AR can be seen as a fundamental prerequisite for reasoning, we expect that models capable of solving this synthetic task are promising LLM when scaled.

Multi-Query Associative Recall. Building on previous research (Arora et al., 2023), our experiments employ a variant of AR known as multi-query associative recall (MQAR). There are two key distinctions between MQAR as implemented in Arora et al. (2023) and standard implementations of AR, both of which align more closely with natural language. First, Arora et al. (2023) introduced

a significantly larger vocabulary: from the 50 tokens of standard AR to approximately 8000 tokens in MQAR. This makes the task more representative of real-world language processing where the vocabulary size is in the order of hundreds of thousands of tokens. Second, instead of recalling a single key-value pair, the MQAR definition in Arora et al. (2023) requires the model to retrieve multiple values based on multiple queries. This more accurately mirrors the nature of language, where meaning is often derived from groups of words and interrelated concepts rather than isolated tokens. For instance, given an input sequence and multiple keys:

$$\mathbf{A}\ \mathbf{6}\ I\ 9\ \mathbf{C}\ \mathbf{7}\ P\ 1\ S\ 4\ \mathbf{D}\ \mathbf{2}\ \mathbf{C} \rightarrow\ ?\ \mathbf{A} \rightarrow\ ?\ \mathbf{D} \rightarrow\ ?$$

we ask the model to recall the relative values **7**, **6** and **2**. While all of our analyses are made using MQAR, throughout this note we use the terms AR and MQAR interchangeably for simplification.

2 Closer Look into AR performance

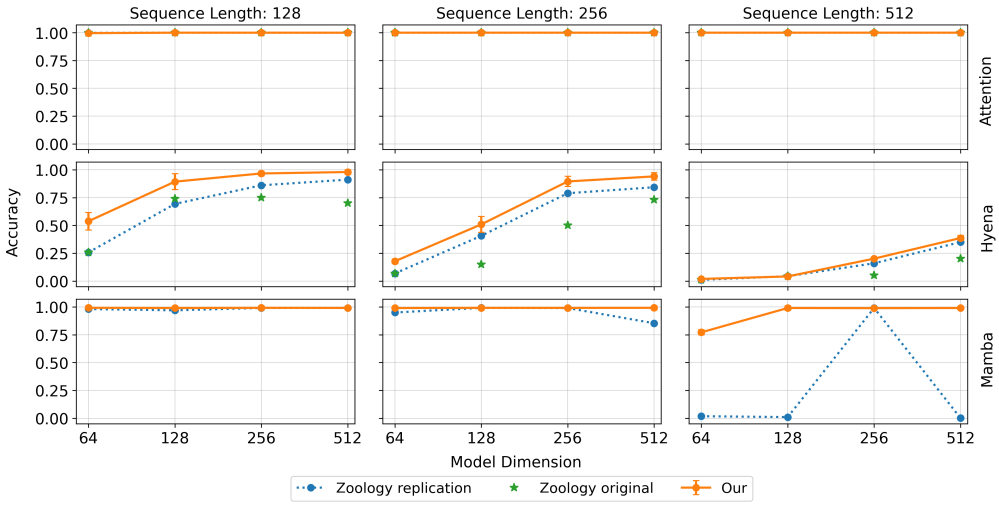

Figure 1: *Performance of 2-layers models. Results for H3 (Fu et al., 2023), RWKV (Peng et al., 2023) and Based (Arora et al., 2024) included in App. A.3. We report the official results*[1] *(green stars) and the replication running the original code of (Arora et al., 2023) (dotted blu line). While for replication, we used the learning rates grid by Arora et al. (2023), we note here that, due to high sensitivity to the learning rate (Fig, 2), tuning drastically affects performance. In solid orange, we provide results with a finer grid (cf. Fig.2). Careful tuning of the learning rate gives a general improvement in the performance of recurrent models. This becomes especially crucial in Mamba, where the task becomes solvable at high sequence lengths >> hidden size. The results show the mean and relative max-min errors for 3 seeds. Attention always solves the task (all curves overlap).*

