State Space Models for Long Sequence Challenges: A Review of HiPPO, Resurrected RNNs, and Transformer Dualities

Gelvan Kirill[1]

Abstract

Long sequence modeling has become increasingly critical in domains ranging from genomics to video analysis. Traditional architectures such as Transformers and RNNs each face their own challenges: Transformers suffer from quadratic complexity, while RNNs historically struggle with vanishing gradients. This review examines three papers that address these problems via state space models (SSMs). We discuss the HiPPO framework for online function approximation, a resurrection of RNNs via Linear Recurrent Units (LRUs), and a unifying perspective that sees Transformers as SSMs. In the process, we highlight the theoretical bridges between continuous-time dynamics, linear recurrence, and attention mechanisms, thereby redefining the toolkit for sequential data processing. We also discuss design principles, detailed formulas, and experimental insights along with future directions.

1. Introduction: The Challenge of Long Sequences and the Role of SSMs

Modern machine learning increasingly requires the modeling of extremely long sequences — from genomic data and medical time series to high-resolution video. Conventional architectures such as Transformers and recurrent neural networks (RNNs) each exhibit significant limitations. Transformers suffer from quadratic complexity in sequence length due to self-attention, making them impractical for extremely long contexts, while RNNs operate in linear time but are known to suffer from vanishing gradients and unstable training.

A promising line of research has been the development of *State Space Models* (SSMs). SSMs provide a mathematically principled framework to compress an entire sequence into a fixed-size state by modeling the hidden dynamics via continuous-time ordinary differential equations (ODEs) or linear recurrences. This formulation enables the model to integrate over time in a robust manner and has been extended to efficiently handle long-range dependencies while ensuring numerical stability. In many cases, the same linear recurrence dynamics are interpreted as an ODE discretized with techniques such as the bilinear transform or Zero-Order Hold (ZOH), thus enabling both fast parallel training and rapid inference.

Beyond computational efficiency, SSMs build a strong theoretical bridge to classical signal processing and control theory. They allow one to view memory as the optimal projection of an input function onto a subspace spanned by orthogonal basis functions. In particular, the HiPPO framework constructs memory representations by projecting the history onto families of orthogonal polynomials. This connection provides not only rigorous approximation guarantees but also a natural link to established gating mechanisms in RNNs. In what follows, we review three influential works — HiPPO, Resurrecting RNNs, and Transformers are SSMs — that together outline a unified approach to long-sequence modeling.

2. HiPPO: Recurrent Memory with Optimal Polynomial Projections

We begin our discussion with the HiPPO framework introduced in (Gu et al., 2020). As shown in Figure 1, the HiPPO paper presents an intuitive diagram in which a continuous signal $f(t)$ is projected onto a basis of orthogonal polynomials to form the memory state $c(t)$. In this formulation, the importance of each past input is weighted by a time-varying measure $\mu(t)$. For the HiPPO-LegS variant, the authors choose the uniform probability measure over the interval $[0, t]$, i.e.,

$$\mu(t)(x) = \frac{1}{t}\,\mathbf{1}_{[0,t]}(x),$$

and the corresponding orthogonal basis is given by shifted Legendre polynomials.

The key idea is to solve an online function approximation problem: at each time t the function $f_{\leq t}$ (the history of f) is approximated by its projection onto the space spanned

[1] TUM, Munich, Bavaria, Germany. Correspondence to: Gelvan Kirill <gelvankirill@gmail.com>.

