# MIRA: MITIGATING POLE GEOMETRY BIAS IN RECURRENT LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We show that the Transformer's recurrent successors — such as selective state-space models (SSMs), Receptance Weighted Key Value (RWKV), and Retentive Network (RetNet) — can be characterized as infinite-impulse-response (IIR) systems. The zero–pole geometries of these IIR systems characterize the temporal modes they can represent. However, to trade parallelism, the algebraic structures of the parameterizations of these recurrent systems often give rise to poles constrained to the real axis, thereby inducing an architectural bias in their recurrent dynamics. To unify these recurrent architectures and eliminate their pole geometry bias, inspired by the lattice–ladder realization of general IIR systems, we present MIRA (Modal Adaptive Recurrent Lattice), a unified family of linear-time recurrent architectures for building large language models (LLMs) with a highly parallel algebraic structure. Each MIRA block realizes a higher-order IIR system that represents temporal modes from token streams through a lattice–ladder structure, followed by a feed-forward network (FFN). This realization enables the model to represent the temporal modes associated with both real and complex-conjugate poles. We pretrain MIRA-S, a 12-layer small-scale LLM with 0.14B parameters composed entirely of MIRA blocks. Throughout the pretraining, MIRA-S achieves lower perplexity than strong recurrent baselines, exhibits stronger context-window generalization beyond the training length, and attains higher throughput with a dedicated CUDA kernel.

## 1 INTRODUCTION

Large language models (LLMs) based on the Transformer architecture (Vaswani et al., 2017) have achieved remarkable performance across a wide range of natural language processing tasks, including language understanding, reasoning, question answering, and code generation. Since GPT-2 (Radford et al., 2019), the prevailing paradigm has converged on decoder-only architectures and has remained largely unchanged: LLMs are built from stacks of Transformer blocks or its variants that combine causal scaled dot-product attention (SDPA) with feed-forward network (FFN), and are pretrained through next-token prediction on large-scale corpora followed by instruction tuning (Christiano et al., 2017; Ouyang et al., 2022; Rafailov et al., 2023). Despite its success, SDPA incurs quadratic computational complexity $\mathcal{O}(T^2)$ with respect to the sequence length $T$.

A growing body of work therefore explores linear-time architectures with computational complexity $\mathcal{O}(T)$. Selective state-space models (SSMs), such as Mamba (Gu & Dao, 2024) and its variants, replace the attention mixing module with selective state-space recurrences, while recurrent linear-attention models, such as Receptance Weighted Key Value (RWKV) (Peng et al., 2023), employ recurrent weighted *key–value* mechanisms. Other approaches preserve attention-like algebra while replacing the softmax operator: the Linear Transformer (Katharopoulos et al., 2020) factorizes kernelized attention through feature maps and associativity, Retentive Network (RetNet) (Sun et al., 2023) introduces a multi-scale retention mechanism with parallel, recurrent, and chunkwise-recurrent formulations, and Gated Linear Attention (GLA) (Yang et al., 2024a) incorporates data-dependent decay gates with efficient chunkwise computation. DeltaNet (Schlag et al., 2021; Yang et al., 2024b) introduces delta-rule updates to recurrent fast-weight states, while Gated DeltaNet (Yang et al., 2025) further augments the delta-rule recurrence with data-dependent gating. HGRN2 (Qin et al., 2024) develops hierarchical gated recurrent dynamics for linear-time sequence modeling, whereas Hyena (Poli et al., 2023) replaces recurrent state updates with implicitly parameterized long convolutions for efficient long-range sequence mixing.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: **Transfer-Function Characterization.** $A(z)$ and $B(z)$ denote the numerator (FIR) and denominator (IIR) polynomials of the transfer function $H(z) = A(z)/B(z)$. The transfer functions are input-dependent. The roots of $A(z) = 0$ are the *zeros* and the roots of $B(z) = 0$ are the *poles*. We also verify the derived transfer functions numerically, comparing the sequence each predicts against the measured output of its recurrence, with a maximum discrepancy of $3.6 \times 10^{-12}$; the pole ranges are verified formally, in Lean 4 (Moura & Ullrich, 2021) and Z3 (De Moura & Bjørner, 2008).

| Method | $A(z)$ | $B(z)$ | Zeroes | Poles |
|---|---|---|---|---|
| SDPA (Vaswani et al., 2017) | $\sum_{\tau=0}^{T-1} \alpha_{t,\tau} z^{-\tau}$ | $1$ | Complex | – |
| Selective SSM (Mamba-1, S6) (Gu & Dao, 2024) | $C\ \mathrm{adj}(I - Az^{-1})B + D\ \det(I - Az^{-1})$ | $\det(I - Az^{-1})$ | Complex | Real $(0, 1)$ |
| RWKV (v4–v6) (Peng et al., 2023) | $1$ | $I - Wz^{-1}$ | – | Real $(0, 1)$ |
| RetNet (v1) (Sun et al., 2023) | $1$ | $1 - \gamma z^{-1}$ | – | Real $[0.968, 1)$ |
| HGRN2 (v2) (Qin et al., 2024) | $B_t$ | $I - G_t z^{-1}$ | – | Real $(0.125, 1)$ |
| Hyena (v1) (Poli et al., 2023) | $\sum_{\tau=0}^{T-1} h_\tau z^{-\tau}$ | $1$ | Complex | – |
| DeltaNet (v1) (Yang et al., 2024b) | $B_t$ | $I - A_t z^{-1}$ | – | Real $(0, 1]$ |
| Gated DeltaNet (v1) (Yang et al., 2025) | $B_t$ | $I - \tilde{A}_t z^{-1}$ | – | Real $(0, 1)$ |
| MIRA | $\sum_{m=0}^{k} a_m(x_t) z^{-m}$ | $1 + \sum_{j=1}^{k} b_j(x_t) z^{-j}$ | Complex | Complex $\{\lvert z\rvert \le 1\}$ |

In the theory of digital signal processing (DSP) (Oppenheim & Schafer, 2010), representations of language sequences are discrete temporal signals. A system that maps an input sequence to an output sequence is a transfer operator. Its input–output response can be characterized by its transfer function. Let $x_{1:n} = (x_1, \cdots, x_n)$ be a finite temporal sequence of length $n$. Let $\mathcal{Z}\{\cdot\}$ be $z$-transform operator, and $\mathcal{Z}^{-1}\{\cdot\}$ be inverse $z$-transform operator, defined as

$$X_n(z) = \mathcal{Z}\{x_{1:n}\} := \sum_{j=1}^{n} x_j z^{-j}, \qquad x_j = \mathcal{Z}^{-1}\{X_n(z)\} = \frac{1}{2\pi \mathrm{i}} \oint_\Gamma X_n(z) z^{j-1} \mathrm{d}z,$$

