CONTEXTUAL SPARSITY AS A TOOL FOR MECHANISTIC UNDERSTANDING OF RETRIEVAL IN HYBRID FOUNDATION MODELS

Davide Zani*, Felix Michalak*
Bernoulli Institute
University of Groningen
{d.zani,k.f.michalak}@student.rug.nl

Steven Abreu
CogniGron Center & Bernoulli Institute
University of Groningen
s.abreu@rug.nl

ABSTRACT

We mechanistically investigate the role of self-attention in hybrid foundation models that combine state-space modules with self-attention. Evaluating the RecurrentGemma-2B model on a synthetic needle-in-a-haystack task, we show that completely deactivating attention heads causes a total retrieval failure—even though overall generation quality is only modestly affected. Using a contextual sparsity approach inspired by (Liu et al., 2023), we find that retaining only 2 out of 10 attention heads is sufficient to nearly preserve full retrieval performance. These findings highlight a specialized function of self-attention for copying and retrieval, suggesting that future work could focus on designing dedicated, interpretable retrieval mechanisms within hybrid architectures.

Introduction Recent advances in large language models have increasingly focused on linear-attention models, especially state-space models (SSMs) (De et al., 2024; Gu & Dao, 2023; Dao & Gu, 2024; Qin et al., 2024). SSMs scale sub-quadratically with sequence length, which is an improvement over transformer models, as their attention mechanism scales quadratically with sequence length. However, SSMs show distinct weaknesses that make them fall behind in large model sizes. Activations in SSMs compress all previously seen tokens into a vector of fixed size which naturally leads to their recall ability declining with sequence length (Jelassi et al., 2024; Arora et al., 2024a). They also show a *fuzzy memory*, which ”forgets” context information depending on the distance to the end of the prompt (Waleffe et al., 2024). As a solution to this, SSMs are often combined with self-attention layers into hybrid models such as RecurrentGemma (Botev et al., 2024; De et al., 2024), Jamba (Lieber et al., 2024) and others (Dong et al., 2024) Hybrid SSMs close the gap to transformer capabilities while remaining more efficient during training and inference at scale (Dong et al., 2024). Despite interpretability efforts in both SSMs and self-attention models, the distinct roles and interactions between these components in hybrid SSMs remain underexplored.

Interpretability on hybrid LLMs There have been succesful attempts to use self-attention interpretability insights (Ali et al., 2024; Zimerman et al., 2024), as well as to manipulate attention in SSMs to achieve better performance (Ben-Kish et al., 2024). Previous work has shown that the key weakness of purely recurrent LLMs lies in recall (Arora et al., 2024a) and copying (Jelassi et al., 2024). In this paper, we analyze the performance of a hybrid model on the needle-in-a-haystack (NIAH) task (Bai et al., 2024) and work towards isolating the role of the self-attention layers in RecurrentGemma through the lens of sparsity–by pruning attention heads.

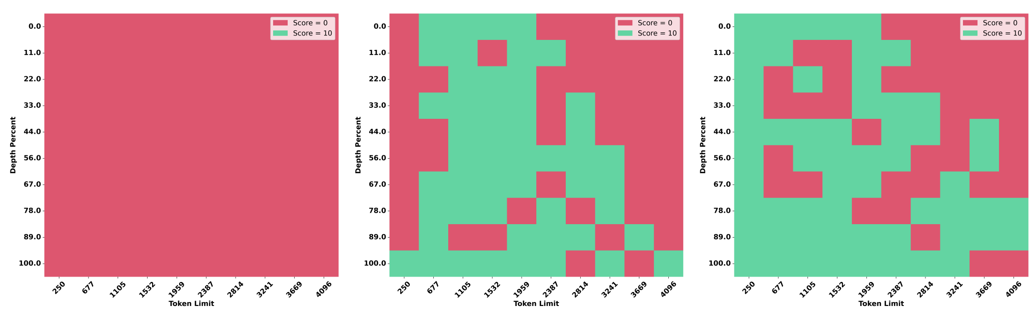
Sparsity for interpretability Contextual sparsity through pruning attention heads has been introduced by Liu et al. (2023) who demonstrated that this can lead to significantly reduced latency for large language models. Sparsity is also a fundamental tool in mechanistic interpretability (Kissane et al., 2024; Lieberum et al., 2024; Huben et al., 2023) and has been applied to explainability of LLMs in various ways (Treviso & Martins, 2020; Pruthi et al., 2022).

Contributions We first show that pruning (see B) all attention heads leads to failure on the NIAH task, although it does not lead to large degradation in text generation, when evaluated qualitatively.

