Mitigating Over-Smoothing in Mamba2 via Spectral Domain Analysis

Seojin Kim [*1] **Yehjin Shin** [*1] **Noseong Park** [1]

Abstract

Mamba2, a rising contender against transformer-based architectures, has garnered significant attention for its impressive performance across diverse tasks, sparking a wave of research into its analysis and improvement. In this paper, we investigate Mamba2 through the lens of spectral analysis, uncovering a critical structural bias: Mamba2 inherently functions as a low-pass filter, leading to over-smoothing. Over-smoothing, where token representations become overly uniform, hampers the model's ability to capture rich and diverse features, ultimately contributing to performance degradation. To address this, we propose a straightforward yet effective high-frequency enhancement method. By selectively amplifying high-frequency components at the layer level, our approach mitigates the over-smoothing effect, restoring token diversity and improving representational richness. Experiments confirm the efficacy of our method, demonstrating its ability to enhance Mamba2's performance across key tasks.

1. Introduction

Transformer-based architectures have emerged as the dominant approach for sequence modeling, ranging from text generation to machine translation. However, the inherent limitations of transformers(e.g. quadratic time complexity) have motivated the development of sub-quadratic alternatives designed to address these challenges (Yang et al., 2024b; Gu et al., 2022b; Smith et al., 2023; Poli et al., 2023a; Peng et al., 2023; Sun et al., 2024). Among these, Mamba and Mamba2 has gained considerable attention due to its strong performance across a wide array of benchmarks (Gu & Dao, 2023; Dao & Gu, 2024b). Nevertheless, a comprehensive analysis of its architectural characteristics and potential limitations is essential for a complete understanding of its capabilities.

[*]Equal contribution [1]KAIST, South Korea. Correspondence to: Noseong Park <noseong@kaist.ac.kr>.

The 3^{rd} Workshop on Efficient Systems for Foundation Models at the 41^{st} International Conference on Machine Learning, Vancouver, Canada.

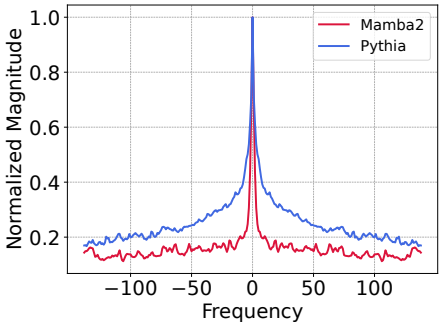


Figure 1. Spectral response of the attention map for Pythia and the M matrix (cf. Eq. 6) for Mamba2

This paper investigates a critical yet overlooked issue in Mamba2: over-smoothing, where token representations within a layer become nearly identical. Although over-smoothing is well-studied in transformers across various tasks (Wang et al., 2022; Shi et al., 2022; Guo et al., 2023), its occurrence and effects in Mamba2 remain largely unexplored. To bridge this gap, we conduct a spectral domain analysis of Mamba2 (Dao & Gu, 2024b), utilizing the M matrix (cf. Eq.6), a structure ideal for spectral examination. Our analysis reveals, for the first time, that Mamba2 has a strong bias toward low-frequency components, as illustrated in Figure 1. This structural limitation not only reduces expressiveness but also diminishes performance in language modeling tasks. To counteract this issue, we propose a high-frequency enhancement strategy, a simple yet effective modification that significantly alleviates over-smoothing. Our method leads to notable performance gains across multiple language modeling benchmarks, validating our hypothesis and demonstrating that targeted structural adjustments can improve Mamba2's expressiveness.

Our key contributions are as follows:

- We identify and analyze over-smoothing in Mamba2 through a novel Fourier spectrum perspective.
- We establish a direct link between Mamba2's spectral bias and its performance degradation.
- We propose and validate a simple high-frequency enhancement strategy that improves language modeling performance.

2. Related Works

2.1. Mamba2

Using definitions from Dao & Gu (2024b), we describe Mamba2's internal dynamics in this section. Each vector is designated as a row vector. Assuming that $U = [u_1, u_2, ..., u_T]^\top \in \mathbb{R}^{T\times d}$, that is, $u_i \in \mathbb{R}^d$, is a discrete time sequence of T tokens, the inner equation for the t-th token of each head of the Mamba2 layer can be understood as follows:

$$\begin{aligned} h_t &= A_t h_{t-1} + B_t x_t^\top \in \mathbb{R}^{N\times P}, \\ y_t &= C_t^\top h_t + D \odot x_t \in \mathbb{R}^P \end{aligned} \tag{1}$$

$$o_t = W_o(\mathrm{Norm}(y_t \odot W_z u_t)) \in \mathbb{R}^d \tag{2}$$

where t is current time token, $x_t, y_t \in \mathbb{R}^P$ are projected input representation and output hidden representations of t-th token respectively, Norm denotes RMS normalization (Zhang & Sennrich, 2019), $\odot$ denotes element-wise multiplication, $D \in \mathbb{R}^P$, $W_z \in \mathbb{R}^{P\times d}$, $W_o \in \mathbb{R}^{d\times P}$ are trainable parameters. Especially, in Mamba2, A_t is scalar-identity matrix, i.e. $A_t = a_t I$. We denote d for hidden representation dimension, N for state size, P for dimension of each head, T for sequence length. Detailed parameterization of $A_t, B_t, C_t, x_t, \Delta_t$ are deferred to Appendix A.2.

2.2. Transformer and Mamba2

Various models can be represented as sequence transformations. A sequence transformation is a parameterized mapping of sequences, defined as $Y = f_\theta(X)$, where $X \in \mathbb{R}^{T\times P}$ is mapped to $Y \in \mathbb{R}^{T\times P}$. Here, θ represents the model parameters. If this mapping can be expressed as $Y = M_\theta X$, it is referred to as a matrix transformation. For simplicity, we drop θ when it is clear from context. By adopting the matrix transformation form, we can describe various sequence modeling mechanisms within a unified framework:

$$\begin{aligned} &Q = \text{input},\ K = \text{input} \in \mathbb{R}^{T\times N}, \\ &V = \text{input} \in \mathbb{R}^{T\times P}, \quad M = (L \odot QK^\top) \in \mathbb{R}^{T\times T}, \\ &Y = MV \in \mathbb{R}^{T\times P} \end{aligned} \tag{3}$$

Let L be a $T \times T$ mask for autoregressive self-attention, typically a lower triangular matrix of 1s representing a causal mask:

$$M = f(L_c \odot QK^\top), \quad L_{c,ij} = \begin{cases} 1 & i \geq j \\ -\infty & i < j \end{cases}, \tag{4}$$

where $f = \mathrm{softmax}$, $Q = W_Q \cdot X$, $K = W_K \cdot X$, and $V = W_V \cdot X$, representing basic query, key, value mapping for the softmax attention.