respectively, where the counter-clockwise closed integral contour $\Gamma$ on complex plane encloses all poles of the polynomial $X_n(z)$ for ensuring causal condition and $z \in \mathbb{C}$ is complex variable (Oppenheim & Schafer, 2010). Let $x_{1:t}$ and $y_{1:t}$ be input and output sequences of $t$ tokens. Let $X_t(z) := \mathcal{Z}\{x_{1:t}\}$ and $Y_t(z) := \mathcal{Z}\{y_{1:t}\}$ denote the $z$-transforms of the input sequence $x_{1:t}$ and output sequence $y_{1:t}$, respectively. The ratio of the output to the input in the $z$-domain $H(z; x_t) = \frac{Y_t(z)}{X_t(z)}$ is referred to as the *transfer function* (Oppenheim & Schafer, 2010). We show that recurrent dynamical systems — such as SDPA, SSM, RWKV, RetNet, HGRN2, Hyena, and DeltaNet — admit following input-conditioned transfer operator,

$$H(z; x_t) = \frac{A(z; x_t)}{B(z; x_t)} = \frac{a_0(x_t) + \sum_{i=1}^{m} a_i(x_t) z^{-i}}{1 + \sum_{j=1}^{n} b_j(x_t) z^{-j}}, \qquad a_0, a_i, b_j \in \mathbb{R}, \tag{1}$$

where $A(z; x_t)$ defines an order-$m$ finite-impulse-response (FIR) system, and $B(z; x_t)$ defines an order-$n$ infinite-impulse-response (IIR) system. This transfer function $H(z; x_t)$ is uniquely characterized by the roots of its numerator and denominator polynomials, where the roots of the numerator polynomial $A(z; x_t)$ are referred to as *zeros*, and the roots of the denominator polynomial $B(z; x_t)$ are referred to as *poles*. The zero–pole geometry of $H(z; x_t)$ governs how temporal modes are retrieved from input sequences: zeros attenuate or cancel specific modes, whereas poles determine persistent or resonant temporal modes. Table 1 summarizes the transfer-function and zero–pole characterizations of SDPA, the selective SSM of Mamba-1, RWKV v4–v6, RetNet, HGRN2, Hyena, DeltaNet, and Gated DeltaNet. The proofs are provided in Appendix A.4. To trade parallelism, the algebraic structures of the parameterizations of these methods, such as a diagonal-matrix parameterization in SSM that preserves channel-wise parallelism, often constrain their poles to lie on the real axis.

This pole geometry constrained on real axis therefore induces an architectural bias in recurrent dynamics and delimits the modes that they can represent. To demonstrate this bias, we construct an experiment as shown in Figure 1, using a synthesized token sequence $x_{1:200}$ and target sequence $y_{1:200}$ with a dimension four, where the true transfer function $\mathcal{Z}\{y_{1:200}\}/\mathcal{Z}\{x_{1:200}\}$ has poles not constrained on real axis. We train single-layer MIRA, SDPA, Mamba-S6, and RWKV-7 models to perform next-token prediction for predicting the target token sequence using MSE (mean squared

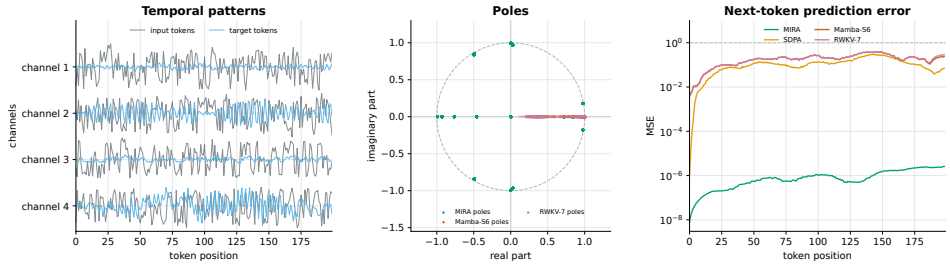


Figure 1: **Provable Failure of Causal Mode Retrieval.** We synthesize 4-dimensional continuous input tokens $x_{1:200}$ and target tokens $y_{1:200}$ of a length 200. The ground-truth transfer function contains poles that are not on the real axis. We use parameter-matched single-layer models trained to predict next-token $y_{t+1}$ from input sequence $x_{1:t}$ by using MSE (mean squared error) loss. Each model is optimized for 1000 steps using *AdamW* (Loshchilov & Hutter, 2019) with a learning rate of $10^{-2}$. *Left:* synthetic temporal signals of length 200. *Middle:* pole geometries of MIRA, Mamba-S6, RWKV-7, and SDPA. *Right:* next-token prediction error with logarithmic scale. The results show that MIRA admits complex poles and more accurately retrieves the temporal modes associated with complex poles. The provable failure signal synthesis is provided in Appendix A.1.

error) loss. The experiment demonstrates the provable failure of causal mode retrieval: models with poles constrained to the real axis fail to retrieve modes associated with complex-conjugate poles.

To unify the recurrent architectures and eliminate the pole geometric bias, inspired by general adaptive IIR filters in the form of equation (1), we present MIRA (**M**odal Adapt**i**ve **R**ecurrent L**a**ttice), pronounced "*mee-ruh*", a unified architectural family with linear time complexity for building LLMs with a highly parallel algebraic structure (see Figure 5). As illustrated in Figure 2, for each block, MIRA realizes a higher-order IIR system through a lattice–ladder structure (Oppenheim & Schafer, 2010), followed by a FFN. This structure enables the model to provably represent the feature modes associated with both real and complex-conjugate poles. Its recurrent dynamics inherently encode temporal order, eliminating the need for explicit positional encoding. The contributions are summarized as follows:

- **A Unified Architectural Family.** We introduce MIRA (Modal Adaptive Recurrent Lattice), a unified, positional-encoding-free, linear-time architectural family that realizes higher-order IIR dynamics through an adaptive lattice–ladder topology, enabling the representation of temporal modes associated with both real and complex-conjugate poles.
- **Dedicated CUDA Kernel.** We show that the algebraic structure of the MIRA realization admits computational associativity despite its higher-order recurrent form. We exploit this associativity to parallelize state propagation in our dedicated CUDA kernel, admitting a minimal critical path of $\mathcal{O}(\sqrt{T})$ where $T$ is sequence length.
- **Experimental Validation.** Controlled experiments on language corpora show that enabling complex poles improves language-modeling performance, while inference measurements with dedicated CUDA kernel show that the MIRA block achieves higher throughput than strong baselines. We further build and pretrain MIRA-S, a small-scale 0.14B-parameter LLM composed entirely of MIRA blocks, which achieves the highest next-token prediction accuracy and the lowest perplexity among strong baselines.