We proceed to show that not all attention heads are necessary to attain maximum performance on the NIAH task. With a simple contextual pruning method that keeps only the top-k attention heads with maximum entropy, we show that performance for $k > 2$ heads out of $H = 10$ heads does not improve much over the performance of $k = 2$ heads.

Compute and memory analysis We found that the recurrent layers contain 19% of all parameters, the MLPs contain 75% of all parameters, and the attention heads only contain around 6% of the total number of parameters (see A.1). This leads us to hypothesize that the attention heads contribute a specialized function to the overall language modeling ability of RecurrentGemma. This is further confirmed by analyzing the FLOPs per layer type for different sequence lengths. FLOPs grow quasi-linearly with sequence length, and each recurrent block uses $> 3\times$ more compute than an attention block (see A.2). There are also $2\times$ more recurrent blocks than attention blocks in RecurrentGemma (18 recurrent blocks, 8 attention blocks in the standard configuration). Based on these findings, we aim to investigate if copying or retrieval are the main task of the attention layers, as argued by Arora et al. (2024b).

Retrieval results The NIAH task was run for k-values $0-10$, thereby ranging from complete deactivation of the self-attention layers to a non-sparsified model. Most noticeable was the performance of the k =2 configuration, which shows a similar performance to the non-sparsified configuration with k =10. Any k below led to a drastic decrease in accuracy in the NIAH task, and k-values >2 showed no comparable performance increase.



a) k = 0 b) k = 2 c) k = 10

Figure 1: Heatmap for the NIAH task for selected k-values. The performance decreases drastically at k =0, but the performance increase from k =2 to k =10 is only marginal.

Discussion Our experiments indicate that the retrieval ability on the NIAH task critically crucially on the self-attention layers. Pruning all attention heads leads to a catastrophic failure in retrieval, while retaining just the top-k heads (with $k = 2$ out of $H = 10$) nearly preserves full performance. This suggests that only a small subset of attention heads functions as dedicated retrieval mechanisms–as also discovered by Olsson et al. (2022). However, as noted by Waleffe et al. (2024), SSMs might possess the underlying knowledge but require additional training to interpret retrieval instructions correctly. Thus, the observed degradation could stem partly from the model's impaired ability to understand the retrieval instruction when attention is removed. Further research will aim to disentangle these factors, exploring whether fine-tuning, architectural adjustments, or prompt formatting can recover retrieval performance in the absence of full attention.

Conclusion We have demonstrated that the retrieval performance of the RecurrentGemma hybrid model is crucially reliant on its self-attention components. Through systematic pruning experiments on a synthetic NIAH task, we found that while complete removal of attention heads leads to total retrieval failure, retaining a minimal subset (e.g., $k = 2$ heads) maintains near-optimal performance. These results underscore the specialized role of self-attention in tasks requiring copying and retrieval. This work lays the groundwork for developing more efficient and interpretable hybrid models. Future research should explore whether dedicated retrieval modules can be integrated into SSM-based architectures, potentially mitigating the high computational cost of attention while maintaining or even enhancing retrieval capabilities.

ACKNOWLEDGEMENTS

We would like to thank Apart Research for funding and supporting the team, without which this work would not have been possible. We thank Natalia Pérez-Campanero Antolín for insightful feedback at various stages of our project.

CODE

Our code can be found on github.com/stevenabreu7/hybrid-interpretability. This GitHub repository includes the RecurrentGemma2B code modified for sparsification, as well as Python Notebooks for the NIAH benchmark and preliminary Analysis.

REFERENCES

Ameen Ali, Itamar Zimerman, and Lior Wolf. The Hidden Attention of Mamba Models, March 2024. URL http://arxiv.org/abs/2403.01590. arXiv:2403.01590 [cs].

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff, February 2024a. URL http://arxiv.org/abs/2402.18668. arXiv:2402.18668 [cs].

Simran Arora, Aman Timalsina, Aaryan Singhal, Benjamin Spector, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré. Just read twice: closing the recall gap for recurrent language models, July 2024b. URL http://arxiv.org/abs/2407.05483. arXiv:2407.05483 [cs].

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. LongAlign: A Recipe for Long Context Alignment of Large Language Models, January 2024. URL http://arxiv.org/abs/2401.18058. arXiv:2401.18058 [cs].

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. DeciMamba: Exploring the Length Extrapolation Potential of Mamba, June 2024. URL http://arxiv.org/abs/2406.14528. arXiv:2406.14528 [cs].