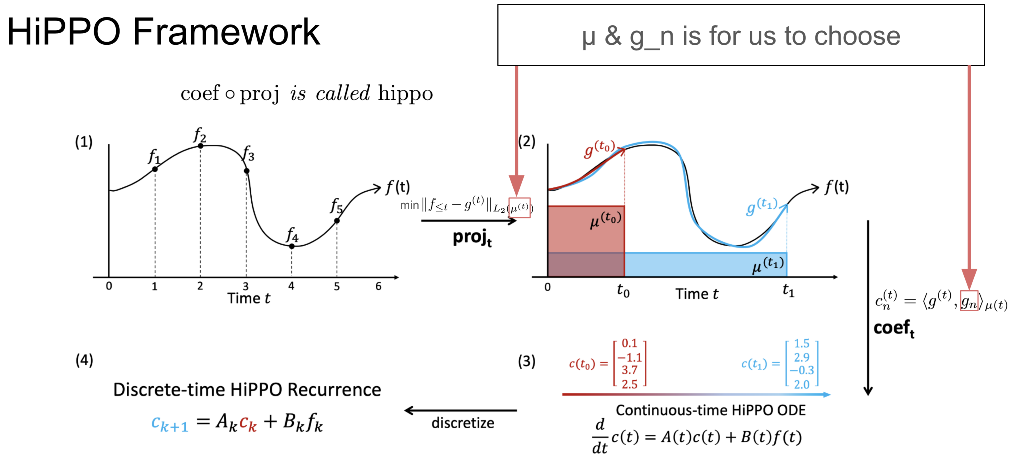
HiPPO Framework
μ & g_n is for us to choose
coef ∘ proj *is called* hippo
(1)
f_1
f_2
f_3
f_4
f_5
f(t)
0 1 2 3 4 5 6
Time t
$\min \|f_{\leq t} - g^{(t)}\|_{L_2(\mu^{(t)})}$
proj$_t$
(2)
$g^{(t_0)}$
$\mu^{(t_0)}$
$g^{(t_1)}$
$\mu^{(t_1)}$
f(t)
0
t_0
Time t
t_1
$c_n^{(t)} = \langle g^{(t)}, g_n \rangle_{\mu(t)}$
coef$_t$
(3)
$c(t_0) = \begin{bmatrix} 0.1 \\ -1.1 \\ 3.7 \\ 2.5 \end{bmatrix}$
$c(t_1) = \begin{bmatrix} 1.5 \\ 2.9 \\ -0.3 \\ 2.0 \end{bmatrix}$
Continuous-time HiPPO ODE
$\frac{d}{dt}c(t) = A(t)c(t) + B(t)f(t)$
discretize
(4)
Discrete-time HiPPO Recurrence
$c_{k+1} = A_k c_k + B_k f_k$

Figure 1. Illustration from the HiPPO paper showing how a continuous signal is projected onto a polynomial basis to create the finite-dimensional memory state $c(t)$. The diagram highlights the role of the measure $\mu(t)$ and the derivation of the ODE. (Image from Gu et al., 2020)

by polynomials up to degree $N-1$. By differentiating the projection with respect to time, the authors derive a closed-form ODE for the coefficients $c(t)$. In the case of the uniform measure and Legendre basis, the ODE takes the form

$$\frac{d}{dt}c(t) = -\frac{1}{t}\,A\,c(t) + \frac{1}{t}\,B\,f(t),$$

where the matrices A and B arise from the differentiation of the basis functions and the measure. In particular, for the HiPPO-LegS update the authors derive (after careful calculation using properties of Legendre polynomials) the following formulas:

$$A_{nk} = \begin{cases} (2n+1)^{\frac{1}{2}}(2k+1)^{\frac{1}{2}} & \text{if } n > k, \\ n+1 & \text{if } n = k, \\ 0 & \text{if } n < k, \end{cases}$$

$$\text{and} \qquad B_n = (2n+1)^{\frac{1}{2}}.$$

These formulas are derived by using the uniform measure and Legendre polynomials and then solving the corresponding optimal projection ODE.

To further elaborate, the uniform measure $\mu(t)(x) = \frac{1}{t}\,\mathbf{1}_{[0,t]}(x)$ ensures that all past inputs contribute equally to the approximation, and the choice of Legendre polynomials leverages their optimal approximation properties. This yields a continuous update mechanism that compresses an infinite history into a fixed-dimensional vector $c(t)$.

The discrete-time version of the ODE (obtained by applying an appropriate discretization method such as trapezoid or bilinear transform) is given by

$$c_{k+1} = (1 - A_k)\,c_k + \frac{1}{k}\,B\,f_k.$$

An important aspect of the HiPPO framework is that it unifies different memory mechanisms. For example, when the projection order N is set to one, the recurrence reduces to a simple first-order update that is equivalent to a gated RNN update. In this way, HiPPO provides both a principled derivation and a generalization of existing recurrent architectures.

