TOWARDS EFFICIENT SEQUENCE MODELING: FROM STATE-SPACE MODELS TO MAMBA2

Shicheng Wen
University of Southern California
wenshich@usc.edu

ABSTRACT

Transformers provide powerful long-range reasoning but suffer from quadratic time and memory complexity in sequence length. Structured State-Space Models offer a principled, linear-time alternative derived from control theory. This report develops the mathematical foundations of SSMs, derives their convolutional and transfer-function forms, analyzes stability and expressivity, and examines recent architectures such as *Mamba* and *Mamba2*. Through the lens of Structured State-Space Duality, we show how attention and SSMs are deeply connected via semiseparable matrix structures.

1 INTRODUCTION AND MOTIVATION

Modern sequence models aim to capture long-range dependencies in data such as text, audio, and time series. While the Transformer architecture (Vaswani et al., 2017) has achieved unprecedented success, its self-attention mechanism scales as $O(T^2)$ with sequence length T, both in computation and memory. This quadratic scaling becomes a bottleneck for very long sequences, motivating research into architectures with linear complexity.

Gu et al. (2021) revisit classical linear dynamical systems and reinterpret them as learnable sequence operators, namely Structured State-Space Models (SSMs). By imposing structure on the state transition matrix and exploiting the equivalence between recurrences and convolutions, SSMs achieve $O(T)$ computation with strong theoretical guarantees. Recent work—*Mamba* (Gu & Dao, 2024) and *Mamba2* (Dao & Gu, 2024) —extends this framework with input-dependent dynamics and efficient GPU implementations. This report focuses on their mathematical underpinnings.

2 BACKGROUND: EFFICIENCY LIMITS OF TRANSFORMERS

The Transformer architecture has become the dominant framework for sequence modeling because its self-attention mechanism captures long-range dependencies effectively. However, this ability comes at a steep computational cost. For an input sequence $X = [x_1, \ldots, x_T] \in \mathbb{R}^{T\times d}$, self-attention computes

$$\text{Attention}(X) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

where $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ are the query, key, and value matrices of dimension $T \times d_k$. The pairwise dot product $QK^\top \in \mathbb{R}^{T\times T}$ scales quadratically in T, yielding

$$\text{Time and Memory Complexity: } O(T^2 d_k)$$

Even with efficient batching, this quadratic cost dominates GPU memory and computation for long sequences, making full attention impractical for tasks such as document modeling, video processing, or genomics.

To address this limitation, several approaches approximate or sparsify the attention kernel in equation 1. Sparse Transformers (Child et al., 2019; Beltagy et al., 2020) restrict attention to local windows, while kernelized methods such as Linear Transformers (Katharopoulos et al., 2020) and Performers (Choromanski et al., 2021) approximate

$$\text{softmax}(QK^\top) \approx \phi(Q)\phi(K)^\top$$

using random or deterministic feature maps $\phi(\cdot)$. These methods reduce complexity to $O(Td_k^2)$ but often at the cost of degraded long-range accuracy.

Structured State-Space Models (SSMs) approach efficiency from a different angle. Instead of pairwise token interactions, an SSM defines a linear recurrence

$$h_{t+1} = \bar{A}h_t + \bar{B}x_t, \qquad y_t = Ch_t \tag{2}$$

which computes the same sequence operator through matrix products with complexity $O(TN)$ for state dimension $N \ll T$. The convolutional equivalence

$$y_t = \sum_{i=0}^{t} C\bar{A}^i\bar{B}x_{t-i}$$

allows parallelization via FFTs or selective scans, providing strictly linear scaling in T. Thus, while attention computes all pairwise interactions explicitly, SSMs model sequence evolution implicitly through stable linear dynamics, achieving similar expressivity with much lower computational cost.