Building on previous research, we aim to provide an in-depth analysis of the differences and similarities between attention and recurrent models through the lens of AR. Prior studies (Arora et al., 2023) have shown that transformers are inherently well-suited for solving the MQAR task, achieving perfect accuracy regardless of model dimension, sequence length or number of key-value pairs to infer. In contrast, it was argued (both theoretically and empirically) that new recurrent models (Peng et al., 2023; Nguyen et al., 2024; Gu & Dao, 2024) can only solve MQAR if the hidden dimension is roughly equal to the sequence length (see analysis by Jelassi et al. (2024) in a related setting). However, a key aspect that has been **overlooked** in some prior works is the crucial role of optimization in recurrent models —particularly, the use of an effective grid search for the choice of learning rate.

Hypothesis from previous works. Recurrent models update their hidden state (which serves as a compressed representation of past information) at each time step, using the current input. Since the model only has access to its hidden state and the current input, its ability to recall previous information depends on how effectively it compresses past data into this state. With a simplified analysis assuming uniform distribution over strings, Jelassi et al. (2024) showed that to successfully copy

[1]Mamba was not included in the official work but some experiments are documented in the blog post

input strings, the hidden size needed grows linearly with the sequence length. In contrast, transformers (Vaswani et al., 2017) dynamically access all previously seen inputs through the softmax attention mechanism, allowing for the explicit computation of interactions between tokens. This makes the task of recalling already seen tokens essentially a lookup table problem when two layers work simultaneously, as described in Jelassi et al. (2024); Olsson et al. (2022) (see App. A.1).

Results. Compared to previous work, in our experiments, we devoted more attention to tuning the learning rates, drastically improving the reported performance for recurrent models (see Fig. 1&2). As shown in Figure 1 and extensively in Appendix A.3, a finer grid not only enhances average performance across all settings but also proves particularly crucial for the Mamba model. With a more suitable learning rate, Mamba (Gu & Dao, 2024), which was previously shown to struggle with long sequence lengths, becomes capable of solving MQAR at relatively small hidden model sizes. All experimental details for this and the next experiments are in Appendix A.2. This highlights a key takeaway for MQAR: the choice of learning rate (and optimization strategy in general) can be decisive in assessing whether a recurrent model can solve the task at all.

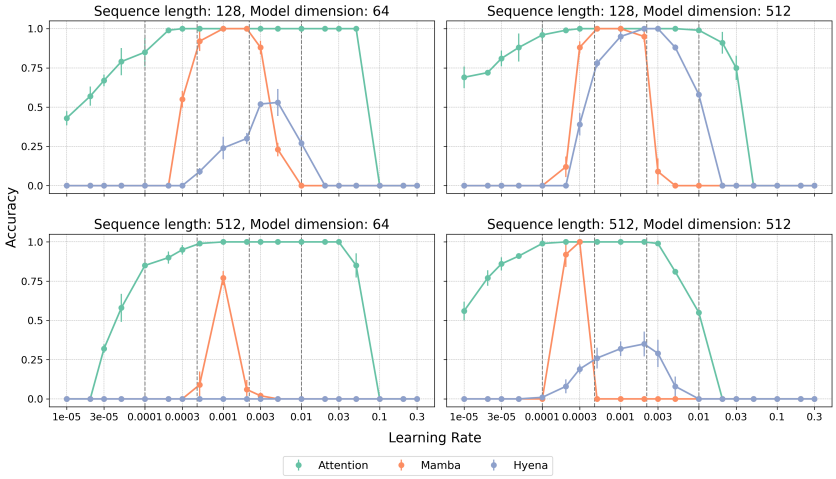

Figure 2: *We show the performance of Attention, Hyena and Mamba using the same learning rate grid search. Differently from attention, the window of suitable learning rates for Mamba and Hyena is relatively narrow. We also compare our grid search with the one used in Zoology (Arora et al., 2023) (**dashed vertical lines**) to highlight how the suitable learning rate can be missed. The results show the mean and relative max-min errors after 3 runs with different seeds.*