## 2 A Unified Parameterized Lattice–Ladder Realization

We treat token sequences in language modeling as the propagation of high-dimensional temporal signals. Since recurrent models admit a unified representation as IIR systems, we seek a unified IIR parameterization whose algebraic structure facilitates GPU parallelism without imposing a pole geometry bias. As illustrated in Figure 2 (left), the lattice–ladder structure (Campbell, 1922; Gray & Markel, 1973) is a canonical realization form of general IIR systems, arising naturally from coupled wave propagation in modern physical systems such as photonic crystals and metawaveguides (Yablonovitch, 1987). Intuitively, a lattice-ladder realization represents the propagation of mode-coupled input and readout signals in an IIR system: (i) forward signals (*i.e.*, input signals)

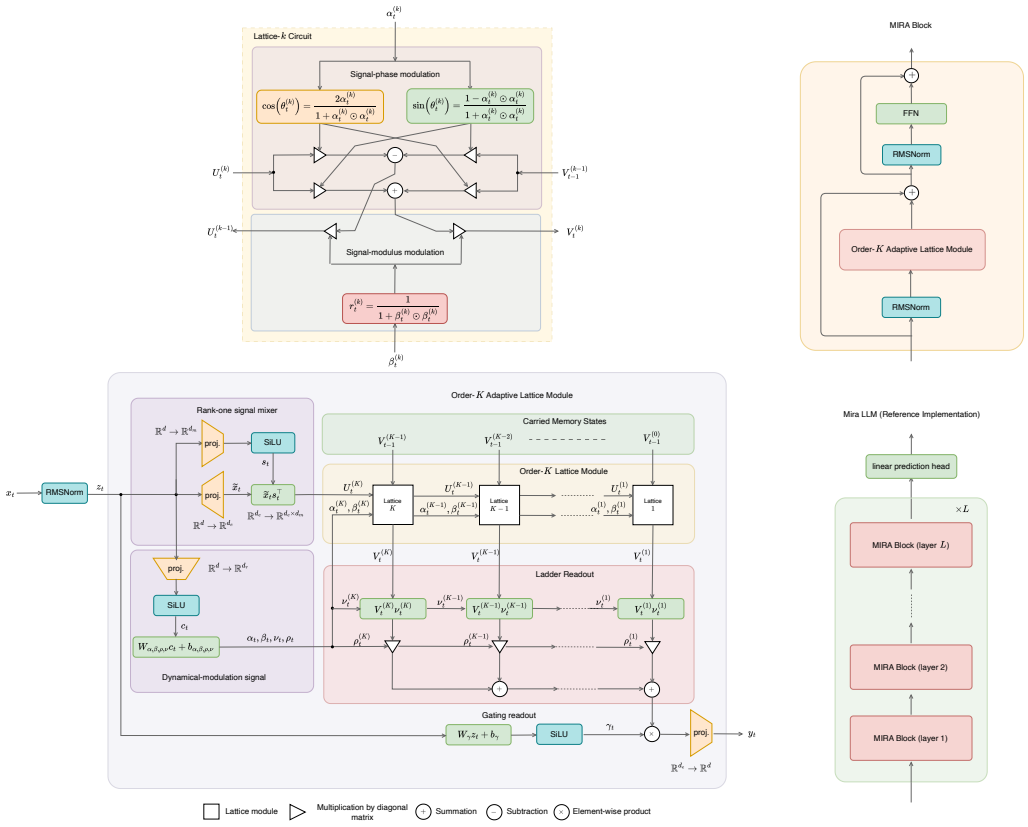


Figure 2: **MIRA Block & Reference LLM.** The diagram illustrates lattice circuit at stage $k$ (top left), an order-$K$ lattice module (bottom left), a MIRA block (top right), and a reference implementation of the MIRA LLM composed entirely of MIRA blocks (bottom right). In the order-$K$ lattice module, a rank-one signal mixer forms the input representations through channel-wise signal mixing, the dynamical-modulation signal modulates the lattice–ladder dynamics, the lattice module implements the order-$K$ lattice dynamics, and the ladder readout produces the output representations.

that represent input tokens and (ii) backward signals (*i.e.*, readout signals) that carry both the temporal memories and token representations. The mode coupling between the input and readout signals enables a parameterized IIR-based recurrent system to represent modes without pole geometric bias. The algebraic structure of this realization enables associative composition along the temporal axis, thereby facilitating GPU parallelism.

**Stable Recurrent Lattice Dynamics.** Let $x_{1:T}$ denote an input token representation sequence, where each token has dimension $d$. As illustrated in Figure 2, we first normalize $x_{1:T}$ using RMSNorm (Root Mean Square Layer Normalization) (Zhang & Sennrich, 2019) and project the representations into an expanded dimension $d_e > d$, allowing the lattice dynamics to operate in a higher-dimensional space

$$z_t = \mathrm{RMSNorm}(x_t), \quad \widetilde{x}_t = W_{\mathrm{in}} z_t, \quad x_t \in \mathbb{R}^d, \quad W_{\mathrm{in}} \in \mathbb{R}^{d_e \times d},$$

where $W_{\mathrm{in}}$ denotes the input projection matrix and $d_e/d$ is referred to as *expansion factor*. Let $U_t^{(k)} \in \mathbb{R}^{d_e \times d_m}$ denote the forward token signal and $V_t^{(k)} \in \mathbb{R}^{d_e \times d_m}$ denote the backward state signal at lattice stage $k$, where $t \in \{1, \ldots, T\}$ represents the token position and $k \in \{K, \ldots, 0\}$ indexes the lattice stage and $d_m$ is referred to as *per-channel memory size*. We construct a generalized,

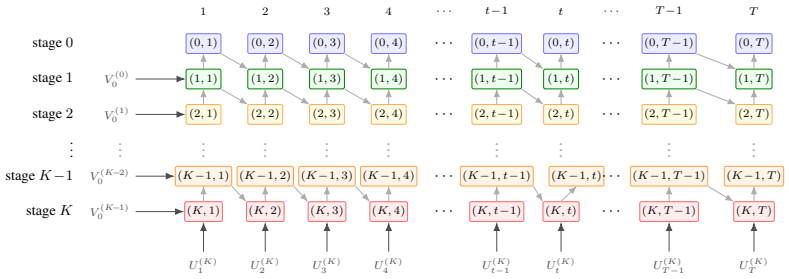


Figure 3: **Algebraic Structure of the Lattice Realization.** Each node $(k, t)$ represents the recursive state transition $\left(U_t^{(k)}, V_{t-1}^{(k-1)}\right) \mapsto \left(U_t^{(k-1)}, V_t^{(k)}\right)$ at lattice stage $k \in \{K, \ldots, 1\}$ and token position $t \in \{1, \ldots, T\}$, with edges encoding dependencies on $(k+1, t)$ and $(k-1, t-1)$.