3 Linear State-Space Systems

3.1 Continuous-Time Formulation

An SSM describes how a latent state $h(t) \in \mathbb{R}^N$ evolves under a driving signal $x(t) \in \mathbb{R}^d$:

$$\frac{d}{dt}h(t) = Ah(t) + Bx(t), \qquad y(t) = Ch(t) \tag{3}$$

where $A \in \mathbb{R}^{N\times N}$ governs the system dynamics, $B \in \mathbb{R}^{N\times d}$ maps inputs to state changes, and $C \in \mathbb{R}^{d\times N}$ projects the state to an observable output. Equation equation 3 defines a continuous-time linear time-invariant (LTI) system.

3.2 Discretization

Neural implementations require discrete updates. For a sampling interval Δt, the exact zero-order-hold (ZOH) discretization gives:

$$\bar{A} = e^{\Delta t A} \tag{4}$$

$$\bar{B} = \int_0^{\Delta t} e^{(\Delta t-\tau)A}B\,d\tau = A^{-1}(e^{\Delta tA} - I)B \tag{5}$$

The discrete system becomes:

$$h_{t+1} = \bar{A}h_t + \bar{B}x_t, \qquad y_t = Ch_t \tag{6}$$

The spectral radius $\rho(\bar{A}) < 1$ ensures BIBO stability; in continuous time this corresponds to A being Hurwitz ($\mathrm{Re}\,\lambda_i(A) < 0$).

3.3 Convolutional and Transfer-Function View

Unrolling equation 6 yields:

$$h_t = \sum_{i=0}^{t-1} \bar{A}^i\bar{B}x_{t-1-i} \tag{7}$$

$$y_t = Ch_t = \sum_{i=0}^{t-1} C\bar{A}^i\bar{B}x_{t-1-i} \tag{8}$$

Defining the kernel coefficients $K_i = C\bar{A}^i\bar{B}$, we obtain

$$y_t = (K * x)_t = \sum_{i=0}^{t} K_i x_{t-i} \tag{9}$$

Thus an SSM is equivalent to a one-dimensional convolutional operator. In the z-domain, the transfer function is:

$$H(z) = C(I - z^{-1}\bar{A})^{-1}\bar{B} = \sum_{i \geq 0} K_i z^{-i} \tag{10}$$

This dual representation (recurrence $\leftrightarrow$ convolution) enables either sequential or fully parallel computation.

4 Expressivity and Stability

The eigenstructure of $\bar{A}$ determines the model's memory and frequency response. If $\bar{A}$ is diagonalizable, $\bar{A} = V\Lambda V^{-1}$, then

$$K_i = CV\Lambda^i V^{-1}\bar{B}$$

so K_i decays geometrically with $|\lambda_j|$. Small $|\lambda_j|$ produce short memory; values close to 1 yield long-range dependencies. Structured parameterizations (diagonal or diagonal-plus-low-rank) allow efficient kernel computation while preserving representational diversity.

Stability and gradient propagation are closely linked: enforcing $\rho(\bar{A}) < 1$ ensures bounded hidden states and prevents exploding gradients. Some implementations reparameterize $\bar{A} = \mathrm{Diag}(\exp(-\Delta t\, \tau))$ with learnable positive τ to maintain stability by construction.

5 Selective State-Space Models: Mamba

Static SSMs use fixed matrices A, B, and C for all time steps. While efficient, they cannot modulate behavior based on content. The *Mamba* architecture introduces *selectivity*, making parameters depend on the input sequence, allowing the system to adapt dynamically while maintaining linear complexity.

5.1 Input-Dependent Dynamics

Mamba defines:

$$h_{t+1} = A(x_t)h_t + B(x_t)x_t, \qquad y_t = C(x_t)h_t \tag{11}$$

The matrices are modulated by the current input x_t through lightweight linear projections and gating functions. Expanding the recurrence gives an input-dependent kernel:

$$K_i(x) = C(x_t)A(x_{t-1}) \cdots A(x_{t-i+1})B(x_{t-i}) \tag{12}$$

Consequently,

$$y_t = \sum_{i=0}^{t} K_i(x)x_{t-i} \tag{13}$$

which generalizes the static convolution by allowing the kernel to change adaptively with context.