Figure 2 illustrates that attention-based models maintain strong performance across a relatively wide range of learning rates. In contrast, Hyena and Mamba exhibit a different behavior: performance remains near zero for most learning rates but suddenly reaches near-optimal levels at specific values which may not be included in the grid by Arora et al. (2023). These findings highlight a key distinction between attention-based and recurrent models: a sparse learning rate grid search can disproportionately impact their training outcomes. **This discrepancy can lead to misleading conclusions** about the capabilities of these models, emphasizing the need for careful tuning.

3 Effects of Width/Depth Scaling into AR Performance

While our findings in Sec. 2 show that some recurrent models can exhibit improved performance on MQAR with proper learning rate tuning, we confirm that a sizeable gap with attention can still be observed for some recurrent models at low widths (e.g. Hyena vs. Attention). The experiments of Sec. 2 focused on comparisons of 2-layer architectures, at different sequence lengths and model widths. This choice stems from prior research (Olsson et al., 2022), where transformers have shown peculiar in-context learning capabilities related to the formation of induction head circuits in 2-layer models. With the intention of going beyond the setup that is known to show strengths for softmax attention, our objective in this section is to explore the effects of scaling in different configurations.

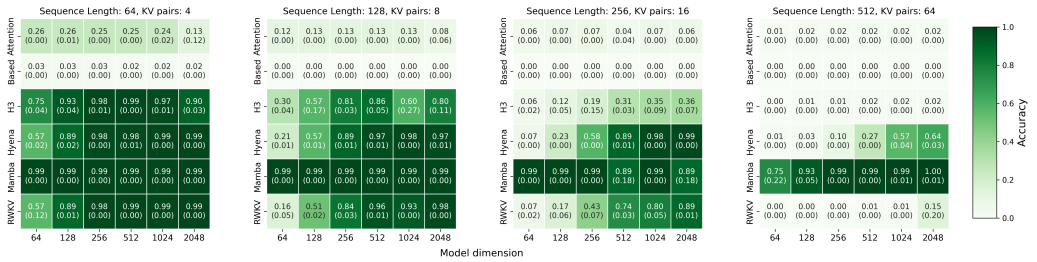

Figure 3: *Performance of 1-layer attention-based (Attention, Based) and recurrent-based (H3, Hyena, Mamba, RWKV) models on AR. We show how for recurrent models, scaling the width boosts performances. On the contrary, attention models cannot solve the task anymore as in the 2-layer setting, and performances are unaffected by the scaling in width. The results show the mean and relative max-min errors after 3 runs with different seeds.*

To achieve this goal, we conducted experiments analogous to Section 2 using single-layer architectures[2]. By doing so, we aim to decouple the effects of inter-communication between layers and to isolate the impact of each model's fundamental structure (attention versus recurrence) on MQAR. Beyond this, our motivation also comes from the notable connections that have been drawn between attention and recurrent models (Dao & Gu, 2024; Ali et al., 2024; Sieber et al., 2024) – all of which concern 1-layer models. Our results, presented in Figure 3, reveal two key insights:

1. First, for a fixed sequence length, recurrent models always benefit from scaling in width – as was happening in 2 layers (Sec. 2). That is, expanding the hidden state dimension enhances their performance. This result aligns well with current literature (Jelassi et al., 2024; Orvieto et al., 2024): as already mentioned, at each time step recurrent models store compressed inputs into a hidden state, which serves as a condensed representation of all past information. A larger hidden dimension facilitates less aggressive compression, allowing the model to retain more information.

2. Attention models exhibit a surprisingly different behavior: when constrained to a single layer, they fail to solve the task and increasing the hidden dimension does not affect their performance. This is in stark contrast to their strong results in 2-layer architectures, where even the smallest model was sufficient to solve the task in the hardest setting. Interestingly, in this setting transformers are capable on average of recalling one key-value pair in every setting, suggesting a memory size issue when only one layer is present.