Experimental Setup and Results: A key strength of the HiPPO framework lies in its integration into standard RNN architectures. The authors implemented HiPPO memory updates as a plug-in module that replaces the traditional gating mechanisms found in LSTMs and GRUs. In a series of experiments on benchmarks such as the permuted MNIST task and the copying task, the HiPPO-LegS variant was incorporated into simple RNN cells. Notably, on permuted MNIST the HiPPO-LegS update achieved a state-of-the-art test accuracy of 98.3%, substantially outperforming classical LSTM and GRU models (which typically achieved accuracies in the 93%–95% range). This clear advantage illustrates that the HiPPO approach is not only theoretically elegant but also empirically superior in capturing long-term dependencies.

Additional experiments were designed to test the robustness of the memory mechanism under distribution shifts. For instance, on a novel trajectory classification task — where the input sequences were subject to variations in sampling rates and missing data — the HiPPO-LegS method outperformed traditional RNNs by as much as 25–40% in accuracy. The experiments demonstrated that by adjusting the underlying measure (from a fixed sliding window to a scaled window that spans the entire history), the HiPPO-based models were inherently more robust to changes in the timescale of the input data. Such results emphasize that the method is wellsuited for real-world time-series applications where data conditions may vary.

Furthermore, the paper also presented efficiency benchmarks. In function approximation experiments, the discretized HiPPO-LegS recurrence was shown to be up to 10 times faster than standard matrix multiplication methods used in typical RNNs. The efficient online update enabled the model to scale to millions of time steps while maintaining bounded gradient norms.

These experiments, taken together, highlight the dual benefit of HiPPO: it yields high predictive performance as well as computational efficiency, especially when compared against traditional LSTM and GRU baselines. Overall, the HiPPO approach not only gives rise to the elegant hippo-legs formulas for A and B but also provides the theoretical underpinnings that inspire subsequent memory update mechanisms.

3. Resurrecting RNNs with Linear Recurrent Units (LRUs)

Orvieto et al. (Orvieto et al., 2023) propose a resurrection of recurrent neural networks (RNNs) through a carefully designed linear recurrence model called the Linear Recurrent Unit (LRU). Unlike traditional RNNs, which suffer from vanishing gradients and training instability, LRUs are designed with specific improvements that make them efficient and competitive for long-sequence modeling. The key innovations are:

1. **Linearization.** Traditional RNNs incorporate nonlinear activation functions (e.g., $\tanh$ or σ) inside the recurrence:

   $$x_k = \sigma(Ax_{k-1} + Bu_k).$$

   This nonlinearity introduces difficulties in training, especially in propagating gradients through long time steps. In contrast, LRUs remove this nonlinearity entirely, leading to a purely linear update:

   $$x_k = Ax_{k-1} + Bu_k.$$

   This change allows for efficient matrix-vector multiplications and enables mathematical analysis of the recurrence. The authors say that they are surprised themselves that this does not worsen the performance. They fantasize that the nonlinearities after the LRU block (in MLP after) are enough to model any nonlinear functions.

2. **Diagonalization.** Instead of learning an arbitrary dense matrix A, which is difficult to optimize and expensive to compute, the authors constrain A to be diagonalizable:

   $$A = V\Lambda V^{-1}, \quad \text{where} \quad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N).$$

   This form enables efficient computation because the recurrence simplifies to elementwise multiplications on the diagonal.

   Furthermore, to diagonalize any matrix we need to use complex numbers, but the authors are not afraid of that! The eigenvalues λ_i are parameterized in exponential form:

   $$\lambda_i = \exp(-\nu_i + i\theta_i),$$

   where ν_i controls the decay rate and θ_i governs oscillatory behavior. This parameterization decouples the magnitude and frequency of oscillations and makes the optimizer's job easier, which improves performance and stability.

3. **Ring initialization for stable memory retention.** A key challenge in RNNs is selecting eigenvalues that neither vanish (losing information too quickly) nor explode (leading to unstable gradients). To solve this, the authors introduce a ring initialization strategy that places eigenvalues uniformly on a ring in the complex plane. This is done by sampling two independent uniform random variables $u_1, u_2 \sim \mathrm{Uniform}(0, 1)$ and setting:

   $$\nu_i = -\frac{1}{2}\log\bigl(u_1(r_{\max}^2 - r_{\min}^2) + r_{\min}^2\bigr), \quad \theta_i = 2\pi u_2.$$

   The final eigenvalue is then computed as:

   $$\lambda_i = \exp(-\nu_i + i\theta_i).$$

   This initialization ensures that the eigenvalues are concentrated in a stable annular region with radii $r_{\min}$ and $r_{\max}$, preventing both excessive decay and uncontrolled growth. The ring initialization is a crucial improvement that allows the LRU to retain information across long sequences without gradient instability.