In this paper, we define S6 as applying Eq. 1 for each tokens. This can be formulated in this matrix transformation form:

$$\begin{aligned} y_t &= \sum_{s=0}^{t} C_t^\top A_{t:s}^\times B_s x_s, \\ Y &= \mathrm{S6}(X) = MX, \\ M_{ji} &= C_j^\top A_j \cdots A_{i+1} B_i \end{aligned} \tag{5}$$

Considering $Q = C$, $K = B$, we can define matrix transformation as below since $A_t = a_t I$:

$$M = L_d \odot CB^\top,\ L_{d,ij} = \begin{cases} a_i \times \cdots \times a_{j+1} & i \geq j \\ 0 & i < j \end{cases} \tag{6}$$

where $a_i \in [0, 1]$ following its formulation. In this regard, we can understand Mamba2 and transformers in matrix transformation form, with main difference of how their mask L is formulated. For detailed equations on this sequence transformation form, refer to Appendix A.3.

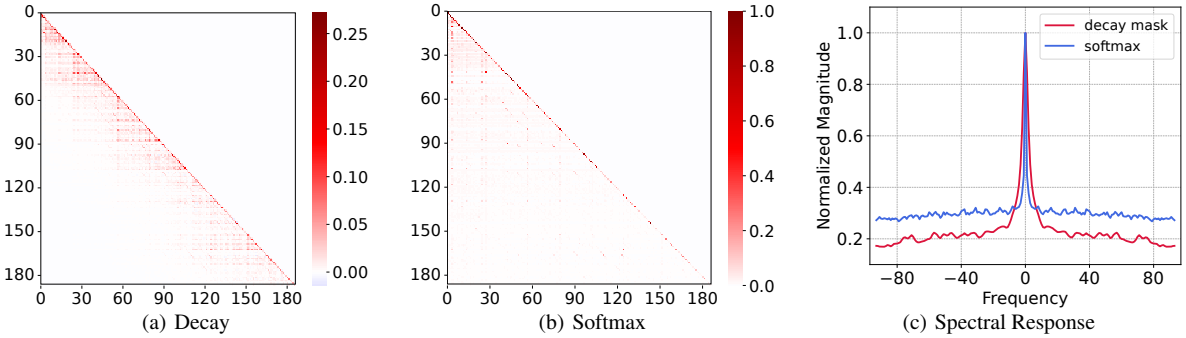
2.3. Over-smoothing Problem

Over-smoothing, a phenomenon predominantly studied in GNNs, arises when repeated message passing across layers leads to overly similar or low-rank node representations, resulting in indistinguishable embeddings and degraded performance (Li et al., 2018; Choi et al., 2023a). This issue has since been observed in Transformer models like BERT and ViTs. Shi et al. (2022) show that self-attention matrices in BERT resemble adjacency matrices, with layer normalization accelerating convergence to low-rank subspaces. To address this, they propose hierarchical fusion strategies. Similarly, Wang et al. (2022) identify self-attention in ViTs as a low-pass filter causing feature collapse and introduce spectral reweighting techniques. Guo et al. (2023) connect over-smoothing to dimensional collapse and propose ContraNorm, a normalization layer inspired by contrastive learning, to alleviate this issue in both GNNs and Transformers. Building on this, a growing number of methods leveraging graph-based algorithms to mitigate over-smoothing are under active investigation (Choi et al., 2023b; Wi et al., 2025).

3. Frequency Bias in Mamba

3.1. Bias of the Model architecture

To better understand the structural behavior of Mamba2, we examine the mathematical formulation of its M matrix, which functions similarly to the attention weights in transformers by determining how input sequences are transformed into outputs. Derived in matrix transformation form of Sec. 2.2, the M matrix of Mamba2 is defined with cumulative product of A_t matrix. Eq. 6 shows how Eq. 5 can be reformulated into masked matrix with a decay mask L



(a) Decay (b) Softmax (c) Spectral Response

Figure 2. L analysis

, shaped by repeated applications of A_t at time $t \in [T]$. As $A_t = a_t I$ where $a_t \in (0, 1)$, defined for simplicity in Mamba2, each multiplication of A_t introduces a cumulative product effect, resulting in a growing decay for tokens further apart in the sequence. Consequently, earlier tokens experience a rapid loss of influence, forcing the model to prioritize recent tokens. This structural bias naturally steers the model toward focusing on local information, making M act as a low-pass filter.

We further investigate empirically how M matrix behaves in the model. Figure 2 shows heatmap and spectral response of $M = L_d \odot CB^\top$ (decay mask) and $M = f(L_c \odot CB^\top)$ (softmax causal mask). Figure 2(a) reveals a concentration of weights on proximal tokens and similar weights on distant tokens, confirming its bias toward local context. In comparison, Figure 2(b) distribute weights more diverse despite distance. Moreover, as observed in Figure 2(c), the decay mask enforces a lower spectral magnitude compared to the softmax mask, particularly at higher frequencies. This result aligns with our theoretical insight that the decay mask acts as a low-pass filter, suppressing high-frequency components more aggressively. The decay mechanism biases the model towards retaining low-frequency information, which overly emphasizes smooth, long-term interactions at the cost of reduced sensitivity to rapid variations, leading to a loss of important high-frequency details.

3.2. Low pass filter leads Over-Smoothing

In this subsection, we conduct a series of analyses focusing on token-wise similarity and singular value distributions of hidden states to investigate the structural tendencies of Mamba2 and their implications for over-smoothing.

We take Pythia (Biderman et al., 2023) as our counterpart in these analyses due to their variable model scales and disclosed training schemes. For more examples and the details of visualizations, refer to Appendix E.

We first examined the token-wise cosine similarity of hidden states across layers. As shown in Figure 3, the similarity between tokens increases significantly with the depth of

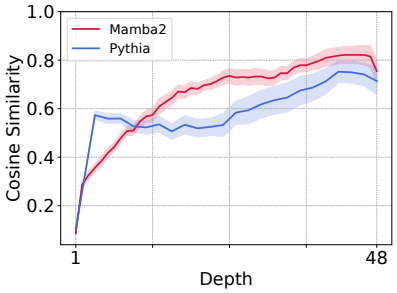


Figure 3. Cosine similarity plots of hidden states for Mamba2-1.3B and Pythia-1.4B. The plot shows the average cosine similarity over 10 sentences; shaded area is the confidence interval.

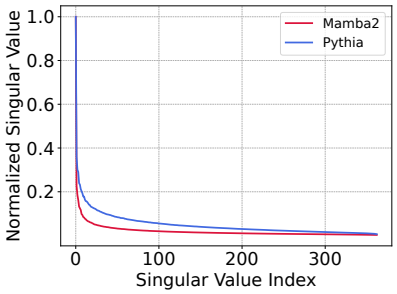


Figure 4. Singular value spectrum of feature maps from Mamba2-1.3B and Pythia-1.4B.

the network. This trend indicates that token representations become progressively more uniform as they pass through layers, a hallmark of over-smoothing. This loss of token diversity suggests that the model struggles to preserve fine-grained information, which is critical for capturing intricate linguistic patterns.