stable, parameterized, matrix-valued dual-port lattice scattering dynamics as follows

$$\begin{aligned} \begin{pmatrix} U_t^{(k-1)} \\ V_t^{(k)} \end{pmatrix} &= A_t^{(k)} \begin{pmatrix} U_t^{(k)} \\ V_{t-1}^{(k-1)} \end{pmatrix}, \quad A_t^{(k)} = r_t^{(k)} \odot R_t^{(k)}, \quad r_t^{(k)} \in \mathbb{R}^{d_e}, \\ R_t^{(k)} &= \begin{pmatrix} \operatorname{diag}\left(\cos\left(\theta_t^{(k)}\right)\right) & -\operatorname{diag}\left(\sin\left(\theta_t^{(k)}\right)\right) \\ \operatorname{diag}\left(\sin\left(\theta_t^{(k)}\right)\right) & \operatorname{diag}\left(\cos\left(\theta_t^{(k)}\right)\right) \end{pmatrix}, \quad R_t^{(k)} \in \mathbb{R}^{2d_e \times 2d_e}, \end{aligned} \tag{2}$$

under $k = \{K, \cdots, 1\}$ and with initial conditions

$$\begin{aligned} &s_t = \mathrm{SiLU}(W_s z_t), \quad U_t^{(K)} = \widetilde{x}_t s_t^\top, \quad W_s \in \mathbb{R}^{d_m \times d}, && \text{(rank-one signal mixer)}, \\ &V_t^{(0)} = U_t^{(0)}, \quad V_0^{(k)} = 0, && \text{(boundary conditions)}, \end{aligned}$$

where operator $\odot$ denotes the generalized element-wise product with broadcasting defined by $r_t^{(k)} \odot R_t^{(k)} = (I_2 \otimes \operatorname{diag}(r_t^{(k)}))R_t^{(k)}$, $U_t^{(k)}$ is referred to as the *forward state* which encodes the representation of token $t$ at lattice stage $k$, $V_t^{(k)}$ is referred to as the *backward state* which summarizes the recurrent memory up to token $t$ at lattice stage $k$, and $A_t^{(k)}$ is referred to as the *reflection coefficient matrix*, which governs the forward-backward state transition. This IIR system is parameterized through

$$\begin{aligned} &c_t = \mathrm{SiLU}(W_c z_t + b_c), \quad W_c \in \mathbb{R}^{d_r \times d}, \quad b_c \in \mathbb{R}^{d_r}, \\ &\sin(\theta_t) = \frac{1-\alpha_t^2}{1+\alpha_t^2}, \quad \cos(\theta_t) = \frac{2\alpha_t}{1+\alpha_t^2}, \quad \alpha_t = W_\alpha c_t + b_\alpha, \quad W_\alpha \in \mathbb{R}^{Kd_e \times d_r}, \\ &r_t = \frac{1}{1+\beta_t^2}, \quad \beta_t = W_\beta c_t + b_\beta, \quad W_\beta \in \mathbb{R}^{Kd_e \times d_r}, \end{aligned}$$

where $d_r \ll d$ introduces a low-rank state-write bottleneck that enables state writes on a low-dimensional manifold. We construct the rotation matrix $R_t^{(k)}$ as a Cayley map (Cayley, 1846) parameterized by a Möbius transformation $\frac{1-\alpha_t^{(k)}}{1+\alpha_t^{(k)}}$ of $\alpha_t^{(k)}$ (Möbius, 1855) (see Appendix A.2). The states $V_0^{(k)}$ are reset for every independent input sequence. We refer to $K$, $d_r$, and $d_m$ as the *lattice order*, *state-write bottleneck size*, and *memory size*, respectively. We further refer to $H_t = -1/\log r_t$ as the *characteristic horizon*, which governs the decay timescale of the backward state mode $V_t$ with discrete-time pole $r_t e^{\mathrm{i}\theta_t}$, since the continuous-time ordinary differential equation (ODE) of the backward state evolution satisfies $\frac{\mathrm{d}V_t}{\mathrm{d}t} = (\log r_t + \mathrm{i}\theta_t)\, V_t = \left(-\frac{1}{H_t} + \mathrm{i}\theta_t\right) V_t$.

**Representation Readout.** Since the backward state $V_t^{(k)}$ carries both the current representation of token $t$ and the recurrent memory of preceding tokens, therefore the ladder readout aggregates the backward states across the lattice stages using the input-conditioned ladder readout coefficient $\rho_t^{(k)} \in \mathbb{R}^{d_e}$, the learned state-read direction $\nu_t^{(k)} \in \mathbb{R}^{d_m}$, and the selective gate $\gamma_t \in \mathbb{R}^{d_e}$

$$y_t = W_{\text{out}} \left( \left( \sum_{k=0}^{K} \rho_t^{(k)} \odot \left( V_t^{(k)}\, \nu_t^{(k)} \right) \right) \odot \gamma_t \right), \quad W_{\text{out}} \in \mathbb{R}^{d \times d_e},$$

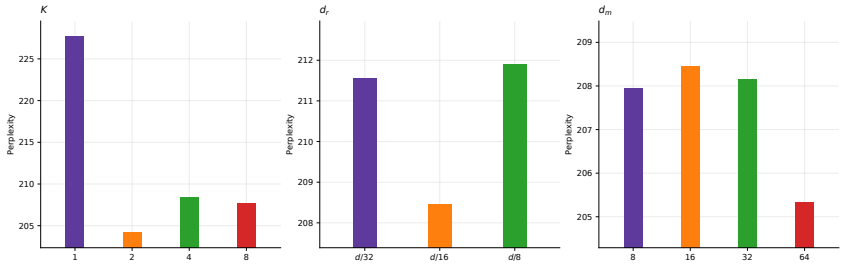


Figure 4: **Effects of Lattice Order $K$, State-Write Bottleneck Size $d_r$, and Memory Size $d_m$.** Using MIRA-XS (65M), we independently vary the lattice order $K \in \{1, 2, 4, 8\}$, state-write bottleneck size $d_r \in \{d/32, d/16, d/8\}$, and memory size $d_m \in \{8, 16, 32, 64\}$. We match model sizes by varying only the model width $d$ and pretrain each model for 10,000 steps on *FineWeb-Edu* using *AdamW*. The results show that models admitting complex-conjugate poles ($K > 1$) achieve lower perplexity than the model with real poles ($K = 1$). Among the evaluated state-write bottleneck sizes, $d_r = d/16$ yields the lowest perplexity. Perplexity generally decreases as $d_m$ increases, reaching its lowest value at $d_m = 64$.

where $W_{\text{out}}$ denotes the readout projection matrix. The readout coefficient $\rho_t$, state-read direction $\nu_t$, and readout gate $\gamma_t$ are parameterized as

$$\begin{aligned}
\rho_t &= W_\rho\, c_t + b_\rho, \qquad W_\rho \in \mathbb{R}^{(K+1)d_e \times r}, \quad b_\rho \in \mathbb{R}^{(K+1)d_e}, \\
\nu_t &= W_\nu\, c_t + b_\nu, \qquad W_\nu \in \mathbb{R}^{(K+1)d_m \times r}, \quad b_\nu \in \mathbb{R}^{(K+1)d_m}, \\
\gamma_t &= \mathrm{SiLU}\left(W_\gamma z_t + b_\gamma\right), \qquad W_\gamma \in \mathbb{R}^{d_e \times d}, \quad b_\gamma \in \mathbb{R}^{d_e}.
\end{aligned}$$

**Stability Condition.** Remaining Schur stability of the recurrent dynamics requires all the eigenvalues of $\mathrm{diag}\big(r_t^{(k)} \odot \sin\big(\theta_t^{(k)}\big)\big)$ to have magnitude strictly smaller than 1 (Schur, 1917; Cohn, 1922; Oppenheim & Schafer, 2010). Our parameterization satisfies this condition almost surely under an absolutely continuous joint distribution of $\alpha_t^{(k)}$ and $\beta_t^{(k)}$, since their corresponding entries vanish simultaneously only on a measure-zero set. The proof is provided in Appendix A.5.