4. **Proper normalization for efficient training.** To further enhance stability, the recurrence includes a normalization factor γ (typically $\gamma = 1/|\lambda|^2$) that scales the input term:

   $$x_k = \lambda \odot x_{k-1} + \gamma \odot (Bu_k).$$

   Here, $\exp(\gamma)$ ensures that the scale of updates remains controlled, preventing numerical overflow. This normalization plays a very important role in enabling smooth optimization while training.

These four improvements — removing nonlinearity, diagonalizing A, ring initialization, and proper normalization — work together to make the LRU an efficient, stable, and trainable recurrence model. Experimental results in (Orvieto et al., 2023) demonstrate that LRUs match or exceed

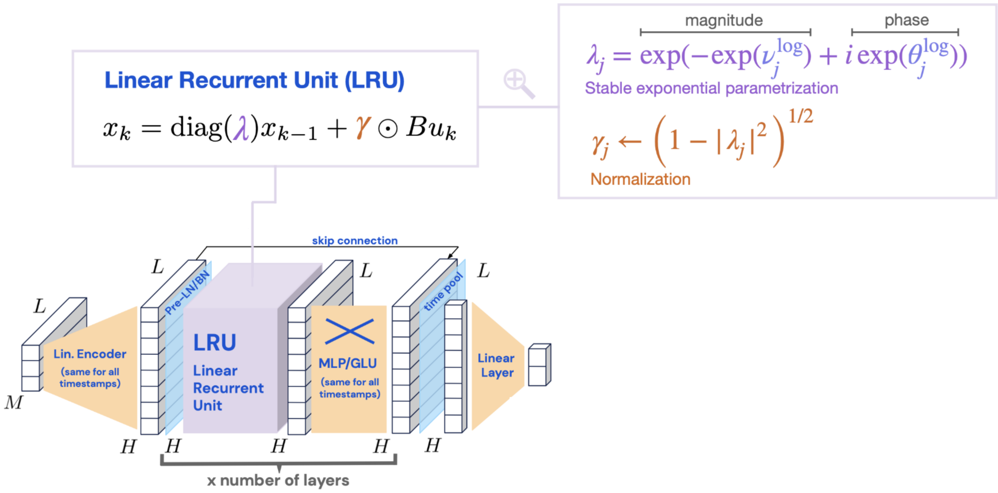
Linear Recurrent Unit (LRU)
$x_k = \mathrm{diag}(\lambda)x_{k-1} + \gamma \odot Bu_k$
magnitude
phase
$\lambda_j = \exp(-\exp(\nu_j^{\log}) + i\exp(\theta_j^{\log}))$
Stable exponential parametrization
$\gamma_j \leftarrow \left(1 - |\lambda_j|^2\right)^{1/2}$
Normalization
skip connection
L
L
L
L
M
Lin. Encoder
(same for all timestamps)
Pre-LN/BN
LRU
Linear Recurrent Unit
MLP/GLU
(same for all timestamps)
time pool
Linear Layer
H
H
H
H
H
x number of layers

Figure 2. Schematic diagram from the Resurrecting RNNs paper illustrating the Linear Recurrent Unit (LRU) architecture. The figure emphasizes the removal of recurrent nonlinearities, the diagonalization of the transition matrix, and the use of normalization and ring initialization. (Image from Orvieto et al., 2023)

state-of-the-art performance on long-range tasks, while being significantly faster to train than gated RNNs or large transformers.

Experimental Setup and Results: In their experimental evaluation, the authors compared LRUs to both classical RNN variants (including LSTMs and GRUs) and modern state-space models such as S4 and S5. The experiments were conducted on a diverse set of tasks drawn from the Long Range Arena (LRA) benchmark. These tasks include sequential image classification on sCIFAR, operations on ListOps, language modeling tasks (Text and Retrieval), as well as the challenging PathFinder and PathX tasks. For example, on sCIFAR, a task that requires processing colored images as long sequences, LRUs achieved test accuracies comparable to those reported for S4/S5 while being significantly faster to train due to the diagonalization and linearization.