Aleksandar Botev, Soham De, Samuel L. Smith, Anushan Fernando, George-Cristian Muraru, Ruba Haroun, Leonard Berrada, Razvan Pascanu, Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Sertan Girgin, Olivier Bachem, Alek Andreev, Kathleen Kenealy, Thomas Mesnard, Cassidy Hardin, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Armand Joulin, Noah Fiedel, Evan Senter, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, David Budden, Arnaud Doucet, Sharad Vikram, Adam Paszke, Trevor Gale, Sebastian Borgeaud, Charlie Chen, Andy Brock, Antonia Paterson, Jenny Brennan, Meg Risdal, Raj Gundluru, Nesh Devanathan, Paul Mooney, Nilay Chauhan, Phil Culliton, Luiz GUStavo Martins, Elisa Bandy, David Huntsperger, Glenn Cameron, Arthur Zucker, Tris Warkentin, Ludovic Peran, Minh Giang, Zoubin Ghahramani, Clément Farabet, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, Yee Whye Teh, and Nando de Frietas. RecurrentGemma: Moving Past Transformers for Efficient Open Language Models, April 2024. URL http://arxiv.org/abs/2404.07839. arXiv:2404.07839 [cs].

Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality, May 2024. URL http://arxiv.org/abs/2405.21060. arXiv:2405.21060 [cs].

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar Gulcehre. Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models, February 2024. URL http://arxiv.org/abs/2402.19427. arXiv:2402.19427 [cs].

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. Hymba: A Hybrid-head Architecture for Small Language Models, November 2024. URL http://arxiv.org/abs/2411.13676. arXiv:2411.13676.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, October 2023. URL https://openreview.net/forum?id=AL1fq05o7H.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. October 2023. URL https://openreview.net/forum?id=F76bwRSLeK.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat After Me: Transformers are Better than State Space Models at Copying, June 2024. URL http://arxiv.org/abs/2402.01032. arXiv:2402.01032 [cs].

Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. Interpreting Attention Layer Outputs with Sparse Autoencoders, June 2024. URL http://arxiv.org/abs/2406.17759. arXiv:2406.17759 [cs].

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A Hybrid Transformer-Mamba Language Model, March 2024. URL http://arxiv.org/abs/2403.19887. arXiv:2403.19887 [cs].

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2, August 2024. URL http://arxiv.org/abs/2408.05147. arXiv:2408.05147 [cs].

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. Deja Vu: Contextual Sparsity for Efficient LLMs at Inference Time. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 22137–22176. PMLR, July 2023. URL https://proceedings.mlr.press/v202/liu23am.html. ISSN: 2640-3498.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context Learning and Induction Heads, September 2022. URL http://arxiv.org/abs/2209.11895. arXiv:2209.11895 [cs].

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. Evaluating Explanations: How Much Do Explanations from the Teacher Aid Students? *Transactions of the Association for Computational Linguistics*, 10:359–375, 2022. doi: 10.1162/tacl_a_00465. URL https://aclanthology.org/2022.tacl-1.21/. Place: Cambridge, MA Publisher: MIT Press.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. HGRN2: Gated Linear RNNs with State Expansion, April 2024. URL http://arxiv.org/abs/2404.07904. arXiv:2404.07904 [cs].

Marcos Treviso and André F. T. Martins. The Explanation Game: Towards Prediction Explainability through Sparse Communication. In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad (eds.), *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 107–118, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.10. URL https://aclanthology.org/2020.blackboxnlp-1.10/.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An Empirical Study of Mamba-based Language Models, June 2024. URL `http://arxiv.org/abs/2406.07887`. arXiv:2406.07887 [cs].

Itamar Zimerman, Ameen Ali, and Lior Wolf. Explaining Modern Gated-Linear RNNs via a Unified Implicit Attention Formulation, October 2024. URL `http://arxiv.org/abs/2405.16504`. arXiv:2405.16504 [cs].

A Preliminary Analysis

A.1 Parameter Analysis

Table 1: Parameter count and proportions of total parameters per layer type in the standard 2B configuration

Layer type	Parameter count	Proportion of total (%)
RecurrentGemmaMlp	1534008320	74.8
RecurrentGemmaRecurrentBlock	377994240	18.4
RecurrentGemmaRglru	23731200	1.2
RecurrentGemmaSdpaAttention	115363840	5.6
RecurrentGemmaRMSNorm	135680	< 0.1

