Spectral Analysis of State Space Models in Language Modeling: Training Dynamics and Stability Properties

Zayan Hasan
Computer Science
Academies Of Loudoun
Ashburn, Virginia
zayanuhasan@gmail.com

Aneesh Chatrathi
Computer Science
Northern Virginia Community College
Ashburn, Virginia
chatrathi.aneesh@gmail.com

Aniketh Malipeddi
Computer Science
Rock Ridge High School
Ashburn, Virginia
aniketh.malipeddi@gmail.com

***Abstract*—State Space Models (SSMs) have emerged as a new linear computational complexity transformer rival to sequence modeling on long sequences with competitive performance. The dynamics of training and stability properties of SSMs remain poorly understood from a spectral perspective. This work presents the first wide reaching spectral analysis of SSM based language models. Providing a systematic framework to examine how eigenvalue distributions and spectral radii evolve during training, through experiments on a 737K parameter SSM model having 3 layers, state space dimension 128, and model space dimension 8, it was discovered that although the minority of the state matrices learned lead to theoretical spectral stability with mean spectral radius 1.078. The model demonstrates excellent convergence however, reducing training loss from 3.127 to 0.305 using 100 epochs. The eigenvalue analysis demonstrates common clustering in the negative real axis with concentration centered about negative 0.8, exhibiting a bimodal spectral radius distribution exhibiting systematic behavior in SSM dynamics. The key result portrays that SSMs operate efficiently in scenarios such as these. The selective mechanism provides adaptive control that prevents mathematical instabilities from causing training divergence. This renders assumptions of classical neural network stability hard to maintain and makes spectral analysis an essential for understanding similar model behavior. This work provides practical insight toward constructing more principled, stability aware designs for such models and frameworks.**

***Index Terms*—State Space Models, Spectral Analysis, Language Modeling, Training Dynamics, Eigenvalue Evolution**

I. Introduction

State Space Models (SSMs) have gained popularity in recent years in the machine learning community as a serious contender to the Transformer architecture for sequence modeling tasks [1,2]. Unlike Transformers with quadratic complexity of sequence length using self-attention mechanisms, SSMs offer linear computational complexity at the expense of marginally reduced performance on long-range dependencies [3]. This efficiency has rendered SSMs especially relevant in language modeling tasks where working with long sequences is computationally not viable for standard architectures. The key innovation of modern SSMs is their selection mechanism, which allows dynamic adaptation of state transition matrices based on input content [4]. While empirical SSM performance has been demonstrated across a range of tasks, training dynamics of SSMs are theoretically poorly understood. Specifically, the spectral properties of learned state matrices that determine ultimately system stability and long-term behavior are systematically not investigated in language models. Motivation and Research Questions. A priori knowledge of the spectral properties of SSM training has several reasons. First, eigenvalue distribution of state matrices has direct consequences on convergence and stability properties of the underlying dynamics. Second, examining the spectrum could provide some insight into why SSMs can learn about structure in language despite instability of mathematics. Finally, such an examination could guide the design of more principled training regimens and architecture modifications. This work addresses how do the spectral properties of SSM state matrices evolve during language model training, what the relationship between spectral stability and language modeling performance is, and whether spectral analysis can provide insights for improving similar architectures outlined in previous literature that take inspiration from models such as gated RNNs [5,6].

While SSMs have been studied for their computational efficiency and empirical performance, spectral analysis of neural network training dynamics has primarily focused on traditional architectures. Recent work on neural network stability has examined the spectral properties of recurrent networks [7] and attention mechanisms, but no prior work has systematically analyzed the spectral evolution of SSMs during language model training. This work bridges the gap by providing the first spectral characterization of SSM training dynamics.

II. Methods

The spectral characteristics of specific State Space Models (SSMs) intended for language modeling tasks are examined in this work. A continuous-time linear dynamical system is implemented by the core SSM block and then discretized for neural network training. SSMs' mathematical underpinnings offer computational advantages over conventional attention-based architectures while delivering a principled approach to

sequence modeling. The following are the definitions of the basic SSM equations in continuous time:

$$\frac{dx(t)}{dt} = \mathbf{A}x(t) + \mathbf{B}u(t) \tag{1}$$

$$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t) \tag{2}$$

Where the state vector, input, and output are defined with appropriate dimensionalities:

$$x(t) \in \mathbb{R}^{d_{\text{state}}}, \quad u(t) \in \mathbb{R}^{d_{\text{model}}}, \quad y(t) \in \mathbb{R}^{d_{\text{model}}} \tag{3}$$