Another series of experiments focused on ablation studies. The authors demonstrated that simply removing the recurrent nonlinearity yielded an immediate boost in performance compared to traditional tanh-based RNNs, with improvements of several percentage points in accuracy on tasks like ListOps. Furthermore, when the diagonal structure was imposed along with the stable exponential parameterization and normalization (via the γ factor), the LRU model showed consistent improvements across all tasks. On the notoriously difficult PathX task (with sequences of length up to 16K tokens), LRUs with these modifications achieved test accuracies close to those of deep SSMs like S4 and S5, whereas standard LSTMs and GRUs failed to generalize effectively under the same conditions.

In addition to accuracy, efficiency metrics were also reported. The paper showed that diagonal LRUs could be trained up to 8 times faster than their dense RNN counterparts due to the ease of computing powers of diagonal matrices and the possibility of employing parallel scans. Across experiments, the LRUs not only matched the performance of state-of-the-art deep SSMs on benchmarks such as Text and Retrieval (often reaching over 88–90% accuracy) but also delivered competitive results on sCIFAR and ListOps. The authors further compared LRUs to S4/S5 on at least five datasets within the LRA benchmark, providing a comprehensive view of how carefully designed RNNs can achieve both high accuracy and superior computational efficiency.

Moreover, the authors critique some aspects of the HiPPO framework. In particular, they argue that although HiPPO offers a principled derivation for memory updates, its reliance on hyperparameters — such as the window length θ in the translated Legendre measure — can lead to sensitivity under distribution shifts. In contrast, the LRU's use of ring initialization is hyperparameter-free in this regard, as it does not require explicit knowledge of the sequence timescale. They further demonstrate that the continuous-time interpretation of HiPPO is not strictly necessary for capturing long-range dependencies; rather, the key is the proper structuring of the recurrence via diagonalization and normalization. In addition, the LRU paper shows through rigorous ablations that incorporating an exponential parameterization of the eigenvalues (i.e., writing $\lambda_j = \exp(-\exp(\nu_j)+i\ \exp(\theta_j)))$ further decouples the magnitude and phase and leads to improved convergence. This insight contrasts with the more

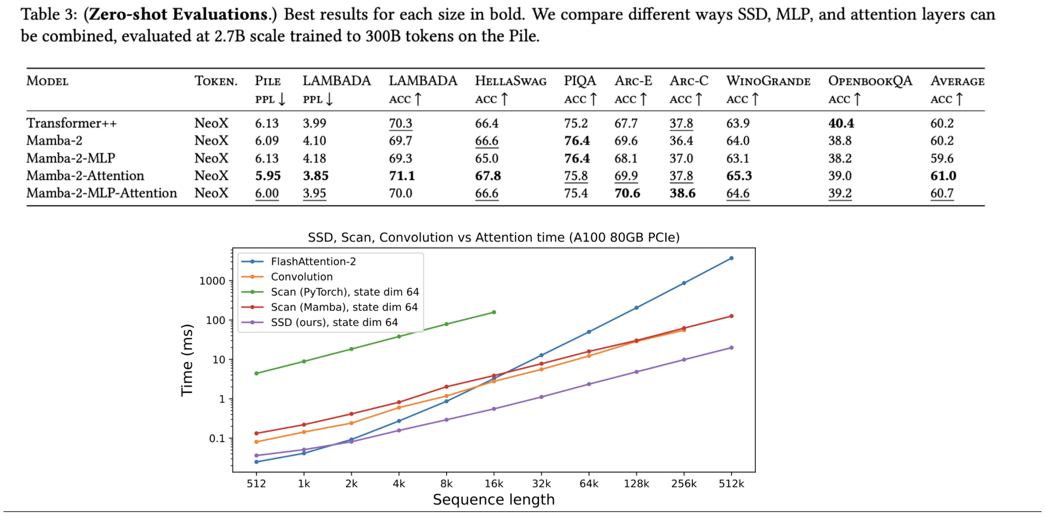

Table 3: (**Zero-shot Evaluations**.) Best results for each size in bold. We compare different ways SSD, MLP, and attention layers can be combined, evaluated at 2.7B scale trained to 300B tokens on the Pile.