A.2 Flop Analysis

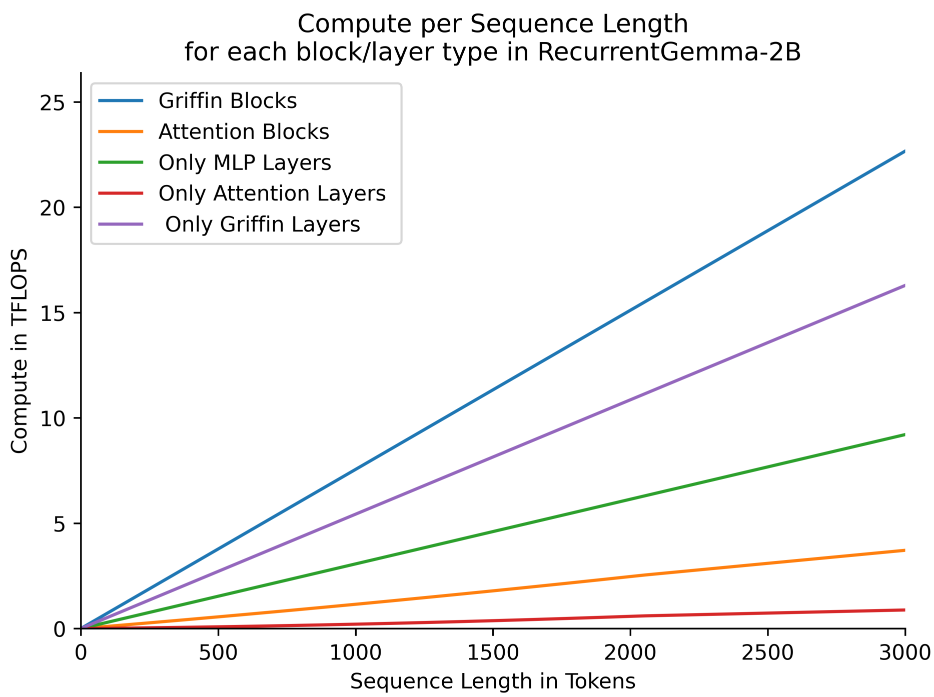


Figure 2: FLOP analysis for RecurrentGemma-2B, sorted by different layer types. These graphs show TFLOPS per sequence length in tokens during inference. Note the exponential growth of the attention layers (red) until 2048 tokens, which continues linearly afterward. This shows the implementation of sliding window attention.

B Pruning implementation

Our chosen pruning strategy was not focused on efficiency improvements to any degree. This paper is supposed to showcase the usefulness of sparsity as an interpretability tool, in which case efficiency can be disregarded to some degree. We chose a run-time implementation.

The pruning implementation first completes a full forward pass to calculate all the attention weights and values. After that, the top-k attention heads are identified, based on their weights. Our top-k

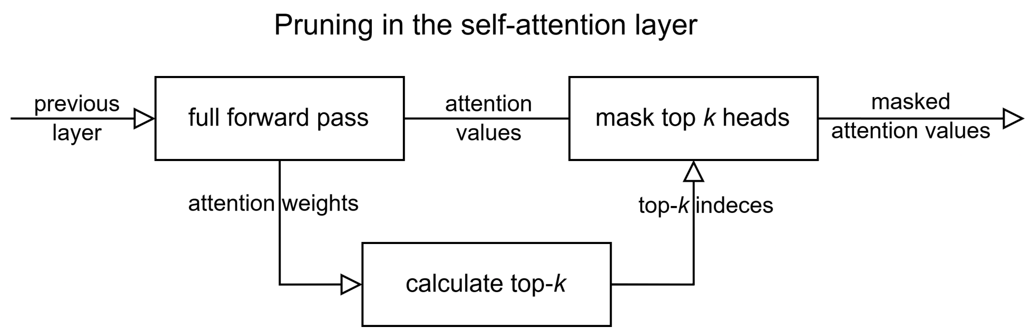


Figure 3: This diagram shows the workflow for pruning the attention heads.

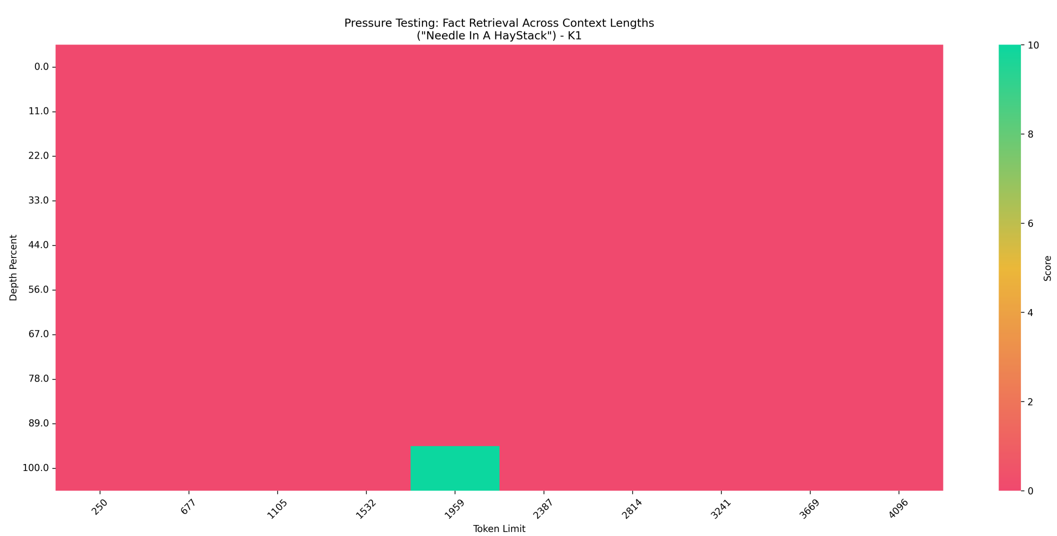
implementation calculates the entropy with