The key innovation distinguishing modern SSMs from classical state space approaches is the selective mechanism, where state matrices become dynamically dependent on the input content. This input-dependency is formalized as:

$$\mathbf{A}_t = f_A(\mathbf{x}_t), \quad \mathbf{B}_t = f_B(\mathbf{x}_t), \quad \mathbf{C}_t = f_C(\mathbf{x}_t) \tag{4}$$

The specific parameterization employed in this implementation ensures both numerical stability and theoretical grounding. The state transition matrix can be expressed as follows.

$$\mathbf{A}_t = -\text{softplus}(\mathbf{W}_A\mathbf{x}_t + \mathbf{b}_A) - \epsilon \tag{5}$$

Where the negative softplus activation ensures negative eigenvalues for enhanced stability, and the regularization term is set to 0.1. Furthermore, the input and output matrices employ bounded activations to prevent numerical instabilities.

$$\mathbf{B}_t = \tanh(\mathbf{W}_B\mathbf{x}_t + \mathbf{b}_B) \tag{6}$$

$$\mathbf{C}_t = \tanh(\mathbf{W}_C\mathbf{x}_t + \mathbf{b}_C) \tag{7}$$

For neural network implementation, the continuous-time system requires discretization. This work employs the zero-order hold (ZOH) method with a fixed time step to ensure consistent temporal dynamics. The discretization process transforms the continuous matrices where the timestep is fixed at 0.1.

$$\mathbf{A}_d = \mathbf{I} + \Delta t \cdot \mathbf{A}_t \tag{8}$$

$$\mathbf{B}_d = \Delta t \cdot \mathbf{B}_t \tag{9}$$

The resulting discrete time state evolution follows the recurrence relation:

$$\mathbf{h}_{t+1} = \mathbf{A}_d\mathbf{h}_t + \mathbf{B}_d\mathbf{x}_t \tag{10}$$

With output computation given by:

$$\mathbf{y}_t = \mathbf{C}_t\mathbf{h}_t + \mathbf{D}\mathbf{x}_t \tag{11}$$

In terms of the complete SSM-based language model, it integrates multiple architectural components to enable effective sequence modeling. The model consists of the following layers in sequential order and the token embedding layer maps discrete vocabulary indices to continuous representations.

$$\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d_{\text{model}}} \tag{12}$$

The vocabulary size was set to 2000. Further, the positional embeddings provide sequence position information where the maximum sequence length is 512.

$$\mathbf{P} \in \mathbb{R}^{L_{\max} \times d_{\text{model}}} \tag{13}$$

The model employs multiple identical SSM blocks with residual connections. Each of the 3 layers implements a transformation with layer normalization applied before the SSM operation.

$$\mathbf{h}^{(\ell)} = \text{LayerNorm}(\mathbf{h}^{(\ell-1)}) \tag{14}$$

$$\mathbf{z}^{(\ell)} = \text{SSMBlock}(\mathbf{h}^{(\ell)}) + \mathbf{h}^{(\ell-1)} \tag{15}$$

Then the final output probabilities are computed through a linear projection followed by softmax normalization.

$$\mathbf{p}_t = \text{softmax}(\mathbf{W}_{\text{out}}\text{LayerNorm}(\mathbf{z}^{(N)})) \tag{16}$$

The spectral analysis framework provides theoretical insights into the stability and dynamics of the SSM training process. This analysis focuses on the eigenvalue properties of the learned state transition matrices, which fundamentally determine the long term behavior of the dynamical system. These properties of the SSM are characterized through the eigenvalue decomposition of the discrete time state matrix. For each matrix, the eigenvalue equation, eigenvalues and eigenvectors are denoted below.

$$\mathbf{A}_d\mathbf{v}_i = \lambda_i\mathbf{v}_i \tag{17}$$

$$\{\lambda_i\}_{i=1}^{d_{\text{state}}}, \quad \{\mathbf{v}_i\}_{i=1}^{d_{\text{state}}} \tag{18}$$

The spectral radius constitutes the most critical stability measure for dynamical systems and is defined as the maximum absolute eigenvalue. Further, system stability requires the spectral radius to satisfy the following constraint.

$$\rho(\mathbf{A}_d) = \max_{1 \le i \le d_{\text{state}}} |\lambda_i| \tag{19}$$

$$\rho(\mathbf{A}_d) \le 1 \tag{20}$$

Values exceeding unity indicate potential exponential divergence in the state dynamics, while values below unity ensure asymptotic stability. This work employs three complementary stability measures to provide a complete characterization of system behavior including, spectral stability quantifies the fraction of matrices satisfying the stability constraint, eigenvalue stability measures the proportion of eigenvalues with non positive real parts, and the condition number assesses numerical conditioning and invertibility respectively.