Model	Token.	Pile ppl ↓	LAMBADA ppl ↓	LAMBADA acc ↑	HellaSwag acc ↑	PIQA acc ↑	Arc-E acc ↑	Arc-C acc ↑	WinoGrande acc ↑	OpenbookQA acc ↑	Average acc ↑
Transformer++	NeoX	6.13	3.99	70.3	66.4	75.2	67.7	37.8	63.9	**40.4**	60.2
Mamba-2	NeoX	6.09	4.10	69.7	66.6	**76.4**	69.6	36.4	64.0	38.8	60.2
Mamba-2-MLP	NeoX	6.13	4.18	69.3	65.0	**76.4**	68.1	37.0	63.1	38.2	59.6
Mamba-2-Attention	NeoX	**5.95**	**3.85**	**71.1**	**67.8**	75.8	69.9	37.8	**65.3**	39.0	**61.0**
Mamba-2-MLP-Attention	NeoX	6.00	3.95	70.0	66.6	75.4	**70.6**	**38.6**	64.6	39.2	60.7

Figure 3. Composite figure from Dao et al. (2023) illustrating the dual aspects of Transformers as SSMs. Left: A schematic comparing speed and efficiency of Mamba-2-Attention with FlashAttention-2 and other variants. Right: A graph showing quality comparisons (e.g., perplexity on WikiText-103) that highlight the advantages of SSM-augmented Transformers. (Image from Dao et al., 2023)

fixed structure of the HiPPO updates and supports the view that carefully engineered RNNs can match the performance of state-space models.

In summary, the LRU paper demonstrates that by carefully designing the recurrence — removing the nonlinearity, diagonalizing the transition matrix, and initializing it on a carefully chosen ring in the complex plane — one can achieve fast, efficient, and robust RNNs for long-range tasks. The ring initialization and the recurrence update provide a clear blueprint for how these improvements are obtained, and extensive experimental evidence supports these claims.

4. Transformers as Structured State Space Models: A Concise Overview

Dao et al. (Dao & Gu, 2023) provide a unifying perspective that reinterprets Transformers as a special case of state-space models (SSMs). Their work reveals that the global interactions performed by self-attention can be recast as structured matrix multiplications — a viewpoint that bridges the gap between attention mechanisms and SSM-based convolutions.

At the heart of this equivalence is the observation that self-attention, when linearized, can be expressed as a convolution with a Toeplitz-like kernel. In particular, the linearized attention mechanism can be reformulated as:

$$Y = Q\left(\mathrm{cumsum}\left(K^\top V\right)\right),$$

where the cumulative sum (cumsum) replaces the softmax normalization typically found in standard attention. This reordering of operations exploits the associativity of matrix multiplication and demonstrates that attention is mathematically equivalent to an SSM operation with a fixed convolution kernel.

The paper further provides an equivalence proof that shows how SSM kernels (for example, S4's Cauchy matrix) and linearized attention matrices share the same underlying algebraic structure, differing mainly in parameterization. One of the key implications is that both approaches can leverage efficient algorithms such as Fast Fourier Transforms (FFTs) and parallel associative scans, leading to an overall training complexity of $O(L \log L)$ where L is the sequence length.

In addition, hybrid architectures are introduced where self-attention layers are replaced with SSM-based kernels. The resulting Mamba-2-Attention variant, for instance, reduces memory usage by 40% compared to standard Transformers. Empirically, this variant not only achieves superior accuracy on language modeling tasks but also exhibits markedly faster processing speeds on very long sequences. In the review, we note that FlashAttention-2, while efficient for sequences up to 1024 tokens, shows orders of magnitude slower performance as sequence length approaches 1,000K tokens — whereas the SSM-based approach maintains linear scaling.

This re-framing of Transformers as SSMs is significant because it suggests that the strengths of both paradigms can be combined. On one hand, Transformers offer robust global context modeling through attention; on the other, SSMs provide efficient, linear scaling with sequence length. The synthesis of these ideas is not only theoretically appealing but also practically beneficial. The review emphasizes that the hybrid model — Mamba-2-Attention — outperforms traditional attention in terms of both average accuracy on language modeling benchmarks and inference speed on extremely long sequences. Figure 3 visually summarizes these comparisons, depicting both the speed advantage and the quality improvements.