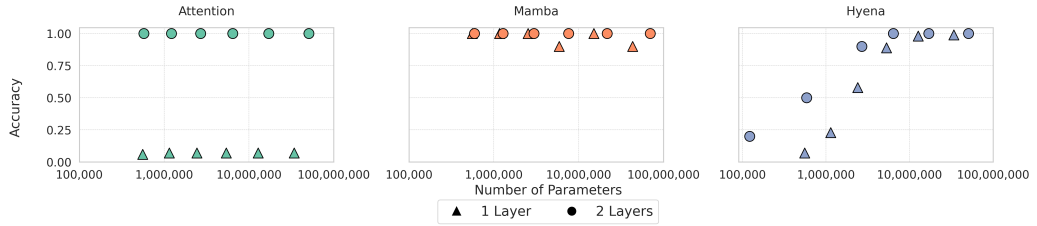

Figure 4: *Scaling models in width and depth (Seq len: 256, KV pairs: 64). Symbols with the same shape and color represent models of increasing size in the following order: 64, 128, 256, 512, 1024, and 2048. We show how rather than the number of parameters, is the way these models are scaled that impacts performance. Specifically, recurrent models benefits from scaling in width, while attention benefits from scaling in depth.*

Our findings highlight a key takeaway from our study: attention and recurrent models exhibit opposite scaling behaviors in width and depth. In other words, as shown in Fig. 4, rather than the number of parameters, it is the way these models are scaled that has most impact on their performance.

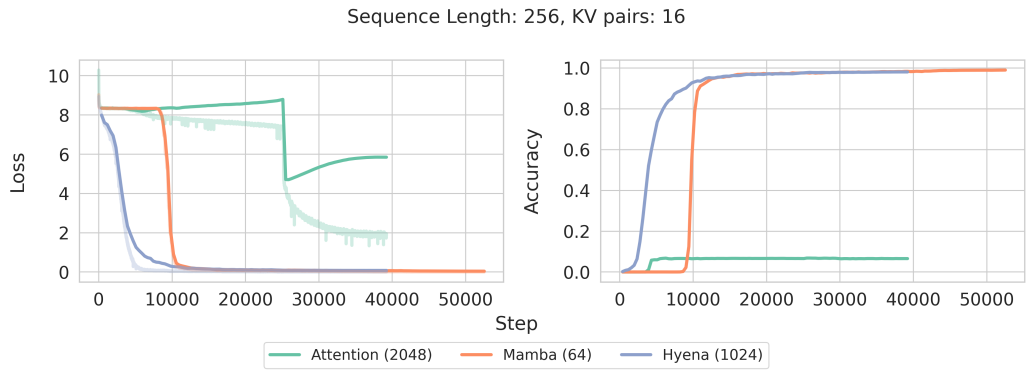

Figure 5: *Training (lower opacity) and Validation dynamics of 1-layer models. We reported within brackets the smallest width that solves the task, if possible; or otherwise the biggest width we tried (for attention). Differently from Mamba, Hyena requires the model dimension to exceed the sequence length. Both exhibit smooth learning dynamics, leading to perfect performance. Attention shows a loss bump, but without accuracy gains, suggesting a failed attempt to form induction heads.*

4 1-LAYER TRAINING DYNAMICS AND INDUCTION HEADS PHENOMENON

Sec. 3 sparked our curiosity, leading us to explore the single-layer architecture setup further – to understand why attention hits a performance ceiling while recurrent models can solve the task.