**Non-Degenerate Schur Initializer.** The initialization of $\alpha_t$ and $\beta_t$ must ensure a non-nilpotent state transition $\big(U_t^{(k)}, V_{t-1}^{(k-1)}\big) \mapsto \big(U_t^{(k-1)}, V_t^{(k)}\big)$ and meaningful characteristic horizons. The recurrent dynamics degenerate under trivial initializations, such as small zero-centered uniform and normal initializers, because the cross-stage magnitude of state transition decreases exponentially with $K$, numerically annihilating higher-order dynamics. When stage coupling vanishes or exact pole–zero cancellations occur, the McMillan degree (Kalman, 1965) falls below $K$, potentially degenerating the IIR dynamics into an FIR system. We therefore design the Schur initializer described in Appendix A.3 to obtain non-nilpotent, dynamically coupled higher-order IIR dynamics.

## 3 Algebraic Structure and Parallelism

Let

$$(k,t) : \big(U_t^{(k)}, V_{t-1}^{(k-1)}\big) \longmapsto \big(U_t^{(k-1)}, V_t^{(k)}\big), \qquad k \in \{K, \ldots, 1\}, \quad t \in \{1, \ldots, T\}$$

be the state transition given by equation (2), as illustrated in Figure 3. Directly using equation (2) for computing the dynamics up to token position $t$ requires all previous states. To parallelize the computation on GPUs, we need to further explore its algebraic structure that allows parallelism.

**Algebraic Associativity & State Interposition.** Since $U_t^{(k)}$ propagates only across lattice stages, while $V_t^{(k)}$ carries the temporal memory up to token position $t$, we define the carried state as

$$\mathbf{V}_t = \big(V_t^{(0)}, \ldots, V_t^{(K-1)}\big)^\top \in \mathbb{R}^{Kd_e \times d_m}, \quad \mathbf{V}_0 = 0.$$

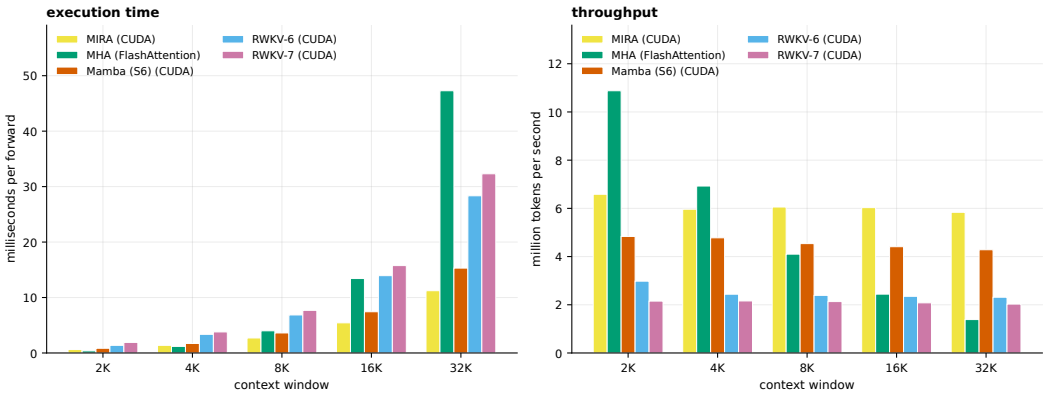


Figure 5: **Performance of Dedicated CUDA Kernel.** We compare MIRA module with parameter-matched baseline modules on an NVIDIA RTX A6000 GPU. *Execution time:* at a context length of $32K$, MIRA completes a forward pass in 11.23 ms, yielding speedups of $4.21\times$, $1.36\times$, $2.52\times$, and $2.88\times$ over MHA (Multi-Head Attention) with FlashAttention, Mamba (S6), RWKV-6, and RWKV-7, respectively (left). *Throughput:* At the same context length, MIRA reaches 5.84 million tokens per second, exceeding the throughput of the respective baselines by factors of $4.21\times$, $1.36\times$, $2.52\times$, and $2.88\times$ (right).

Because $\widetilde{x}_t$ and $s_t$ depend only on $x_t$, we show that the complete cascade $\{(k,t)\}_{k=K}^{1}$ admits an affine operator

$$\mathcal{T}_t = \big(\Phi_t(\alpha_t), \mathbf{S}_t(\alpha_t, \widetilde{x}_t, s_t)\big), \quad \mathcal{T}_t(\mathbf{V}) = \Phi_t(\alpha_t)\mathbf{V} + \mathbf{S}_t(\alpha_t, \widetilde{x}_t, s_t), \quad \mathbf{V}_t = \mathcal{T}_t\big(\mathbf{V}_{t-1}\big), \tag{3}$$

where $\Phi_t(\alpha_t) \in \mathbb{R}^{Kd_e \times Kd_e}$ denotes the transition matrix and $\mathbf{S}_t(\alpha_t, \widetilde{x}_t, s_t) \in \mathbb{R}^{Kd_e \times d_m}$ denotes the input contribution independent of any $V_t^{(k)}$. For any $t_1 < t_2 < t_3$, temporal composition of two affine operators is associative

$$\big(\mathcal{T}_{t_3} \odot \mathcal{T}_{t_2}\big) \odot \mathcal{T}_{t_1} = (\Phi_{t_3}\Phi_{t_2}\Phi_{t_1}, \Phi_{t_3}\Phi_{t_2}\mathbf{S}_{t_1} + \Phi_{t_3}\mathbf{S}_{t_2} + \mathbf{S}_{t_3}) = \mathcal{T}_{t_3} \odot \big(\mathcal{T}_{t_2} \odot \mathcal{T}_{t_1}\big). \tag{4}$$

Therefore, for any $1 \le s < t \le T$, this algebraic structure interposes the preceding memory and the segment dynamics within the carried state

$$\mathbf{V}_t = \underbrace{\Phi_{s\to t}\,\mathbf{V}_s}_{\text{preceding memory up to token } s} + \underbrace{\mathbf{S}_{s\to t}}_{\text{segment dynamics from } \mathbf{V}_s=0}, \tag{5}$$

where $\big(\Phi_{s\to t}, \mathbf{S}_{s\to t}\big)$ composes $\mathcal{T}_{s+1}, \ldots, \mathcal{T}_t$: the transition $\Phi_{s\to t}$ transports the preceding memory $\mathbf{V}_s$ accumulated over tokens $1, \ldots, s$, whereas the input contribution $\mathbf{S}_{s\to t}$ is accumulated over the segment alone from the zero entry state $\mathbf{V}_s = 0$, and is therefore independent of all tokens preceding $s$. The proof and further details are provided in Appendix A.6.