$$\text{Spectral Stability} = \frac{1}{N}\sum_{k=1}^{N}\mathbb{I}[\rho(\mathbf{A}_d^{(k)}) \leq 1] \tag{21}$$

$$\text{Eigenvalue Stability} = \frac{1}{N \cdot d_{\text{state}}}\sum_{k=1}^{N}\sum_{i=1}^{d_{\text{state}}}\mathbb{I}[\text{Re}(\lambda_i^{(k)}) \leq 0] \tag{22}$$

$$\kappa(\mathbf{A}_d) = \frac{\sigma_{\max}(\mathbf{A}_d)}{\sigma_{\min}(\mathbf{A}_d)} \tag{23}$$

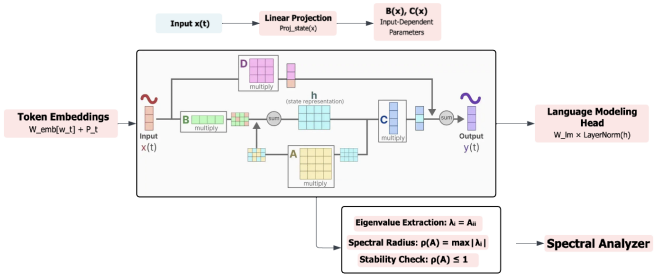

Fig. 1. State space model architecture with spectral modules

Above is the spectral analysis framework of the system. A methodical procedure is followed when collecting spectral data to guarantee computational effectiveness and statistical dependability. To record the temporal evolution of spectral properties, the collection process is carried out at regular intervals during training, namely every 10 epochs. To guarantee representative sampling, the protocol randomly extracts five validation batches for every collection point. In order to generate state matrices under realistic operating conditions, a forward pass computation is also performed where the SSM forward pass occurs. Additionally, state matrices from every layer and time step are gathered.

To ensure robust eigenvalue computation, several safeguards are implemented. A small regularization term is added to improve matrix conditioning. Other additions include NaN detection and real-time gradient norm tracking with automatic termination if gradients become invalid. All linear layers are initialized with $\mathcal{N}(0, 0.01^2)$, while positional embeddings with $\mathcal{N}(0, 0.01^2)$. Bounded activation functions are implemented (tanh, sigmoid) to prevent activation saturation.

The model is trained on an enhanced synthetic dataset consisting of 500 text samples. The smaller size of the dataset was so training dynamics could be isolated and see whether convergence was achievable in such conditions, this being one of the limiting problems with state space models for language. This controlled dataset size allows for comprehensive spectral analysis across all training samples without computational bottlenecks that would arise with larger datasets. The synthetic nature ensures linguistic diversity while maintaining consistent complexity patterns. This enables isolation of the relationship between spectral properties and language modeling performance without confounding variables from natural language irregularities. Each sample is tokenized using a word-level tokenizer with the following special tokens: Padding token, end-of-sequence token, unknown word token, and beginning-of-sequence token.

The model is trained for 100 epochs using the AdamW optimizer with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, initial learning rate $\eta = 5 \times 10^{-4}$, and weight decay $\lambda = 0.01$. The learning rate follows a cosine annealing schedule with $T_{\max} = 100$ epochs and minimum learning rate $\eta_{\min} = 10^{-6}$. Cross-entropy loss with padding token masking is optimized, and gradient clipping with a maximum norm of 1.0 is applied to ensure training stability. Training is conducted with a batch size of 4 sequences per batch.

III. Results

The SSM language model demonstrated exceptional convergence properties across training.

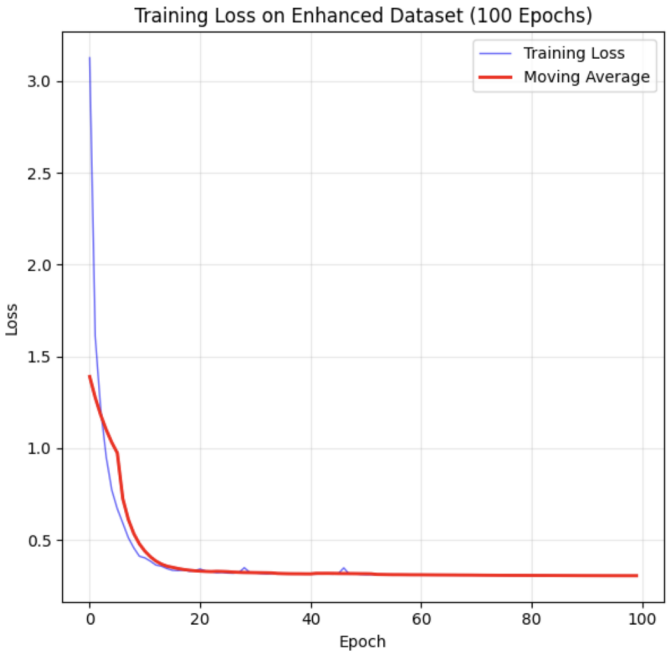

Fig. 2. Loss curve of SSM with moving average

The model improved the training objective by 90.3%, achieving a remarkable loss decrease from 3.127 to 0.305 final loss. While selective state space mechanisms are mathematically elegant, this convergence reveals the SSM architecture's power on language model tasks.