In this section, we analyze the training dynamics of well-tuned Hyena, attention and Mamba models. As illustrated in Fig. 5 we identify two main patterns. First, Hyena (and similarly other non-selective recurrent models like H3 and RWKV) exhibits consistently smooth learning dynamics, with a gradual and steady improvement that eventually lead to convergence at the solution. Specifically, loss reductions align closely with increases in accuracy. Differently, attention accuracy remains largely unchanged throughout training. A similar trend appears in the test loss, which remains relatively stable until a sudden bump occurs, after which the test loss settles again. This bump resembles the formation of an induction head circuit (Olsson et al., 2022), and to the best of our knowledge has previously only been observed during the training of multi-layer transformer architectures. However, as opposed to what can observe in 2-layer models, this phase transition in the loss does not correspond to an accuracy improvement for attention. Based on previous work (Olsson et al., 2022), we hypothesize that during this phase transition, the attention mechanism *attempts* to form induction heads. However, in the single-layer setting, the model lacks the expressivity needed to effectively leverage this mechanism for task resolution. Interestingly, the dynamics of Mamba is mixed:

1. Like single-layer attention models, we report a significant loss bump, reinforcing the connection between Mamba and attention mechanisms, as suggested in Ali et al. (2024); Dao & Gu (2024).
2. However, unlike transformers, Mamba can successfully solve the task even in a single-layer setting – provided the learning rate is properly tuned, similarly to other recurrent models.

Our results highlight a crucial distinction: while attention and recurrent models share some common ground, yet distinct inductive biases. Moreover, their performance is in strong interaction with the optimization algorithm at hand (in our case, Adam (Kingma, 2014)), as we also saw in Figure 2. Understanding these nuances is key to optimally leveraging both architectures, perhaps also towards hybrid models (Waleffe et al., 2024; Dao & Gu, 2024).

5 DISCUSSION AND CONCLUSIONS

In this work, we used MQAR as a benchmark to compare attention and recurrent models at a small scale. Our findings shed additional light on how the underlying mechanisms of these models influence their performance. Specifically, we showed that recurrent models are highly sensitive to optimization, with their performance significantly affected by the choice of learning rate. This

[2]By single layer in attention and recurrent models we mean a sequence mizer followed by an MLP.

underscores the need for further research to improve their stability. Additionally, we observed contrasting scaling behaviors: recurrent models benefit from the increased width and hidden state size, whereas transformers struggle with MQAR in a single-layer configuration. Interestingly, despite their poor performance, single-layer transformers exhibit training dynamics resembling the induction head phenomenon, previously reported only in multi-layer settings. Instead, Mamba displays similar behavior but successfully solves the task. Our findings suggest overlaps between the optimization landscapes of Mamba and attention, yet with crucial differences related to expressivity, to study further. Looking ahead, we think that exploring other synthetic reasoning tasks and other positional embeddings could provide further insights into the mechanisms behind these models.

REFERENCES

Ameen Ali, Itamar Zimerman, and Lior Wolf. The hidden attention of Mamba models. *arXiv preprint arXiv:2403.01590*, 2024.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Re. Zoology: Measuring and improving recall in efficient language models. In *International Conference on Learning Representations*, 2023.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Re. Simple linear attention language models balance the recall-throughput tradeoff. In *International Conference on Machine Learning*, 2024.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, 2024.

Daniel Y Fu, Tri Dao, Khaled Kamal Saab, Armin W Thomas, Atri Rudra, and Christopher Re. Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*, 2023.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines, 2014.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024.

Samy Jelassi, David Brandfonbrener, Sham M Kakade, et al. Repeat after me: Transformers are better than state space models at copying. In *International Conference on Machine Learning*, 2024.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. In *Advances in Neural Information Processing Systems*, 2024.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Antonio Orvieto, Soham De, Caglar Gulcehre, Razvan Pascanu, and Samuel L Smith. Universality of linear recurrences followed by non-linear projections: Finite-width guarantees and benefits of complex eigenvalues. In *International Conference on Machine Learning*, 2024.

Bo Peng, Eric Alcaide, Quentin Gregory Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Nguyen Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Kiran 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 S. Wind, Stanisław Woźniak, Zhenyuan Zhang,

Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics*, 2023.