Figure 4 presents the singular value plots, showing a sharp decline in singular values for Mamba2 compared to the baseline. This rapid decrease suggests that the model's feature representations are dominated by top-k single singular value, highlighting a bias toward low-frequency information and limiting representation richness, consistent with cosine similarity and frequency analyses. We provide more

Table 1. Main language modeling results against Pythia and Mamba2. Each task is performed zero-shot. The last column (Avg.) shows the average over all benchmarks that use (normalized) accuracy as the metric. +HE represents our High-Enhance layer is applied to Mamba2.

#Params	Model	Wiki. ppl ↓	LMB. ppl ↓	LMB. acc ↑	PIQA acc ↑	Hella. acc_n ↑	Wino. acc ↑	ARC-e acc ↑	ARC-c acc_n ↑	Avg. 6 tasks ↑
130M	Pythia	33.44	38.00	32.84	61.37	30.30	52.17	43.56	**24.15**	40.73
	Mamba2	35.26	32.18	34.35	62.40	30.89	**52.96**	44.15	23.46	41.37
	Mamba2+HE	**32.76**	**29.65**	**34.78**	**63.33**	**31.76**	51.85	**44.53**	23.72	**41.66**
370M	Pythia	30.70	25.76	37.53	63.82	31.28	51.30	44.28	24.06	42.05
	Mamba2	29.41	20.00	40.09	63.76	33.06	50.91	45.24	23.55	42.77
	Mamba2+HE	**28.87**	**19.33**	**40.73**	**64.69**	**33.83**	**52.17**	**47.60**	**23.72**	**43.79**
1.3B	Pythia	26.57	18.80	41.68	64.53	33.84	51.30	48.11	24.40	43.98
	Mamba2	29.30	**13.35**	**46.15**	64.15	35.01	50.99	47.64	23.29	44.54
	Mamba2+HE	**25.09**	15.71	43.20	**65.40**	**36.25**	**52.49**	**48.32**	**25.00**	**45.11**

visualizations in Appendix E.

4. Layer-wise High Frequency Enhance Filter

To address the over-smoothing issue in Mamba2 models, we propose a high-frequency enhancement method based on gaussian blurring, widely used in various domains (Polesel et al., 2000; Kotera et al., 2000; Deng, 2010). This simple yet effective approach operates after every N_{HE} layers, selectively enhancing high-frequency information to improve performance.

Let $O^{(l)} = [o_1^{(l)}, o_2^{(l)}, ..., o_T^{(l)}]$ be the output sequence of l-th layer of Mamba2, $\text{Mamba}^{(l)}$. For simplicity, we omit layer index in this section, i.e. $O = O^{(l)}$. First, a gaussian kernel is applied to the layer output O to extract low-frequency components O_{low}:

$$O_{\text{low}} = \sum_{k=0}^{n} O[n-k] \cdot \frac{G(k)}{\sum_{j=0}^{n} G(j)}, \quad G(k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{k^2}{2\sigma^2}} \tag{7}$$

where $k \geq 0$ denotes kernel size and σ denotes standard deviation of gaussian distribution. Then, the high-frequency component O_{high} is simply computed with deduction. Finally, we emphasize high-frequency information by applying a tunable weight α, which we call strength:

$$O_{\text{high}} = O - O_{\text{low}}, \quad O_{\text{out}} = O + \alpha \cdot O_{\text{high}} \tag{8}$$

By enhancing high-frequency components, our method mitigates over-smoothing and restores token diversity, enabling Mamba2 to capture richer linguistic features with minimal complexity.

5. Experiments

Following prior works (Gu & Dao, 2023; Yang et al., 2024a), we evaluate our method against original Mamba2 and Pythia on Wikitext (Wiki.) and LAMBADA (LMB.; (Paperno et al., 2016)) perplexity and zero-shot commonsense reasoning tasks, including LAMBADA, PiQA (Bisk et al., 2019), HellaSwag (Hella.; (Zellers et al., 2019)), WinoGrande (Wino.; (Sakaguchi et al., 2019)), ARC-easy(ARC-e), and ARC-challenge (ARC-c) Clark et al. (2018). We report perplexity (ppl) on WikiText and LAMBADA, accuracy normalized by length (acc_n) on HellaSwag and ARC-challenge, and accuracy (acc) on the other tasks (since normalized accuracy is higher for almost all models for these tasks). Avg. denotes average of the 6 zero-shot commonsense reasoning tasks. For training details, refer to Appendix C.1. All results are obtained through `lm-evaluation-harness` (Liang et al., 2023).

Table 1 presents the performance of each model across multiple benchmarks. On every model scale, our enhanced Mamba2 (Mamba2 + HE) outperforms the original Mamba2 (Mamba2) and Pythia in terms of average performance. Except few tasks, our method demonstrates consistent improvements on every tasks, proving that our high-frequency enhancement method effectively addresses the over-smoothing issue, leading to a more expressive and robust model.

6. Conclusion

We conduct a spectral analysis of the Mamba2 architecture to investigate its inherent over-smoothing issue. Our analyses show that Mamba2's decay mechanism acts as a low-pass filter, emphasizing low-frequency components while suppressing high-frequency information. This behavior supports the modeling of smooth, long-range dependencies but limits the model's ability to capture fine-grained interactions. To address this, we introduce a spectral sharpening technique that selectively enhances high-frequency components, effectively mitigating the over-smoothing effect. Experimental results across multiple benchmarks demonstrate consistent performance improvements, validating the effectiveness of our approach. Our findings provide insights into Mamba2's spectral characteristics and open pathways

for further refinement using frequency-based techniques.

Acknowledgments

This work was partly supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korean government (MSIT) (No. RS-2022-II220113, Developing a Sustainable Collaborative Multi-modal Lifelong Learning Framework, 50%; No. RS-2024-00457882, AI Research Hub Project, 20%), and by Samsung Electronics Co., Ltd. (No. G01240136, KAIST Semiconductor Research Fund (2nd), 30%).

References

Ben-Kish, A., Zimerman, I., Abu-Hussein, S., Cohen, N., Globerson, A., Wolf, L., and Giryes, R. Decimamba: Exploring the length extrapolation potential of mamba. *arXiv preprint arXiv:2406.14528*, 2024.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language, 2019. URL https://arxiv.org/abs/1911.11641.

Black, S., Biderman, S., Hallahan, E., Anthony, Q. G., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., Pieler, M. M., Prashanth, U. S., Purohit, S., Reynolds, L., Tow, J., Wang, B., and Weinbach, S. GPT-neox-20b: An open-source autoregressive language model. In *Challenges & Perspectives in Creating Large Language Models*, 2022. URL https://openreview.net/forum?id=HL7IhzS8W5.