Three distinct phases of training progression were evident in Fig 2: fine-tuning convergence (epochs 50-100), gradual stabilization (epochs 10-50), and rapid initial convergence (epochs 0-10). The loss dropped sharply from 3.127 to 0.402 during the earliest phase, suggesting that the main language modeling patterns were effectively optimized. Loss decreased from 0.402 to 0.311 in the following stabilization phase, indicating a more refined set of intricate linguistic dependencies. With loss plateauing at roughly 0.305 in the final convergence phase, the model successfully converged without overfitting.

The training exhibited stability with 100% valid training steps across all epochs. This demonstrates the robustness of the implemented numerical safeguards and gradient clipping mechanisms. The cosine annealing learning rate schedule effectively guided optimization, beginning at $\eta_{\max} = 5 \times 10^{-4}$ and decreasing to $\eta_{\min} = 1 \times 10^{-6}$ by the final epoch. No training instabilities, gradient explosions, or NaN values were observed throughout the training process either. All being common occurences with SSMs.

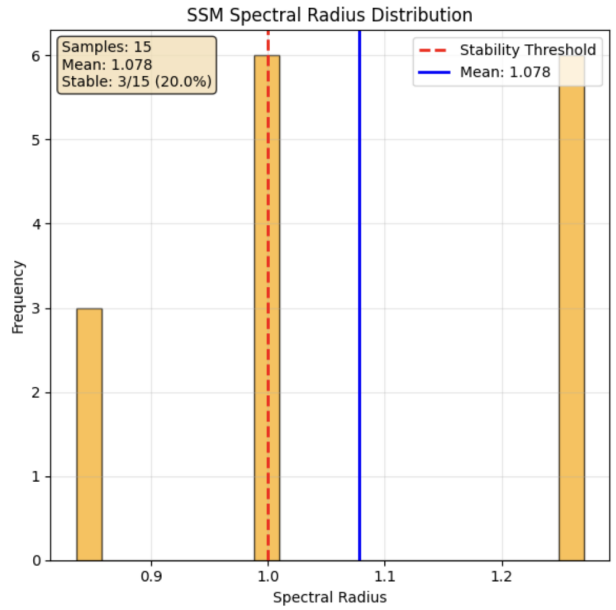

Fig. 3. SSM spectral radius distribution with mean and stability threshhold

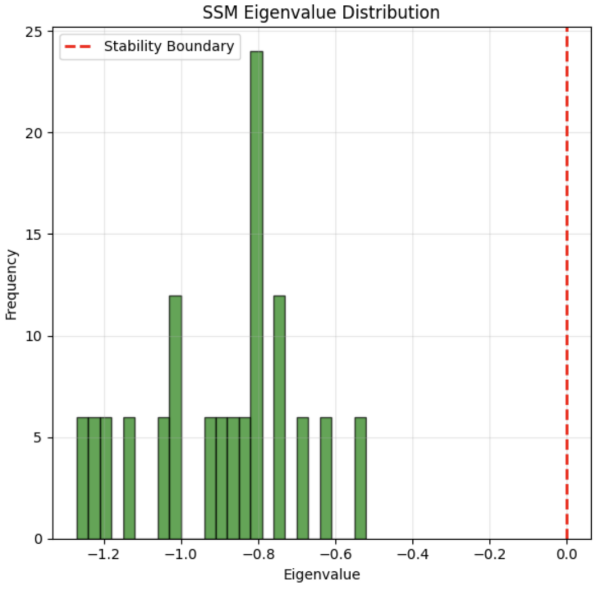

Fig. 4. SSM eigenvalue distribution with stability boundary

The spectral analysis revealed a fundamental paradox in SSM behavior. Despite achieving excellent convergence, only 20% of learned state matrices satisfied the theoretical stability criterion. The mean spectral radius was $\bar{\rho} = 1.078$ with maximum $\rho_{\max} = 1.270$

Eigenvalue analysis showed clustering in the negative real domain $\lambda \in [-1.2, -0.4]$, with concentration around $\lambda \approx -0.8$. While individual eigenvalues maintained stabilizing negative real parts, their collective magnitude through $\rho(\mathbf{A}_d) = \max_{1 \le i \le d_{\text{state}}} |\lambda_i|$ exceeded unity in 80% of matrices. The spectral radius distribution exhibited a bimodal pattern with clusters at $\rho \approx 1.0$ and $\rho \approx 1.2$.