Jerome Sieber, Carmen Amo Alonso, Alexandre Didier, Melanie Zeilinger, and Antonio Orvieto. Understanding the differences in foundation models: Attention, state space models, and recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. An empirical study of Mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024.

A APPENDIX

A.1 RELATED WORKS

Induction Heads While investigating the capabilities of transformers in few-shot learning, previous work ((Olsson et al., 2022)) showed the phenomenon of induction heads. The main insight was that during training, with transformers with at least 2 layers, a special kind of attention heads called "**Induction heads**" is formed, causing a noticeable drop in the loss perplexity, while giving a sudden boost in In-context learning performances.

More formally, induction heads are implemented by a circuit consisting of a pair of attention heads in different layers that work together to copy or complete patterns. The first attention head copies information from the previous token into each other tokens. In this way, the second attention head can attend to tokens based on what happened before them, rather than their own content. Specifically, the second head (the proper "induction head") searches for a previous place in the sequence where the present token **A** occurred and attends to the next token (call it **B**), copying it and causing the model to be more likely to output **B** as the next token. That is, the two heads working together cause the sequence ...[**A**][**B**]...[**A**] to be more likely completed with [**B**].

Induction heads are named by analogy to inductive reasoning, where we might infer that if **A** is followed by **B** earlier in the context, **A** is more likely to be followed by **B** again later in the same context. Induction heads are capable of crystallizing that inference. They search the context for previous instances of the present token, attend to the token which would come next in the pattern repeated, and increase its probability in terms of logit. Induction heads attend to tokens that would be predicted by basic induction (over the context, rather than over the training data).

A.2 EXPERIMENTAL DETAILS

In this section, we describe the experimental setup used throughout our study. Clearly outlining these details is crucial for interpreting the results presented in subsequent sections. Our implementation is inspired by methodologies from Zoology (Arora et al., 2023).

Data. The dataset consists of sequences of tokens representing key-value pairs. Tokens are sampled from a fixed vocabulary of $8,192$ tokens. Within each sequence, key-value tokens are assigned randomly, ensuring that the model cannot learn a static mapping. Consequently, each sample is independent, requiring the model to infer the role of tokens in context rather than relying on memorization. The synthetic dataset is structured with four specific sequence lengths, each paired with a corresponding number of key-value pairs to recall:

- 64 tokens with 4 key-value pairs;
- 128 tokens with 8 key-value pairs;
- 256 tokens with 16 key-value pairs;
- 512 tokens with 64 key-value pairs.

For the first three sequence lengths, the ratio of key-value pairs to sequence length is $1:16$, whereas for the longest sequence, the ratio is $1:8$, making it the most challenging case. For each sequence length, a dedicated dataset is created, consisting of $100,000$ training samples and $3,000$ test samples. Model evaluation is performed by training each model on a specific sequence length and subsequently assessing its performance on that same length.

Models Our experiments utilize a total of six models:

- Two attention-based models: Attention and Based.
- Four recurrent models: H3, Hyena, RWKV and Mamba.

Each model is tested across six model dimensions: 64, 128, 256, 512, 1024 and 2048. Additionally, models are implemented in two configurations: 1-layer and 2-layer. Notably, a "layer" in our context refers to the concatenation of two blocks: a sequence mixer (e.g., attention, RWKV, etc.) followed

by an MLP. Thus, a 1-layer model consists of two blocks, aligning with the terminology used in prior work (Arora et al., 2023; Olsson et al., 2022). Positional information is used only in attention and Based.

Training and Evaluation We used GPU A100 in all our experiments. We trained for 100 epochs using AdamW as optimizer, weight decay 0.1, warmup duration 10%, linear warmup. The batch size varied depending on the sequence length: 128 for sequence length 512, 256 for sequence length 256 and 512 otherwise. Each configuration (combining model type, model dimension, and sequence length) undergoes a learning rate sweep to identify the optimal learning rate. The reported accuracy for each configuration corresponds to the best performance achieved across the tested learning rates. We want to highlight that the accuracy reported should be interpreted as the average percentage of key-value pairs correctly labeled. Specifically, achieving 50% accuracy with sequence length 64 and 4 as relative number of key-value pairs means that on average the model recalls correctly 2 values given 4 keys. To ensure robustness, all experiments are conducted using three random seeds (42, 123, and 777), with results reported as the mean and standard deviation across these trials.