Choi, J., Hong, S., Park, N., and Cho, S.-B. Gread: Graph neural reaction-diffusion networks. In *International Conference on Machine Learning*, pp. 5722–5747. PMLR, 2023a.

Choi, J., Wi, H., Kim, J., Shin, Y., Lee, K., Trask, N., and Park, N. Graph convolutions enrich the self-attention in transformers! *arXiv preprint arXiv:2312.04234*, 2023b.

Cirone, N. M., Orvieto, A., Walker, B., Salvi, C., and Lyons, T. Theoretical foundations of deep selective state-space models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=3SzrqwupUx.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Dao, T. and Gu, A. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=ztn8FCR1td.

Dao, T. and Gu, A. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024b.

Deng, G. A generalized unsharp masking algorithm. *IEEE transactions on Image Processing*, 20(5):1249–1261, 2010.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Re, C. Hungry hungry hippos: Towards language modeling with state space models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=COZDy0WYGg.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Glorioso, P., Anthony, Q., Tokpanov, Y., Whittington, J., Pilault, J., Ibrahim, A., and Millidge, B. Zamba: A compact 7b ssm hybrid model. *arXiv preprint arXiv:2405.16712*, 2024.

Grazzi, R., Siems, J. N., Schrodi, S., Brox, T., and Hutter, F. Is mamba capable of in-context learning? In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. URL https://openreview.net/forum?id=Iia0cnjMh2.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Gu, A., Goel, K., Gupta, A., and Ré, C. On the parameterization and initialization of diagonal state space models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 35971–35983. Curran Associates, Inc., 2022a.

Gu, A., Goel, K., and Re, C. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Guo, X., Wang, Y., Du, T., and Wang, Y. Contranorm: A contrastive learning perspective on oversmoothing and beyond. In *The Eleventh International Conference on Learning Representations*, 2023.

Hwang, S., Lahoti, A., Puduppully, R., Dao, T., and Gu, A. Hydra: Bidirectional state space models through generalized matrix mixers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=preo49P1VY.

Jelassi, S., Brandfonbrener, D., Kakade, S. M., and eran malach. Repeat after me: Transformers are better than state space models at copying. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=duRRoGeoQT.

Kotera, H., Yamada, Y., and Shimo, K. Adaptive edge sharpening by multiple gaussian filters. In *NIP & Digital Fabrication Conference*, volume 16, pp. 814–817. Society of Imaging Science and Technology, 2000.

Lee, I., Jiang, N., and Berg-Kirkpatrick, T. Is attention required for ICL? exploring the relationship between model architecture and in-context learning ability. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Qwq4cpLtoX.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C. A., Manning, C. D., Re, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., WANG, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N. S., Khattab, O., Henderson, P., Huang, Q., Chi, R. A., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=iO4LZibEqW. Featured Certification, Expert Certification.

Lieber, O., Lenz, B., Bata, H., Cohen, G., Osin, J., Dalmedigos, I., Safahi, E., Meirom, S., Belinkov, Y., Shalev-Shwartz, S., et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Merrill, W., Petty, J., and Sabharwal, A. The illusion of state in state-space models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2025.

Nambiar, A., Heflin, M., Liu, S., Maslov, S., Hopkins, M., and Ritz, A. Transforming the language of life: transformer neural networks for protein prediction tasks. In *Proceedings of the 11th ACM international conference on bioinformatics, computational biology and health informatics*, pp. 1–8, 2020.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, N. Q., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Erk, K. and Smith, N. A. (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Park, J., Park, J., Xiong, Z., Lee, N., Cho, J., Oymak, S., Lee, K., and Papailiopoulos, D. Can mamba learn how to learn? a comparative study on in-context learning tasks. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=GbFluKMmtE.

Patro, B. N. and Agneeswaran, V. S. Simba: Simplified mamba-based architecture for vision and multivariate time series. *arXiv preprint arXiv:2403.15360*, 2024.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Derczynski, L., Du, X., Grella, M., Gv, K., He, X., Hou, H., Kazienko, P., Kocon, J., Kong, J., Koptyra, B., Lau,

H., Lin, J., Mantri, K. S. I., Mom, F., Saito, A., Song, G., Tang, X., Wind, J., Woźniak, S., Zhang, Z., Zhou, Q., Zhu, J., and Zhu, R.-J. RWKV: Reinventing RNNs for the transformer era. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.936. URL https://aclanthology.org/2023.findings-emnlp.936/.

Polesel, A., Ramponi, G., and Mathews, V. J. Image enhancement via adaptive unsharp masking. *IEEE transactions on image processing*, 9(3):505–510, 2000.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: towards larger convolutional language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023a.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023b.

Ren, L., Liu, Y., Lu, Y., Shen, Y., Liang, C., and Chen, W. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *arXiv preprint arXiv:2406.07522*, 2024.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

Shi, H., GAO, J., Xu, H., Liang, X., Li, Z., Kong, L., Lee, S. M., and Kwok, J. Revisiting over-smoothing in bert from the perspective of graph. In *International Conference on Learning Representations*, 2022.

Sieber, J., Alonso, C. A., Didier, A., Zeilinger, M., and Orvieto, A. Understanding the differences in foundation models: Attention, state space models, and recurrent neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=iF7MnXnxRw.

Smith, J. T., Warrington, A., and Linderman, S. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive network: A successor to transformer for large language models, 2024. URL https://openreview.net/forum?id=UU9Icwbhin.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qVyeW-grC2k.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Waleffe, R., Byeon, W., Riach, D., Norick, B., Korthikanti, V., Dao, T., Gu, A., Hatamizadeh, A., Singh, S., Narayanan, D., Kulshreshtha, G., Singh, V., Casper, J., Kautz, J., Shoeybi, M., and Catanzaro, B. An empirical study of mamba-based language models, 2024. URL https://arxiv.org/abs/2406.07887.

Wang, J., Paliotta, D., May, A., Rush, A. M., and Dao, T. The mamba in the llama: Distilling and accelerating hybrid models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=uAzhODjALU.

Wang, P., Zheng, W., Chen, T., and Wang, Z. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *International Conference on Learning Representations*, 2022.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Wi, H., Choi, J., and Park, N. Learning advanced self-attention for linear transformers in the singular value domain. *arXiv preprint arXiv:2505.08516*, 2025.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=ia5XvxFUJT.

Yang, S., Wang, B., Zhang, Y., Shen, Y., and Kim, Y. Parallelizing linear transformers with the delta rule over sequence length, 2024b. URL https://arxiv.org/abs/2406.06484.

Yu, A., Lyu, D., Lim, S. H., Mahoney, M. W., and Erichson, N. B. Tuning frequency bias of state space models. *arXiv preprint arXiv:2410.02035*, 2024.