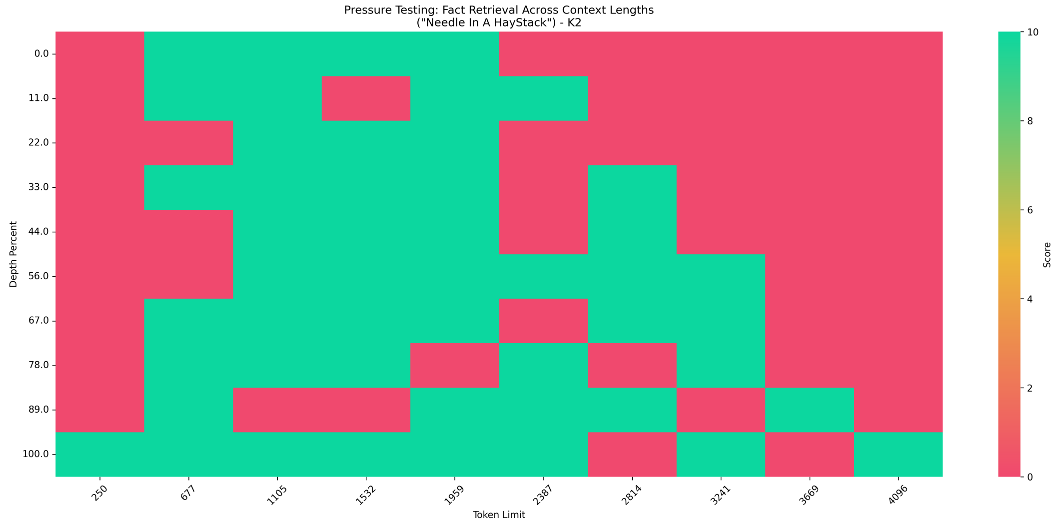
$$H(h) = \sum x * \log_2(x),$$

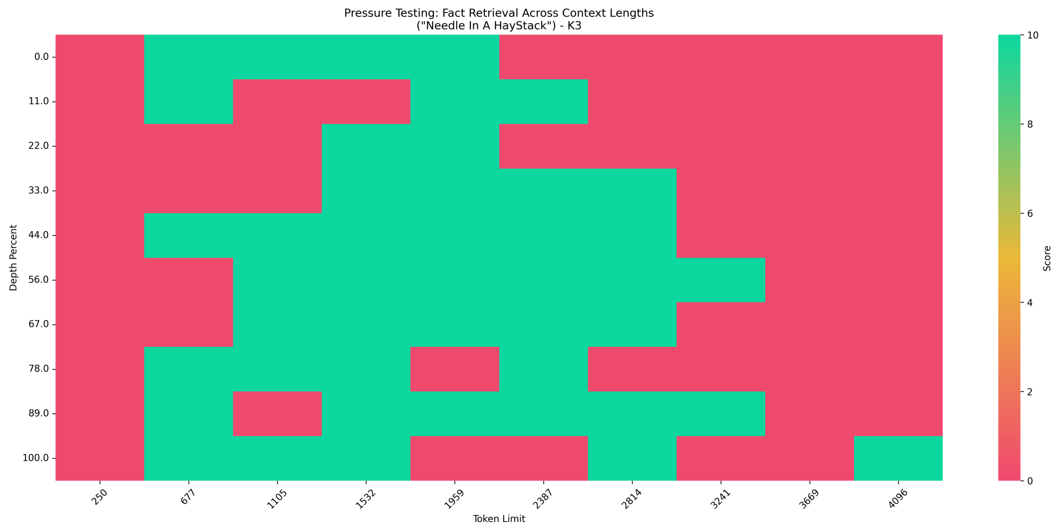
where h is an attention head and x represents all attention weights of h. Note that this equation is missing the negative sign $(-)$ in front. Entropy measures the uncertainty in a distribution, and so ascribes a uniform distribution the highest value, and a deterministic distribution the lowest value. However, we want to use entropy as an inverse metric for uncertainty, and thereby simply dismiss the negative sign. The resulting pruning mechanism keeps the top k most peaked attention weight distributions, as attention is only useful if it points to something specific, not to everything at the same time.