A.3 Full Tables

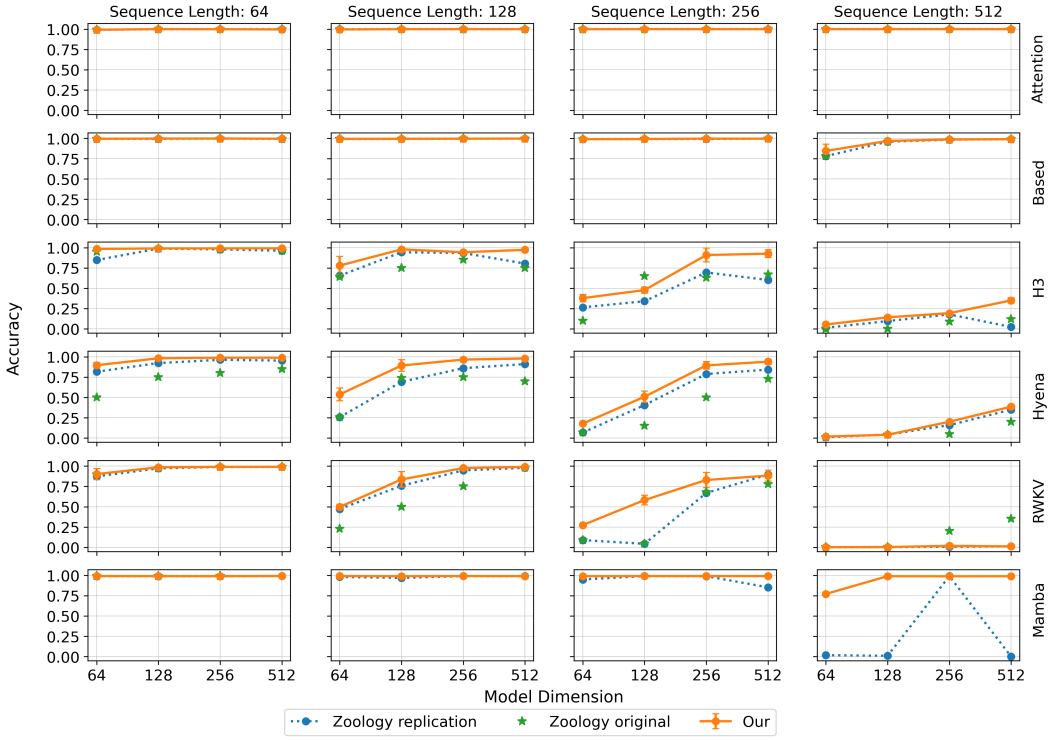

Figure 6: *Performance of 2-layers models. Results for H3 (Fu et al., 2023), RWKV (Peng et al., 2023) and Based (Arora et al., 2024) included in App. A.3. We report the official results*[3] *(green stars) and the replication running the original code of (Arora et al., 2023) (dotted blu line). While for replication, we used the learning rates grid by Arora et al. (2023), we note here that, due to high sensitivity to the learning rate (Fig, 2), tuning drastically affects performance. In solid orange, we provide results with a finer grid (cf. Fig.2). Careful tuning of the learning rate gives a general improvement in the performance of recurrent models. This becomes especially crucial in Mamba, where the task becomes solvable at high sequence lengths >> hidden size. The results show the mean and relative max-min errors for 3 seeds. Attention always solves the task (all curves overlap).*

[3]Mamba was not included in the official work but some experiments, with different settings compared to ours, are documented in the blog post.