5.2 Selective Scan Algorithm

Naively computing equation 12 is sequential. Mamba circumvents this with a *selective scan*—a parallel prefix algorithm that efficiently accumulates products of the dynamic transition matrices. This design provides $O(T)$ training and inference time, $O(T)$ memory, and constant per-step cost in autoregressive generation. It fuses recurrence and gating into a single efficient CUDA kernel, improving throughput by up to $5\times$ compared with equivalent Transformers.

6 Mamba2: Structured State-Space Duality

6.1 Semiseparable Matrix Representation

Mamba2 formalizes a theoretical equivalence between SSMs and attention using *Structured State-Space Duality*. Define a lower-triangular matrix M whose elements are:

$$M_{j,i} = C_j^\top A_j A_{j-1} \cdots A_{i+1} B_i, \qquad j \geq i \tag{14}$$

M is a *semiseparable* matrix of rank N, meaning each off-diagonal block has rank at most N. Matrix multiplication $Y = MX$ reproduces the forward recurrence of an SSM, while its explicit form resembles attention:

$$Y = (L \circ QK^\top)V \tag{15}$$

where L encodes the time-dependent propagation factors and $\circ$ denotes the Hadamard product. Hence, SSMs and attention are dual algorithmic perspectives on the same underlying structured operator.

6.2 Algorithmic and Practical Improvements

Mamba2 reparameterizes and fuses selective SSM operations into efficient linear layers. Key contributions include:

- GPU-friendly block matrix multiplication using semiseparable decomposition;
- multi-input, multi-value extensions analogous to multi-head attention;
- higher memory throughput and kernel fusion for selective scans;
- improved theoretical stability by constraining $\rho(A(x_t)) < 1$ for all t.

Empirically, Mamba2 outperforms equivalently sized Transformers and Mamba models on long-context benchmarks such as The Pile and Long Range Arena while using less memory.

7 Complexity and Theoretical Comparisons

The following table summarizes the asymptotic cost per layer (ignoring constants):

Model	Training Time	Memory	Autoregressive Step
Transformer (Self-Attention)	$O(T^2)$	$O(T^2)$	$O(T)$
LTI SSM (Static)	$O(T)$ or $O(T \log T)$	$O(T)$	$O(1)$
Mamba (Selective)	$O(T)$	$O(T)$	$O(1)$
Mamba2 (SSD)	$O(T)$ (faster constant)	$O(T)$	$O(1)$

SSMs thus combine the expressivity of recurrent systems with the parallelism of convolution and outperform Transformers on extremely long sequences.

8 Open Problems

1. **Expressive Completeness:** Under what conditions can selective SSMs approximate arbitrary attention mappings?
2. **Stability–Memory Tradeoff:** How does constraining $\rho(A(x_t)) < 1$ limit the effective receptive field or gradient propagation depth?
3. **Hybrid Architectures:** What is the optimal integration between sparse attention and selective SSM layers?
4. **Continuous-Time Extensions:** Can neural ODE/SDE formulations yield continuous-time generalizations of selective SSMs with provable invariances?

9 Conclusion

Structured State-Space Models provide a mathematically elegant and computationally efficient framework for long-sequence modeling. By recasting recurrence as convolution and enforcing stability through spectral constraints, they achieve linear-time inference while maintaining expressivity. Mamba introduces input-dependent selectivity, and Mamba2 unifies SSMs and attention through structured state-space duality, achieving both theoretical clarity and practical efficiency. Together, these advances redefine our understanding of sequence modeling as learning structured linear operators governed by dynamical systems principles.

REFERENCES

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. In *arXiv preprint arXiv:2004.05150*, 2020.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. In *arXiv preprint arXiv:1904.10509*, 2019.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Andreea Anghel, Tamás Sarlos, and Adrian Weller. Rethinking attention with performers. In *ICLR*, 2021.

Tri Dao and Albert Gu. Transformers are ssms: generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2021.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

USE OF GENERATIVE AI

ChatGPT is used for deriving the mathematical equations, understanding the concepts (prompts like "help me to explain the mathematical equations in Mamba 2") and refining the writing style in the report.