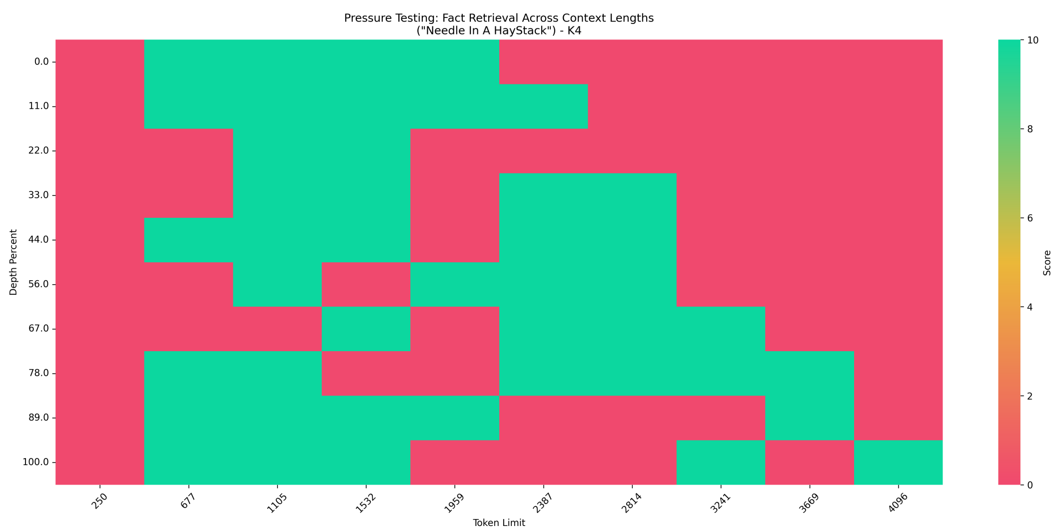
The top k attention heads are kept, the rest is masked out by nullifying all attention weights.