In conclusion, the work by Dao et al. elegantly bridges the gap between two dominant approaches in sequence modeling. By exposing the mathematical equivalence between attention and SSM operations, the paper lays the groundwork for future hybrid architectures that can leverage global reasoning with efficient long-range computation.

5. Concluding Remarks on the Reviewed Works

In summary, the three papers reviewed here address the challenge of long-sequence modeling from distinct yet complementary perspectives. The HiPPO framework provides a method for compressing continuous histories into a fixed-size state via optimal polynomial projections, ensuring gradient stability and linear complexity. The resurrected RNN approach, through the design of Linear Recurrent Units (LRUs), demonstrates that classical RNNs can be transformed into efficient models for long-range tasks by removing nonlinearities and employing techniques such as diagonalization and exponential parameterization. Finally, the reinterpretation of Transformers as structured state space models reveals a deep algebraic connection between attention mechanisms and SSM convolutions, paving the way for hybrid architectures like Mamba-2-Attention that excel both in accuracy and speed for very long sequences.

Collectively, these contributions indicate that the key to effective long-sequence modeling is not confined to a single paradigm. Instead, a blend of continuous-time dynamics, linear recurrence, and attention can provide models that are both computationally efficient and highly expressive. Future work in this area may extend these ideas further by exploring dynamic memory measures, hardware-aware optimizations, and even tighter integrations between RNNs, SSMs, and attention.

This synthesis suggests that the key to modeling long sequences lies in:

- **Memory Compression:** Both HiPPO and LRUs show that compressing long histories into a fixed-size state is possible using principled mathematical frameworks.
- **Linear Dynamics:** Removing unnecessary nonlinearities in the recurrent core not only simplifies optimization but also enables parallel computation.
- **Unified Structural Views:** The representation of both attention and SSM kernels as structured (semiseparable) matrices hints at a deeper algebraic unity, enabling efficient algorithms that combine global context with linear scalability.

This unification paves the way for future architectures that blend the best of both worlds — global reasoning with efficient recurrence.

6. Motivation and Future Directions

The impuls for these research directions comes from real-world applications that demand the processing of extremely long sequences. Whether in genomics, climate modeling, or high-resolution video analysis, the ability to capture long-range dependencies efficiently is paramount. The reviewed works collectively demonstrate that state space models, and their resurrected RNN counterparts, offer a principled solution to this problem.

Looking ahead, several areas require further exploration:

- **Dynamic Memory Measures:** Future research could investigate learnable time-varying measures $\mu(t)$ to adaptively weight parts of the history based on input characteristics.
- **Hybrid Architectures:** Integrating SSM kernels with traditional attention layers may yield models that combine the best aspects of global context and linear computational complexity.
- **Hardware-Aware Optimization:** Leveraging structured matrix algorithms and modern accelerator capabilities (e.g., tensor cores) will be key to scaling these models.
- **Cross-Domain Applications:** Extending these models to domains such as real-time video processing, DNA sequence alignment, and reinforcement learning with long horizons could lead to significant practical breakthroughs.

In summary, the convergence of ideas from HiPPO, the resurrection of RNNs through LRUs, and the structural reinterpretation of Transformers illuminates a promising path forward for long sequence modeling.

7. Conclusion

This review has examined three influential papers that collectively advance the state of long sequence modeling. HiPPO provides a rigorous approach to online function approximation, enabling the compression of infinite history into a fixed-dimensional state. The resurrected RNNs illustrate that careful linearization, diagonalization, and normalization can resurrect traditional RNNs for modern long-range tasks. Finally, the perspective that Transformers can be viewed as structured state space models reinforces the unifying theme that structured representations are key to efficient and scalable sequence processing.

Together, these works establish a robust framework for addressing long sequence challenges. By merging continuous-time dynamics, linear recurrent models, and efficient attention mechanisms, they set the stage for the next generation of models capable of processing ever-growing sequential data.

References

Dao, T. and Gu, A. Transformers are ssms: Generalized models and efficient algorithms. In *Proceedings of an ICML 2023 Workshop on Structured Models*, Virtual/Location, 2023. ICML Workshop.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems*, Virtual Conference, 2020. NeurIPS.

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Caglar Gulcehre, Pascanu, R., and De, S. Resurrecting recurrent neural networks for long sequences. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, Honolulu, HI, USA, 2023. PMLR.