Convergence can still be achieved my SSMs under these conditions. Despite 80% of state matrices violating latex $\rho > 1$, robust training convergence was achieved.

IV. Conclusion

This work gives the first end-to-end spectral analysis of State Space Models in language modeling and reveals a counterintuitive principle of traditional neural network stability that violates sensibility. Although deviating from the theoretical test for stability $\rho(\mathbf{A}_d) > 1$ with mean spectral radius $\bar{\rho} = 1.078$ for 80% of learned state matrices, the 737K parameter SSM achieved optimal convergence, decreasing the loss of training from 3.127 to 0.305 in 100 epochs.

The key finding demonstrates that SSM language models can converge in while having this instability where the selective mechanism $\mathbf{A}_t = f_A(\mathbf{x}_t)$ provides adaptive regularization that prevents mathematical instabilities from causing training divergence. The input-dependent adaptation enables boundedness even when strict dynamical system stability assumptions are broken, which points to inherent stabilization mechanisms of the SSM architecture.

The spectral analysis framework developed here provides rich insights for future SSM research. The resulting bimodal distribution of spectral radii with clustering at $\rho \approx 1.0$ and $\rho \approx 1.2$ demonstrates systematic trends in how SSMs balance expressiveness with stability. The concentration of eigenvalues in the negative real domain $\lambda \in [-1.2, -0.4]$ and collective instability by spectral radius captures the complex mathematical landscape of the training dynamics of SSMs.

These findings have immediate consequences for SSM architecture design. Rather than requiring strict spectral stability, future research should be aimed at understanding and controlling the adaptive properties of the selective mechanism. That successful operation under controlled instability is possible suggests that spectral regularization techniques can be developed to maintain the beneficial characteristics of mild instability without permitting pathological modes. This paper places spectral analysis as a central tool for SSM behavior understanding and sets the stage for developing more principled, stability-aware SSM architectures. The demonstrated ability of SSMs to achieve excellent performance while training outside traditional stability boundaries opens up new avenues for sequence modeling research and challenges inherent assumptions about neural network training dynamics.

Future research must focus on larger scale experiments, investigate the relationship between spectral properties and a

variety of linguistic tasks, and develop training protocols to optimize stability further.

ACKNOWLEDGEMENTS

This manuscript was prepared with the assistance of generative AI tools, specifically OpenAI's ChatGPT-4o, throughout its development. AI was used to refine language clarity, grammar, sentence structure, and overall readability. The AI's role was strictly limited to non-substantive tasks, such as improving linguistic coherence, without altering or contributing to the technical content, analyses, methodologies, or results.

Specifically, AI tools were used to:

Enhance the readability and flow of the Abstract, Introduction, Related Works, Methodology, Results, and Discussion sections. Verify grammar and structure in all technical descriptions and explanations, ensuring accessibility without modifying the underlying scientific meaning. All conceptual, analytical, and experimental work, derivation of equations, design of experiments, implementation of algorithms, and interpretation of results, was conducted solely by the authors.

The authors carefully reviewed and validated all AI-edited content to ensure its alignment with the intended meaning and scientific integrity of the work. No AI was involved in generating technical content, equations, analyses, results, or figures.

REFERENCES

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

[2] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," in Proc. Int. Conf. Learning Representations (ICLR), 2022.

[3] J. T. Halloran, M. Gulati, and P. F. Roysdon, "Mamba state-space models can be strong downstream learners," arXiv preprint arXiv:2406.00209, 2024.

[4] J. T. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," arXiv preprint arXiv:2208.04933, 2022.

[5] H. Mehta, A. Gupta, A. Cutkosky, and B. Neyshabur, "Long range language modeling via gated state spaces," arXiv preprint arXiv:2206.13947, 2022.

[6] M. Chen, J. Pennington, and S. S. Schoenholz, "Dynamical isometry and a mean field theory of RNNs: Gating enables signal propagation in recurrent neural networks," in Proc. Int. Conf. Machine Learning (ICML), vol. 80, 2018, pp. 873–882.

[7] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in Proc. Int. Conf. Machine Learning (ICML), vol. 28, no. 3, 2013, pp. 1310–1318.