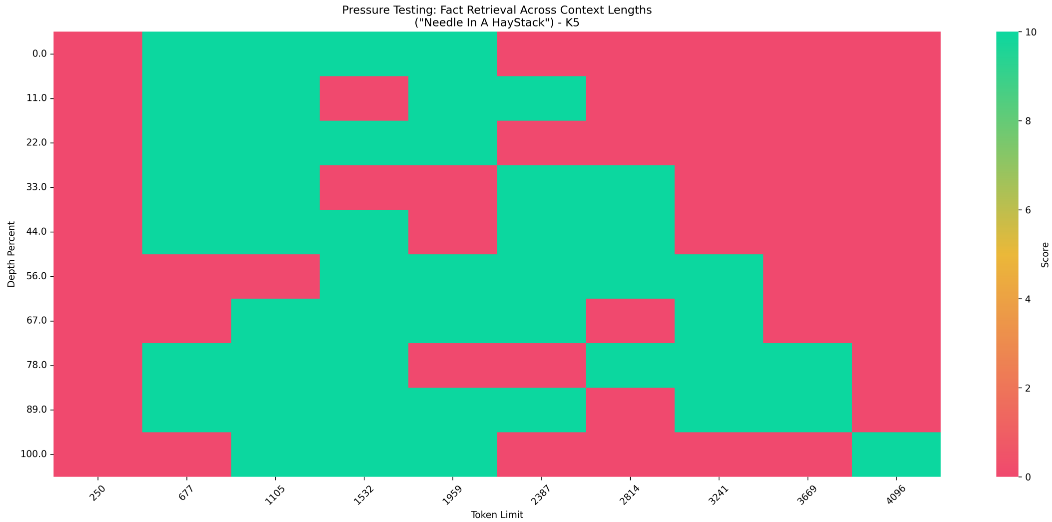
C NIAH RESULTS

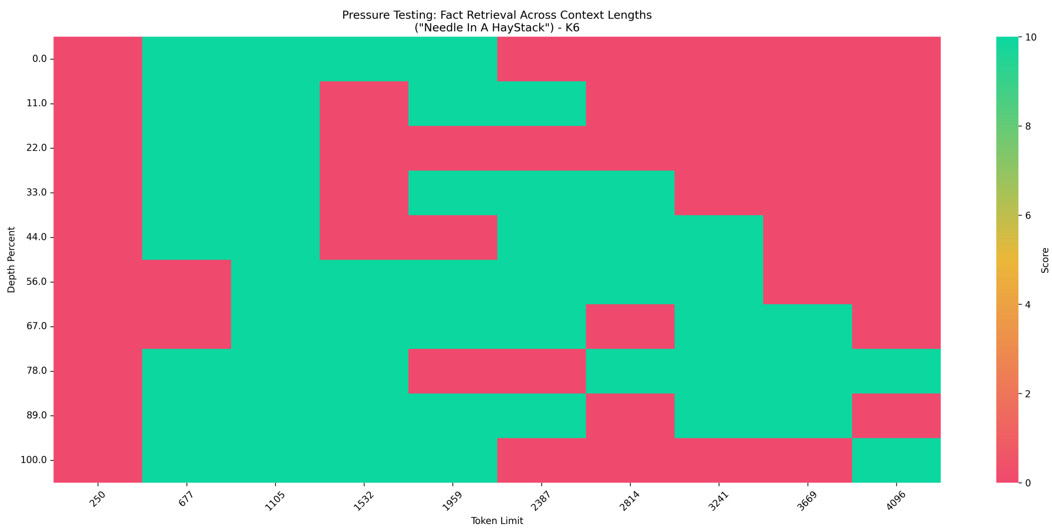


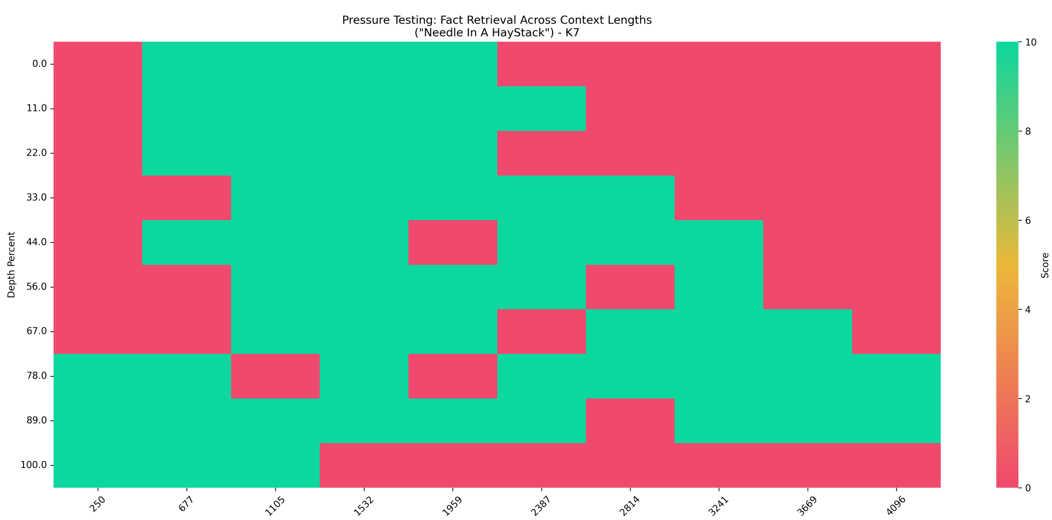

Pressure Testing: Fact Retrieval Across Context Lengths
("Needle In A HayStack") - K2
Depth Percent
0.0
11.0
22.0
33.0
44.0
56.0
67.0
78.0
89.0
100.0
Token Limit
250
677
1105
1532
1959
2387
2814
3241
3669
4096
Score
0
2
4
6
8
10


Pressure Testing: Fact Retrieval Across Context Lengths
("Needle In A HayStack") - K3
Depth Percent
0.0
11.0
22.0
33.0
44.0
56.0
67.0
78.0
89.0
100.0
Token Limit
250
677
1105
1532
1959
2387
2814
3241
3669
4096
Score
0
2
4
6
8
10


Pressure Testing: Fact Retrieval Across Context Lengths
("Needle In A HayStack") - K4
Depth Percent
0.0
11.0
22.0
33.0
44.0
56.0
67.0
78.0
89.0
100.0
Token Limit
250
677
1105
1532
1959
2387
2814
3241
3669
4096
Score
0
2
4
6
8
10


Pressure Testing: Fact Retrieval Across Context Lengths
("Needle In A HayStack") - K5
Depth Percent
0.0
11.0
22.0
33.0
44.0
56.0
67.0
78.0
89.0
100.0
Token Limit
250
677
1105
1532
1959
2387
2814
3241
3669
4096
Score
0
2
4
6
8
10