**Critical Parallel Path.** Theoretically, this algebraic structure enables chunkwise parallel state propagation along with token axis $1 \to T$ and admits a critical path $2L + T/L$, where $L$ is chunk size of token sequence. A minimal critical path of $\mathcal{O}(\sqrt{T})$ is obtained by choosing the chunk size as $L \approx \sqrt{T/2}$.

**Dedicated CUDA Kernel.** We realize this algebraic structure as a dedicated CUDA kernel, such that the computation within the MIRA module remains on-chip. Exploiting associativity in equation (4) and state interposition in equation (5), the token axis is partitioned into $N_C = \lceil T/L \rceil$ chunks of length $L$, and three phases are mapped onto thread blocks, each processing one chunk across adjacent channels with the carried state held in registers: (i) chunk summarization evaluates every chunk concurrently from the zero carried state, yielding its pair $(\Phi_{s\to t}, \mathbf{S}_{s\to t})$; (ii) associative composition applies equation (5) at the chunk boundaries alone, recovering every chunk-entry state; and (iii) chunk replay restores the interior states concurrently and produces the gated ladder readout. The details are provided in Appendix A.7.

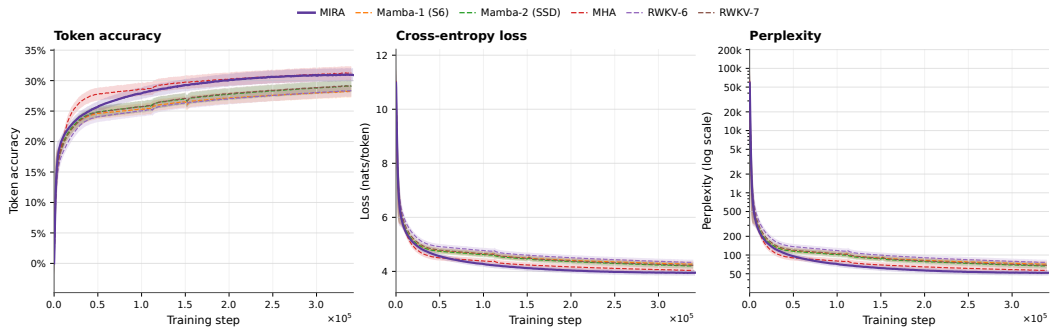


Figure 6: **Pretraining Performance.** We implement a small-scale reference LLM composed entirely of MIRA blocks, denoted as MIRA-S. Pretraining results show that MIRA-S achieves the lowest perplexity among strong recurrent language model baselines and Transformer-based MHA, while attaining higher token-prediction accuracy than the recurrent baselines and comparable token-prediction accuracy to MHA.

## 4 EXPERIMENTS

We build a small-scale model MIRA-S for benchmarking against strong baselines of comparable size, comprising 12 layers with model width $d = 768$, expansion width $d_e = 1536$, memory dimension $d_m = 16$, state-write bottleneck dimension $d_r = d/16$, lattice order $K = 2$, and FFN width 3072. The model has 0.14B parameters and a state-to-parameter ratio of $4.076$ ‰. We also build an extra-small-scale model MIRA-XS for architectural ablation experiments, comprising 10 layers with model width $d = 512$, expansion width $d_e = 1024$, memory dimension $d_m = 16$, write bottleneck dimension $d_r = d/16$, lattice order $K = 2$, and FFN width 2048. The model has 65M parameters and a state-to-parameter ratio of $9.760$ ‰. For both MIRA-S and MIRA-XS, we initialize the characteristic horizon of Schur initializer ranging from 10 tokens to 2000 tokens.

**Training Protocol.** We use the *FineWeb-Edu* dataset (text corpus) (Penedo et al., 2024) to train a BPE tokenizer and yield a tokenizer with a vocabulary of 50,259 tokens. The tokenizer reserves three special tokens: <|endoftext|>, <|unk|> and <|pad|>. We pretrain each model with a weight-update budget corresponding to three billion tokens, following the Chinchilla scaling rule (Hoffmann et al., 2022), with a fixed context length of 512 tokens. For the next-token prediction objective, we weight each sequence-level cross-entropy loss by its number of valid target tokens, thereby averaging the loss uniformly over all predicted tokens (Kaplan et al., 2020). To reduce the GPU memory footprint, we use chunked loss (Liu & Abbeel, 2023) with a chunk size of 8,192 and mixed-precision training, where the model weights and optimizer states are stored in 32-bit floating point (FP32), while activations are stored in brain floating point (BF16) (Kalamkar et al., 2019). We optimize the model using *AdamW* (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0, and a numerical stability constant of $\varepsilon = 10^{-6}$ for mixed-precision training and $\varepsilon = 10^{-8}$ for full-precision training. We use a peak learning rate of $2 \times 10^{-5}$, and apply a cosine learning-rate scheduler with a minimum learning rate of $2 \times 10^{-6}$ and a linear learning-rate warmup of 1000 steps. Before each weight update, we apply gradient clipping with a maximum norm of 1.0.

**Architectural Ablation.** For MIRA-XS, we independently vary the lattice order $K \in \{1, 2, 4, 8\}$ while fixing $d_r = 32$ and $d_m = 16$; vary the write bottleneck size $d_r \in \{d/32, d/16, d/8\}$ while fixing $K = 2$ and $d_m = 16$; and vary the memory size $d_m \in \{8, 16, 32, 64\}$ while fixing $K = 2$ and $d_r = 32$. We match model sizes by varying only the model width $d$ and pretrain each model for 10,000 steps on *FineWeb-Edu*. Figure 4 shows that models admitting complex-conjugate poles ($K > 1$) achieve lower perplexity than the model with real poles ($K = 1$), while $d_r = d/16$ yields the lowest perplexity among the evaluated state-write bottleneck sizes. Perplexity generally decreases as $d_m$ increases, reaching its lowest value at $d_m = 64$. Since the MIRA module has parameter size $\mathcal{O}(K(d_e + d_m)d_r + d_e + d_m + d_r)$, therefore higher lattice orders, wider state-write bottlenecks, and larger memory sizes incur greater computational costs and memory footprints, we choose $K = 2$, $d_r = d/16$, and $d_m = 16$ as the default configuration scheme of MIRA, balancing performance and computational efficiency.