Yu, Y.-Y., Choi, J., Cho, W., Lee, K., Kim, N., Chang, K., Woo, C.-S., Kim, I., Lee, S.-W., Yang, J.-Y., et al. Learning flexible body collision dynamics with hierarchical contact mesh transformer. *arXiv preprint arXiv:2312.12467*, 2023.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Zhang, B. and Sennrich, R. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., and Wang, X. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=YbHCqn4qF4.

A. Detailed explanation of Mamba2 architecture

A.1. Structured State Space Models

Structured state space models represent a new category of sequence models in deep learning, drawing connections to RNNs, CNNs, and traditional state space models. These models are motivated by a specific continuous system that processes a one-dimensional input sequence $x \in \mathbb{R}^T$ into an output sequence $y \in \mathbb{R}^T$ via an implicit latent state $h \in \mathbb{R}^{T \times N}$.

Eq. 9 is a fundamental representation of organized SSMs.

$$\begin{aligned} h'(t) &= \bar{A}h(t-1) + \bar{B}x(t) \\ y(t) &= C^\top h(t) \end{aligned} \tag{9}$$

$$\begin{aligned} h_t &= Ah_{t-1} + Bx_t \\ y_t &= C^\top h_t \end{aligned} \tag{10}$$

where $A \in \mathbb{R}^{N \times N}$, $B \in \mathbb{R}^{N \times 1}$, $C \in \mathbb{R}^{N \times 1}$. This continuous SSMs in Eq. 9 are discretized to Eq. 10 through fixed formulas: $A = f_A(\Delta, \bar{A})$, $B = f_B(\Delta, \bar{B})$.

A.2. Full architecture of Mamba2

Given an input sequence $U = [u_1, u_2, ..., u_T]^\top \in \mathrm{R}^{T \times d}$, a Mamba2 block with d channels is built on top of the S6 layer via the following formula, generating output sequence $O = [o_1, o_2, ..., o_T]^\top \in \mathbb{R}^{T \times d}$:

$$\begin{aligned} h_t &= A_t h_{t-1} + B_t x_t^\top \in \mathbb{R}^{N \times P}, \\ y_t &= C_t^\top h_t + D \odot x_t \in \mathbb{R}^P \\ o_t &= W_o(\mathrm{Norm}(y_t \odot W_z u_t)) \in \mathbb{R}^d \end{aligned} \tag{11}$$

where $D \in \mathbb{R}^P$, $W_x, W_z \in \mathrm{R}^{d \times P}$, $W_o \in \mathbb{R}^{P \times d}$ are trainable parameters. Each Mamba2 block consists of H heads, so that $H \times P = d$, which are computed in parallel, the result of which is summed together. We can specify how each matrices are created for each head:

$$\begin{aligned} \bar{A}_t &= a_t I \in \mathbb{R}^{N \times N} \\ a_t &= \exp(-\Delta_t \exp(A)) \in \mathbb{R} \\ B_t &= \Delta_t \bar{B}_t \in \mathbb{R}^{N \times 1} \\ \bar{B}_t &= \sigma(\mathrm{Conv}(W_B u_t)) \in \mathbb{R}^{N \times 1} \end{aligned} \tag{12}$$

$$\begin{aligned} C_t &= \sigma(\mathrm{Conv}(W_C u_t)) \in \mathbb{R}^{N \times 1} \\ \Delta_t &= \mathrm{Softplus}(W_\Delta u_t + b_\Delta) \in \mathbb{R} \\ x_t &= \sigma(\mathrm{Conv}(W_x u_t)) \in \mathbb{R}^{P \times 1} \end{aligned} \tag{13}$$

where $W_B, W_C \in \mathbb{R}^{N \times d}$, $W_\Delta \in \mathbb{R}^{1 \times d}$. σ denotes SiLU activation function and $\mathrm{Conv}(\cdot)$ denotes a channel-wise one-dimensional convolution. By Δ, Mamba2 implements input-dependent selection mechanism. A_t performs as decay-ratio as it is cumulatively multiplied. Ben-Kish et al. (2024) elaborate the condition of a_t. For computational stability, $\Delta > 0$ and $A < 0$ is guaranteed in original implementation. Therefore, we can conclude $a_t \in (0, 1)$.

Using this Mamba2 block, we can derive layer-wise Mamba2 architecture with L layers as below. For initial input, input sequence is $I = [i_0, i_1, ..., i_T] \in \mathbb{R}^T$ where $i_t \in [V]$ and we have $U^{(l-1)} = [u_1^{(l-1)}, u_2^{(l-1)}, ...u_T^{(l-1)}]$ as input sequence for the l-th layer. $O^{(l)} = [o_1^{(l)}, o_2^{(l)}, ...o_T^{(l)}]$ serves as output sequence of l-th Mamba2 layer $\mathrm{Mamba}^{(l)}$, V denotes vocab size and $P \in \mathbb{R}^{T \times V}$ denotes final logits.

$$\begin{aligned} U^{(0)} &= \mathrm{Embedding}_{in}(I) \in \mathbb{R}^{T \times d} \\ O^{(l)} &= \mathrm{Mamba}^{(l)}(\mathrm{Norm}[U^{(l-1)}]) \in \mathbb{R}^{T \times d} \\ P &= \mathrm{Embedding}_{out}(\mathrm{Norm}([O^{(L)}]) \in \mathbb{R}^{T \times V} \end{aligned} \tag{14}$$

Here, the output of the l-th layer is used as the input for the $l+1$-th layer, i.e. $O^{(l)} = U^{(l)}$.

A.3. Matrix transformation derivation

We can derive matrix transformation form using these equations. By definition, $h_0 = B_0 x_0$. By induction,

$$\begin{aligned} h_t &= A_t \dots A_1 B_0 x_0 + A_t \dots A_2 B_1 x_1 + \cdots + A_t A_{t-1} B_{t-2} x_{t-2} + A_t B_{t-1} x_{t-1} + B_t x_t \\ &= \sum_{s=0}^{t} A_{t:s}^{\times} B_s x_s \end{aligned} \tag{15}$$

To produce y_t, we multiply C_t. Then, by vectorizing over sequence length $t \in [T]$, we can produce the matrix transfomration form of SSMs:

$$\begin{aligned} y_t &= \sum_{s=0}^{t} C_t^{\top} A_{t:s}^{\times} B_s x_s \\ Y &= \mathrm{S6}(X) = MX \\ M_{ji} &:= C_j^{\top} A_j \cdots A_{i+1} B_i \end{aligned} \tag{16}$$

In this paper, we refer to this matrix as the M matrix, which serves as our primary subject of analysis.