Pressure Testing: Fact Retrieval Across Context Lengths
("Needle In A HayStack") - K6
Depth Percent
0.0
11.0
22.0
33.0
44.0
56.0
67.0
78.0
89.0
100.0
Token Limit
250
677
1105
1532
1959
2387
2814
3241
3669
4096
Score
0
2
4
6
8
10


Pressure Testing: Fact Retrieval Across Context Lengths
("Needle In A HayStack") - K7
Depth Percent
0.0
11.0
22.0
33.0
44.0
56.0
67.0
78.0
89.0
100.0
Token Limit
250
677
1105
1532
1959
2387
2814
3241
3669
4096
Score
0
2
4
6
8
10

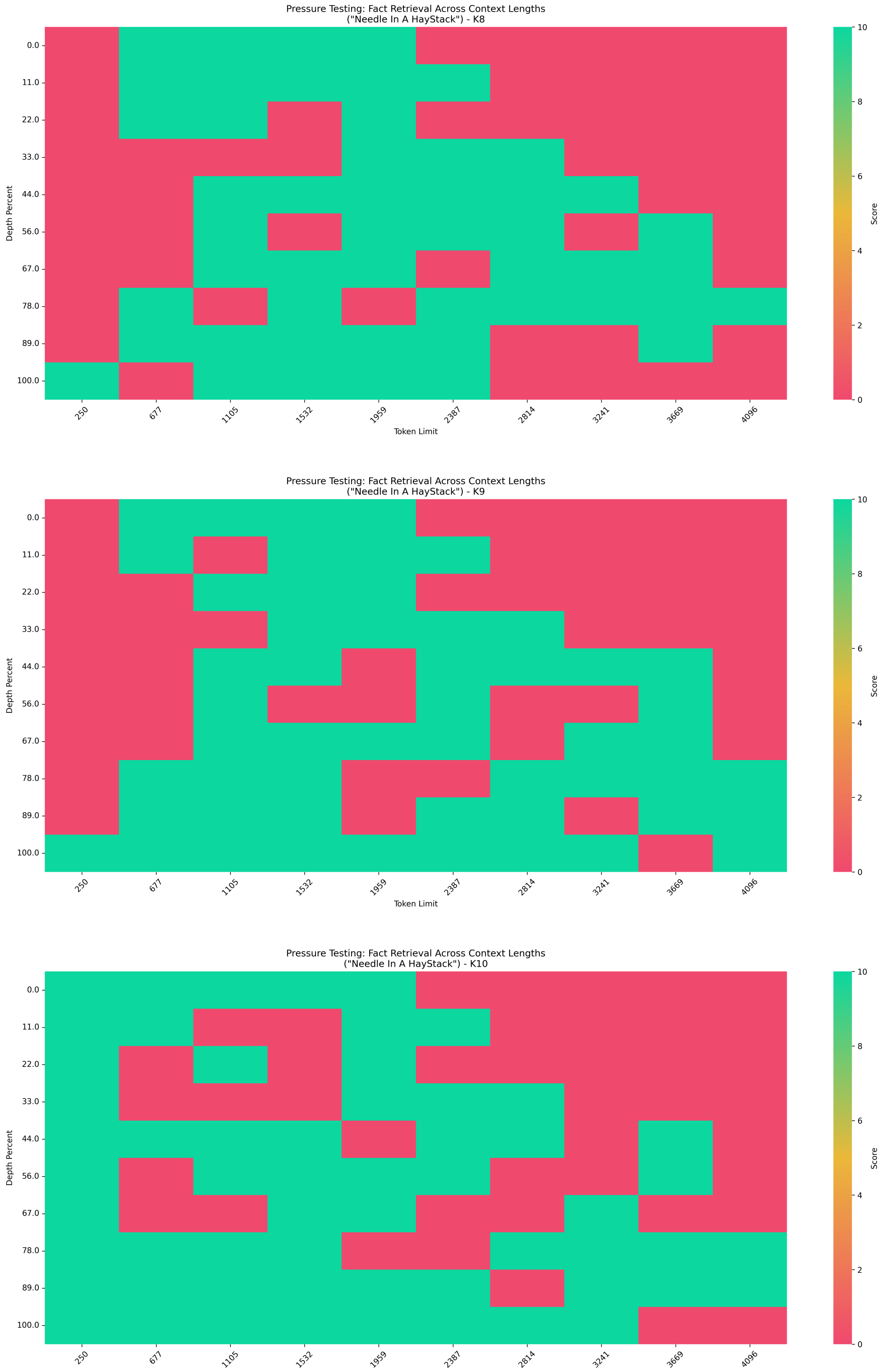


Figure 4: This is a collection of heatmaps on the needle-in-a-haystack-task for k-values $(1, 10)$