B. Extended Related Works

B.1. Frequency Analysis

We use Fourier transform as our main analytic tool. The Discrete Fourier Transform and Inverse Discrete Fourier Transform can be denoted as $\mathcal{F} : \mathbb{R}^n \to \mathbb{C}^n$ and $\mathcal{F}^{-1} : \mathbb{C}^n \to \mathbb{R}^n$, respectively. Following Wang et al. (2022), we regard matrices as multi-channel signals. Applying DFT to a signal x can be interpreted as left-multiplying a DFT matrix which has rows of Fourier basis $f_k = [e^{2\pi j(k-1)\cdot 0} \dots e^{2\pi j(k-1)\cdot(n-1)}]^T/\sqrt{n} \in \mathbb{R}^n$. Here, k denotes the k-th row of DFT matrix and j is imaginary unit. In a matrix form, we can write DFT matrix as below:

$$\mathrm{DFT} = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{2\pi j} & e^{2\pi j(n-1)} & \dots & e^{2\pi j(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{2\pi j(k-1)\cdot 1} & e^{2\pi j(k-1)\cdot(n-1)} & \dots & e^{2\pi j(k-1)\cdot(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{2\pi j(n-1)} & e^{2\pi j(n-1)^2} & \dots & e^{2\pi j(n-1)^2} \end{bmatrix} \tag{17}$$

and the inverse DFT is $\mathrm{DFT}^{-1} = \mathrm{DFT}$.

We can also define operators $\mathcal{DC}[\cdot]$ and $\mathcal{HC}[\cdot]$ using DFT matrix. The spectrum of z is $\tilde{z} = \mathcal{F}z$, and $\tilde{z}_{dc} \in \mathbb{C}$, $\tilde{z}_{hc} \in \mathbb{C}^{n-1}$ means the first and rest elements of spectrum. The Direct-Current component of signal z can be defined as $\mathcal{DC}[\mathbf{z}] = \tilde{z}_{dc} f_1$. This can also be interpreted as below:

$$\begin{aligned} \mathcal{DC}[x] &= \mathrm{DFT}^{-1}\,\mathrm{diag}(1, 0, \dots, 0)\mathrm{DFT}, \\ x &= \frac{1}{n} 1 1^T x, \end{aligned} \tag{18}$$

The complementary high-frquency component is defined as $\mathcal{HC}[\mathbf{z} = [f_2 ... f_n]]\tilde{z}_{hc} \in \mathbb{C}^n$. Similarly, we can write $HC[\cdot]$ as below:

$$\begin{aligned} \mathcal{HC}[x] &= \mathrm{DFT}^{-1}\,\mathrm{diag}(0, 1, \dots, 1)\mathrm{DFT}\,x \\ &= \mathrm{DFT}^{-1}\big(I - \mathrm{diag}(1, 0, \dots, 0)\big)\mathrm{DFT}\,x \\ &= I - \frac{1}{n} 1 1^T x, \end{aligned} \tag{19}$$

Low pass filter in signal processing is a system that presereves low-frequency components while suppresses high-frequency components. In this perspective, when other components have significantly low frequency compared to $\mathcal{DC}[\cdot]$, we can call this filter as a low pass filter.

B.2. Foundational Model based on State Space Models

Transformers (Vaswani et al., 2017) have become foundational across a wide range of domains (Yu et al., 2023; Nambiar et al., 2020; Wen et al., 2022; Dosovitskiy et al., 2020), due to their strong performance and scalability. However, their quadratic complexity with respect to sequence length has led to the development of various sub-quadratic models. Among them, models based on SSMs (State Space Models) originated from HiPPO theory (Gu et al., 2020), approaching sequence modeling through the concept of dynamics. S4 (Gu et al., 2022b) enabled parallel computation by defining a linear time-invariant matrix as a kernel and demonstrated strong performance in the Long Range Arena benchmark. S4D (Gu et al., 2022a) optimized computation by diagonalizing the parameter matrices of S4. Hyena Hierarchy (Poli et al., 2023b) introduced a fully convolutional model, leveraging convolution to handle long sequences. H3 (Fu et al., 2023) focused on language modeling, identifying and addressing limitations of SSMs. These models achieve efficient computation, scaling nearly linearly with sequence length, and can operate as either convolutions or recurrences. They excel in tasks requiring long-range dependency modeling, as evidenced by strong performance on benchmarks like the Long Range Arena (Tay et al., 2021). However, SSMs face challenges in representing discrete, information-rich inputs such as text.

Amid this landscape, Mamba (Gu & Dao, 2023) introduced an input-dependent selection mechanism, overcoming the challenges where traditional SSMs struggled due to their reliance on processing all inputs with a single state. With hardware-efficient training, Mamba2 (Dao & Gu, 2024b) showcased superior modeling performance across diverse tasks not only in language modeling, but also in various domains such as image processing, DNA modeling, and time-series forecasting. Following its success, various models leveraging Mamba2 for different tasks have emerged (Zhu et al., 2024; Patro & Agneeswaran, 2024; Hwang et al., 2024), with ongoing research exploring hybrid models (Glorioso et al., 2024; Ren et al., 2024; Lieber et al., 2024), distillation (Wang et al., 2024; Dao & Gu, 2024a), and other approaches (Ben-Kish et al., 2024).

B.3. Analysis of Mamba2 and SSMs

Many experimental and theoretical researches have emerged on Mamba2's ability on language modeling. Waleffe et al. (2024) scrutinized Mamba2's empirical ability on various tasks and analyzed weakness and strength of Mamba2 model and also demonstrated reasonable performance of Mamba-Transformer hybrid model. On the other hand, Merrill et al. (2025) insist inherent limitation of SSMs due to their architectures in the perspective of algorithmic problems. Park et al. (2024); Grazzi et al. (2024) study on in-context learning ability of Mamba2 model, showing that Mamba2 has comparable ICL ability on variant tasks. Nevertheless, Mamba2 showed poor performance on associative recall and selective copying of ICL tasks. (Lee et al., 2024; Jelassi et al., 2024)

Another line of research focuses on theoretical analysis. Cirone et al. (2024) analyzed SSMs ability with CDE, theoretically explaining where Deep SSM's learning ability emerges. Sieber et al. (2024) organized recent foundation models as a unified framework called Dynamical Systems Framework. They derived thorough comparison with a common framework, which stems from dynamical systems, whereas Dao & Gu (2024b) tries to generalize as sequence-transformation format. Yu et al. (2024) concentrates on frequency bias of State Space Models. They go through mathematical formulation of State Space models and find out that SSMs' initialization injects bias to model which cannot be overcome by training. Though they scope on frequency bias, their study does not extend to current Mamba2 architecture. In this paper, we focus on spectral properties of Mamba2 architecture, thereby addressing fundamental issues in Mamba.

C. Experimental Setup

C.1. Training details

In this section, we elaborate the training details for models in Sec. 5. Mamba2 and our models' are trained on the Pile dataset, deduplicated version (approximately 200B tokens) (Gao et al., 2020). We used the basic configuration provided from (Dao & Gu, 2024b) for Mamba2 and Mamba2 + HE (Ours). We used GPTNeoX 20B tokenizer (Black et al., 2022) following prior work. We trained our models with eight A100 GPUs for language modeling experiments on every scale.

For model size 130M and 370M, we used AdamW for optimization, $\beta \in [0.9, 0.95]$ following Mamba2, weight decay of 0.1 with a peak learning rate of 4.8e-3. We used linear learning rate warmup with cosine decay, with a warm-up phase of 375M tokens. The 130M and 370M model were trained with a batch size of 2^{23} tokens (# sequences $\times$ sequence length) and the number of training steps as 23,521 (# tokens / # tokens in one batch) steps. This exactly equals 1 epoch of training for the Pile dataset. We used 1.0 as gradient clip value. For Pythia, all other settings are same without learning rate for better performance. We used learning rate $6e-4$ for model size 130M, $8e-4$ for model size 370M. For model size 1.3B, we also

# Params	Model	Training steps	Peak LR	Batch size	# Tokens	Weight decay	Gradient clip
130M	Pythia	23,251	6e-4	8M	∼200B	0.1	1.0
	Mamba2	23,251	4.8e-3	8M	∼200B	0.1	1.0
	Mamba2+HE	23,251	4.8e-3	8M	∼200B	0.1	1.0
370M	Pythia	23,251	8e-4	8M	∼200B	0.1	1.0
	Mamba2	23,251	4.8e-3	8M	∼200B	0.1	1.0
	Mamba2+HE	23,251	4.8e-3	8M	∼200B	0.1	1.0
1.3B	Pythia	47,042	2e-4	4M	∼200B	0.0	0.0
	Mamba2	47,042	8e-4	4M	∼200B	0.0	0.0
	Mamba2+HE	47,042	8e-4	4M	∼200B	0.0	0.0

Table 2. Summary of training settings.

used AdamW for optimization, but with $\beta \in [0.9, 0.999]$ following default value of PyTorch, weight decay of 0.0 with a peak learning rate 8e-4 for better performance. The batch size used for training was 2^{22} tokens and the number of training steps was 47,042, which also equals 1 epoch of training with the Pile dataset. We used linear learning rate scheduler, without warm-up steps. We did not utilize gradient clipping for this setting. We used learning rate $2e-4$ for Pythia, model size 1.3B. Our proposed method consists of 3 hyperparameters: kernel size k, standard deviation of gaussian distribution σ, and enhance strength α. For our results presented in this paper, we used $k = 3,\ \sigma = 3,\ \alpha = 1$. Our implementation is publicly available at `https://github.com/sjiinkim/mamba2-high-freq-enhance`.

C.2. Evaluation

For evaluation of language modeling performance, we used `lm-evaluation-harness` (Liang et al., 2023). We provide details of evaluation tasks below.

- WikiText (Merity et al., 2017): A dataset consisting of high-quality, clean text extracted from Wikipedia articles, commonly used to evaluate language modeling tasks by measuring a model's ability to predict and generate coherent and fluent text.
- LAMBADA (Paperno et al., 2016): A text completion task that measures a model's ability to predict the final word of a passage, requiring comprehension of the context, commonsense reasoning, as well as the ability to generate text coherently.
- PIQA (Bisk et al., 2019): A physical commonsense reasoning task focused on selecting the most plausible solution to everyday scenarios.
- HellaSwag (Zellers et al., 2019): A multiple-choice task that evaluates a model's ability to select the most coherent continuation of a given situation based on commonsense and narrative reasoning.
- WinoGrande (Sakaguchi et al., 2019): An expanded version of the Winograd Schema Challenge: a pronoun resolution task designed to test commonsense reasoning by identifying which noun a pronoun refers to in a given sentence.
- ARC-easy (Clark et al., 2018): A subset of the AI2 Reasoning Challenge focusing on questions that require basic scientific and commonsense knowledge.
- ARC-challenge (Clark et al., 2018): A more difficult subset of the AI2 Reasoning Challenge that tests advanced reasoning and deep understanding of scientific and commonsense knowledge.

D. Limitations and future works

We discuss the limitations of our study and propose directions for future work. First, our study lacks a theoretical proof that Mamba2 operates as a low-pass filter, which leads to over-smoothing. Additionally, we did not conduct an ablation study to analyze the hyperparameter sensitivity of our proposed methodology due to limited time and resources. For future work, we plan to address this limitation by conducting comprehensive sensitivity analyses and refining our approach accordingly. Furthermore, we aim to explore the connection between over-smoothing and over-squashing.

E. More visualizations

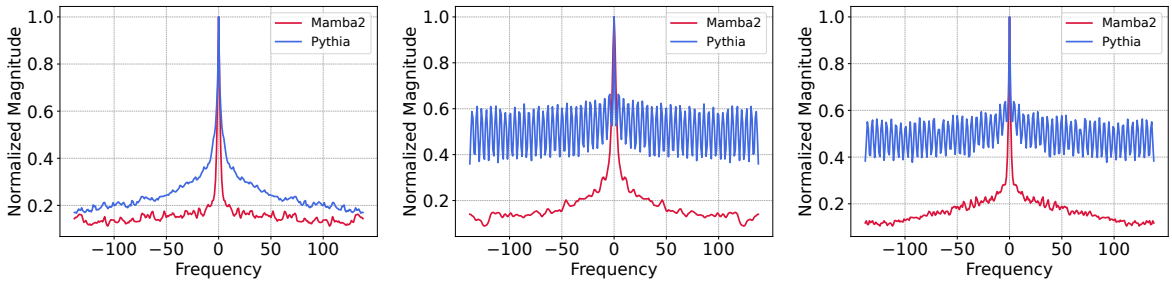
In this appendix, we provide more visualizations and how these metrics are calculated.

Visualization on specturm We compute the spectrum of M matrix by regarding M as a linear filter. The Fourier-domain response of a linear filter is another linear kernel $\Lambda = \mathcal{F}M\mathcal{F}^{-1}$. Given a spectrum $\tilde{x} = \mathcal{F}x$, the i-th frequency response will be $\Lambda_i x$ where Λ_i is the i-th row of Λ. Therefore, $||\Lambda_i||_2$ is used to evaluate the spectral response intensity of the i-th frequency band. We provide how M matrix performs as a filter in Mamba2-1.3B over all layers in Figure 8. In addition, we provide more comparison of Mamba2-130M and Pythia-160M in Figure 5.

How similarity curves are calculated Following Wang et al. (2022), we compute the pairwise cosine similarity between every two different tokens. Given the layer index l and its output $O^{(l)}$, the cosine similarity is estimated by:

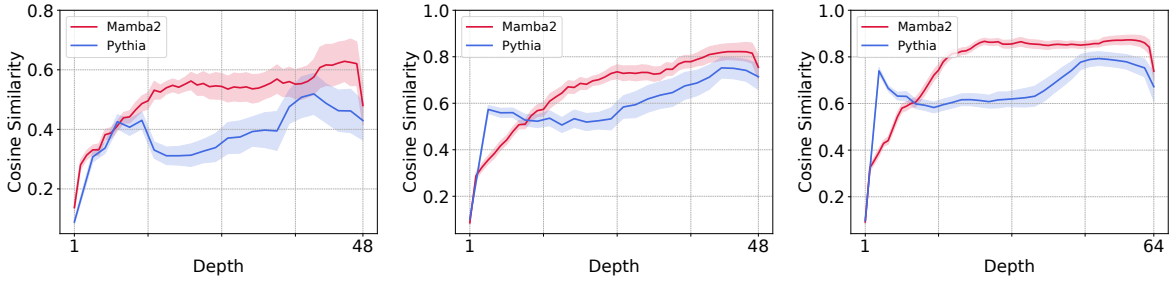
$$\text{CosSim}^l = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{|X_i^{(l)\top} X_j^{(l)}|}{||X_i^{(l)}||_2 ||X_j^{(l)}||_2} \tag{20}$$

where $X_i^{(l)}$ denotes the i-th row of $X^{(l)}$. This measures the token-wise cosine similarity: how similar the feature representations of two tokens are. We demonstrate visualizations of this metric in Figure 3 and Figure 7. We plot average cosine similarity of 10 sentences randomly selected from the Pile dataset with 95% confidence interval.



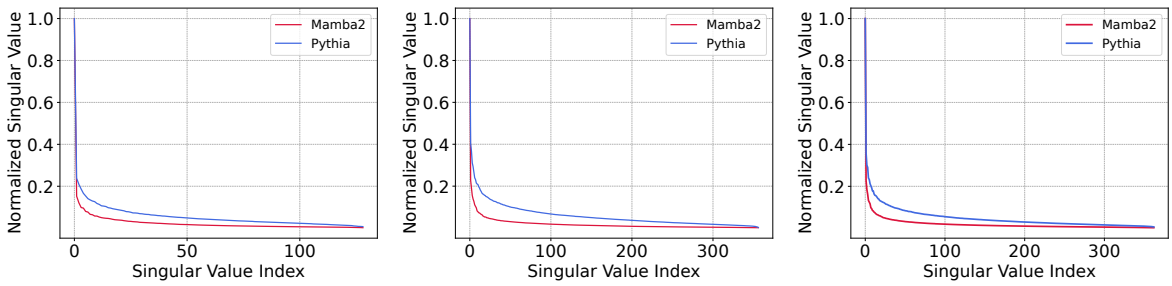
(a) Layer 2 (b) Layer 5 (c) Layer 8

Figure 5. Comparison on spectral response of Mamba2-130M and Pythia-160M.



(a) 370M (b) 1.3B (c) 2.8B

Figure 6. Cosine similarity plots of feature maps across various model sizes.



(a) 130M (b) 370M (c) 1.3B

Figure 7. Singular value plots of feature maps across various model sizes.

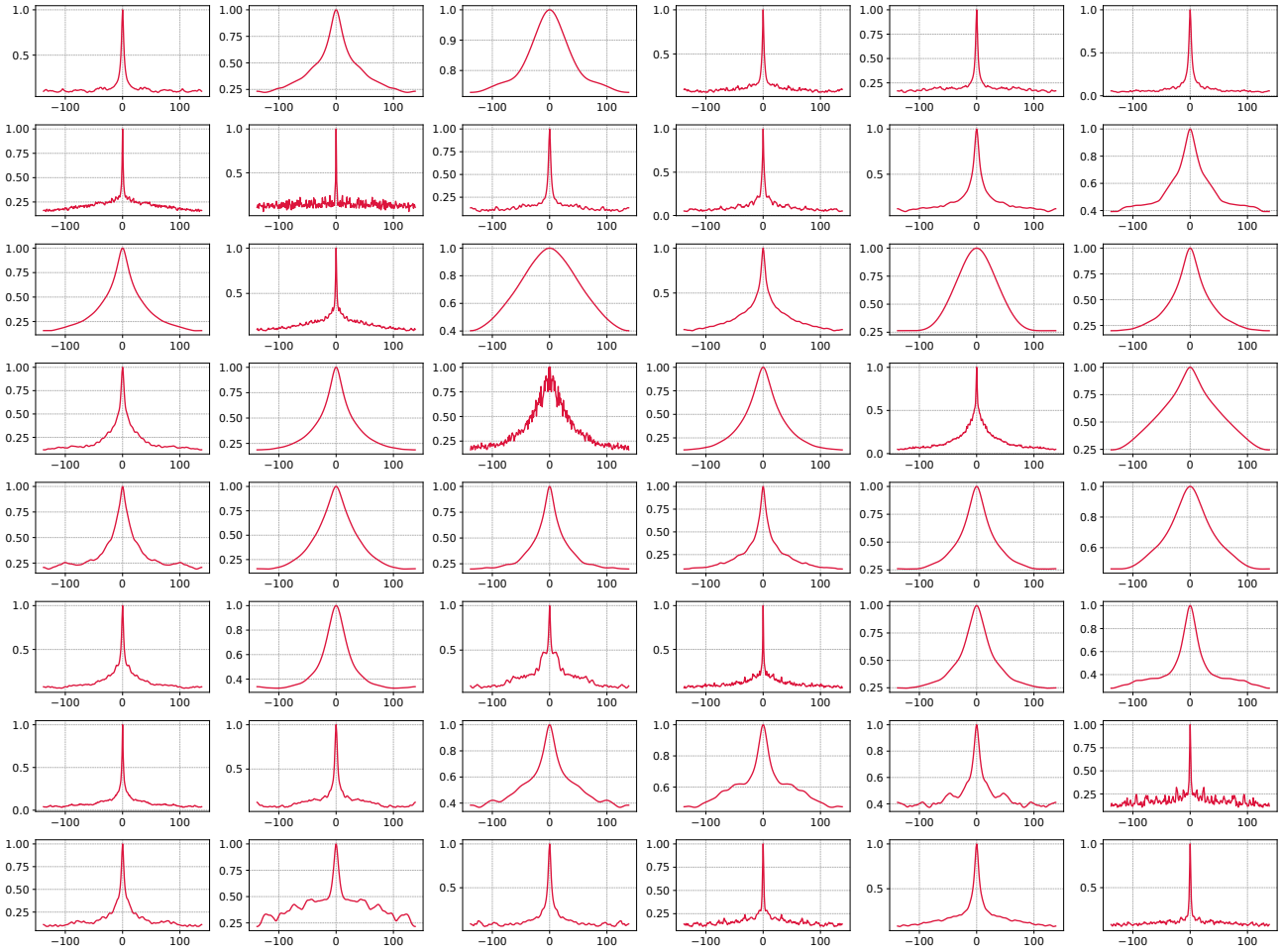


Figure 8. Spectral response of Mamba2-1.3B for all layers, arranged sequentially in row-wise order