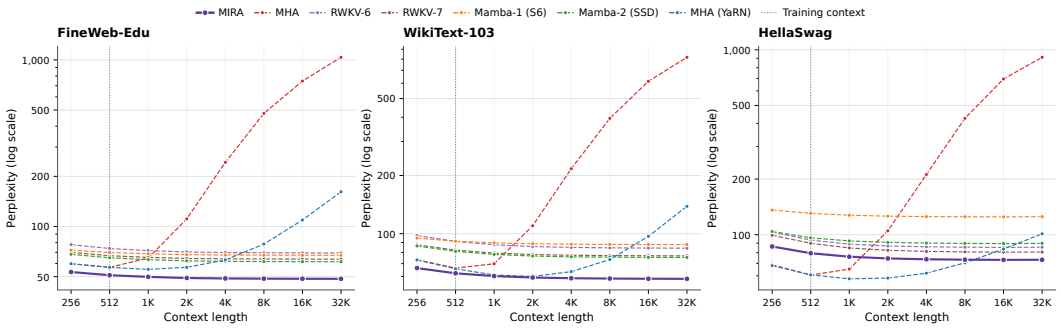


Figure 7: **Context-Window Generalization.** We evaluate the context-window generalization for the pretrained MIRA on a holdout set of *FineWeb-Edu*, *WikiText-103*, and *HellaSwag*, respectively. The dotted vertical line marks the training context length. MIRA attains the lowest perplexity among the recurrent baselines and matches or improves on the MHA reference.

**CUDA Kernel Performance.** We benchmark the dedicated MIRA CUDA kernel against strong GPU-optimized baselines whose CUDA kernels are available, including MHA with FlashAttention (Dao et al., 2022), Mamba (S6), RWKV-6, and RWKV-7, using parameter-matched models on an NVIDIA RTX A6000 GPU. As shown in Figure 5, MIRA consistently achieves lower execution time and higher throughput across the evaluated context lengths. At a context length of 32,768, MIRA completes a forward pass in 11.23 ms, corresponding to speedups of $4.21\times$, $1.36\times$, $2.52\times$, and $2.88\times$ over MHA with FlashAttention, Mamba (S6), RWKV-6, and RWKV-7, respectively. At the same context length, MIRA achieves a throughput of 5.84 million tokens per second, exceeding the respective baselines by $4.21\times$, $1.36\times$, $2.52\times$, and $2.88\times$. Compared with the baselines, MIRA exposes parallelism along with token chunks, expanded channels, and memory components, while its fixed number of lattice stages is unrolled at compile time.

**Pretraining Benchmark.** We pretrain MIRA-S on *FineWeb-Edu* and compare MIRA-S with strong recurrent language model baselines, including Mamba-v1 (S6), Mamba-v2 (SSD), RWKV-6, and RWKV-7, with MHA as a reference baseline. The Transformer-based MHA baseline uses RoPE (Rotary Position Embedding) (Su et al., 2024) for positional encoding. For a fair comparison, we adjust the model width of each architecture to maintain the approximately identical parameter size. The training results are shown in Figure 6. Throughout the pretraining, MIRA-S consistently exhibits lower perplexity and higher next-token prediction accuracy than the strong baselines. *Note:* Pretraining resumes from an earlier checkpoint at approximately steps $1.1 \times 10^5$ and $1.5 \times 10^5$, respectively, because the baseline losses exhibit spikes.

**Context Window Generalization.** We evaluate context-length generalization for MIRA, Mamba-1 (S6), Mamba-2 (SSD), RWKV-6, and RWKV-7, with MHA as a reference, on three corpora: a held-out set of *FineWeb-Edu* (230 documents, 263,837 tokens), *WikiText-103* (Merity et al., 2016), and *HellaSwag* (Zellers et al., 2019). Besides MHA without positional-encoding extrapolation (legend "MHA"), we also apply YaRN (Peng et al., 2024) to MHA beyond the training context length, with legend "MHA (YaRN)". Figure 7 reports perplexity for context lengths from 256 to 32,768 tokens, with a fixed training context of 512 tokens. Throughout the evaluation, MIRA attains the lowest perplexity among the strong recurrent baselines and matches or improves on MHA.

## 5 Conclusions

We present MIRA, a recurrent, linear-time architectural family that eliminates the pole-geometry bias. MIRA unifies recurrent architectures and removes the pole-geometry bias through lattice–ladder realizations of higher-order IIR systems. The lattice–ladder parameterization admits chunk-wise associativity, enabling efficient GPU parallelism. We implement a dedicated CUDA kernel by leveraging this algebraic structure. Language modeling experiments demonstrate that MIRA outperforms strong recurrent language model baselines and generalizes beyond the training context without explicit positional encoding. These results establish MIRA as a promising direction for efficient LLMs.

## AI USE STATEMENT

In this work, we used generative AI tools for: polishing writing drafted by the human authors; checking for typographical and grammatical errors; refining wording choices; partially assisting with the development of experimental code; assisting human mathematicians in verifying and normalizing proofs (*e.g.*, standardizing notations); and retrieving related literature. We have not used generative AI tools for drafting the manuscript, fully deriving proofs without the supervision of the human mathematician (the first author), or developing the theoretical model, retrieving bibliography, and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. For example, we carefully reviewed LLM-polished sentences; verified LLM-normalized proofs through numerical experiments, symbolic algebra, and independent examination by human mathematicians; and manually verified LLM-aided literature retrieval by consulting the ACM Digital Library or the literature's original publication venues, such as *OpenReview*, and having human experts read the retrieved works. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

George A. Campbell. Physical theory of the electric wave-filter. *Bell System Technical Journal*, 1(2): 1–32, 1922. doi: https://doi.org/10.1002/j.1538-7305.1922.tb00386.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1922.tb00386.x.

Arthur Cayley. Sur quelques propriétés des déterminants gauches. *Journal für die reine und angewandte Mathematik*, 32:119–123, 1846.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

A. Cohn. Über die anzahl der wurzeln einer algebraischen gleichung in einem kreise. *Mathematische Zeitschrift*, 14(1):110–148, 1922. doi: 10.1007/BF01215894. URL https://doi.org/10.1007/BF01215894.

Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Re. Flashattention: Fast and memory-efficient exact attention with IO-awareness. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=H4DqfPSibmx.

Leonardo De Moura and Nikolaj Bjørner. Z3: an efficient smt solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'08/ETAPS'08, pp. 337–340, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 3540787992.

A. Gray and J. Markel. Digital lattice and ladder filter synthesis. *IEEE Transactions on Audio and Electroacoustics*, 21(6):491–500, 1973. doi: 10.1109/TAU.1973.1162522.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. A study of bfloat16 for deep learning training, 2019. URL https://arxiv.org/abs/1905.12322.

R. E. Kalman. Irreducible realizations and the degree of a rational matrix. *Journal of the Society for Industrial and Applied Mathematics*, 13(2):520–544, 1965. doi: 10.1137/0113034. URL https://doi.org/10.1137/0113034.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Hao Liu and Pieter Abbeel. Blockwise parallel transformers for large context models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KbqQMoqfLQ.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

August Ferdinand Möbius. Die theorie der kreisverwandtschaft in rein geometrischer darstellung. *Abhandlungen der mathematisch-physischen Classe der Königlich Sächsischen Gesellschaft der Wissenschaften*, 2:529–595, 1855.

Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings*, pp. 625–635, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-79875-8. doi: 10.1007/978-3-030-79876-5_37. URL https://doi.org/10.1007/978-3-030-79876-5_37.

Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Pearson, 3rd edition, 2010.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In Houda Bouamor, Juan Pino, and Kalika Bali (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/.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: towards larger convolutional language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. HGRN2: Gated linear RNNs with state expansion. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=y6SqbJfCSk.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/schlag21a.html.

J. Schur. Über potenzreihen, die im innern des einheitskreises beschränkt sind. *Journal für die reine und angewandte Mathematik*, 1917(147):205–232, 1917. doi: doi:10.1515/crll.1917.147.205. URL https://doi.org/10.1515/crll.1917.147.205.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. arXiv preprint, August 2023. URL https://www.microsoft.com/en-us/research/publication/retentive-network-a-successor-to-transformer-for-large-language-models/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Eli Yablonovitch. Inhibited spontaneous emission in solid-state physics and electronics. *Phys. Rev. Lett.*, 58:2059–2062, May 1987. doi: 10.1103/PhysRevLett.58.2059. URL https://link.aps.org/doi/10.1103/PhysRevLett.58.2059.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56501–56523. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/yang24ab.html.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=y8Rm4VNRPH.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Biao Zhang and Rico Sennrich. *Root mean square layer normalization*. Curran Associates Inc., Red Hook, NY, USA, 2019.

# A Appendix

## A.1 Synthesizing Provable Failure Signals for Causal Mode Retrieval

We synthesize four-channel input and target signals related by a true transfer function with nonreal poles. We first construct the input signal $X(z)$ and the true transfer function $H(z)$, and then generate the target signal as $Y(z) = H(z)X(z)$. The transfer function $H(z)$ is withheld from the models and must be learned from the input–target sequences.

**Input Signal.** We construct an input signal containing both magnitude and phase noise. For a sequence of $T$ tokens, with token index $t \in \{1, \dots, T\}$ and channel index $c \in \{1, \dots, 4\}$, the latent input is

$$u_{t,c} = 0.75\epsilon_{t,c} + 0.45\sin\big(\omega_c(t-1) + \phi_c\big) + 0.25\sin\big(1.7\omega_c(t-1) - \phi_c\big), \tag{6}$$

where $\epsilon_{t,c} \sim \mathcal{N}(0,1)$ denotes Gaussian noise, $\phi_c \sim \mathcal{U}(-\pi, \pi)$ denotes a random phase, and

$$(\omega_1, \omega_2, \omega_3, \omega_4) = \pi(0.07, 0.11, 0.17, 0.23). \tag{7}$$

The unnormalized input is

$$\begin{aligned}\widetilde{x}_{t,c} = {} & \sin(s_c u_{t,c}) + \lambda_c \operatorname{sgn}(u_{t,c}) \log(1 + |u_{t,c}|) \\ & + q_c \left( e^{-|u_{t,c}|} - \frac{1}{T} \sum_{\tau=1}^{T} e^{-|u_{\tau,c}|} \right) + 0.12\xi_{t,c},\end{aligned} \tag{8}$$

where $\xi_{t,c} \sim \mathcal{N}(0,1)$ denotes independent Gaussian noise and

$$(s_1, s_2, s_3, s_4) = (0.85, 1.05, 1.20, 1.40), \tag{9}$$

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0.35, 0.25, 0.30, 0.20), \tag{10}$$

$$(q_1, q_2, q_3, q_4) = (0.20, 0.30, 0.25, 0.35). \tag{11}$$

Using the channel mean

$$\mu_c = \frac{1}{T} \sum_{t=1}^{T} \widetilde{x}_{t,c} \tag{12}$$

and channel standard deviation

$$\sigma_c = \left[ \frac{1}{T} \sum_{t=1}^{T} (\widetilde{x}_{t,c} - \mu_c)^2 \right]^{1/2}, \tag{13}$$

the normalized input is

$$x_{t,c} = \frac{\widetilde{x}_{t,c} - \mu_c}{\sigma_c}. \tag{14}$$

**True Transfer Function.** For each channel $c \in \{1, \dots, 4\}$, we construct the true transfer function as

$$H_c(z) = \frac{Y_c(z)}{X_c(z)} = \frac{z^{-1} \sum_{j=0}^{4} b_{c,j} z^{-j}}{1 - \sum_{j=1}^{4} a_{c,j} z^{-j}}, \tag{15}$$

where the coefficient $a_{c,j}$ is entry $(c, j)$ of

$$\mathbf{A} = \begin{pmatrix} 1.952735784 & -1.974230161 & 1.930926548 & -0.974392257 \\ -1.400051941 & -1.315949473 & -1.333286522 & -0.427346553 \\ -1.030066375 & -0.458960346 & 0.497746636 & 0.527108251 \\ -1.086474355 & -0.263198919 & 0.686737838 & 0.743993180 \end{pmatrix} \tag{16}$$

and the numerator coefficient $b_{c,j}$ is entry $(c, j+1)$ of

$$\mathbf{B} = \begin{pmatrix} -0.055171491 & 0.136620626 & -0.188283824 & 0.201492045 & -0.097161900 \\ -0.146266818 & -0.064363154 & 0.155433501 & 0.021868375 & -0.057037339 \\ 0.144469820 & 0.163038836 & 0.134055985 & -0.021828775 & -0.044674776 \\ -0.226198180 & -0.514066284 & -0.387318389 & -0.199410167 & -0.092755378 \end{pmatrix}. \tag{17}$$

Multiplying the numerator and denominator by $z^5$ gives

$$H_c(z) = \frac{b_{c,0}z^4 + b_{c,1}z^3 + b_{c,2}z^2 + b_{c,3}z + b_{c,4}}{z\left(z^4 - a_{c,1}z^3 - a_{c,2}z^2 - a_{c,3}z - a_{c,4}\right)}, \tag{18}$$

in which the factor $z$ in the denominator is the one-token delay of (15) and contributes a further pole at the origin. The zeros for channel $c$ are the roots of the numerator polynomial:

$$\mathcal{Z}_c = \left\{z \in \mathbb{C} : b_{c,0}z^4 + b_{c,1}z^3 + b_{c,2}z^2 + b_{c,3}z + b_{c,4} = 0\right\}, \tag{19}$$

and the poles are the roots of the denominator polynomial:

$$\mathcal{P}_c = \left\{z \in \mathbb{C} : z\left(z^4 - a_{c,1}z^3 - a_{c,2}z^2 - a_{c,3}z - a_{c,4}\right) = 0\right\}. \tag{20}$$

For the specified values of $a_{c,j}$ and $b_{c,j}$, the resulting zero and pole sets are listed below, rounded to six decimal places. The zero sets are

$$\mathcal{Z}_1 = \{0.171722 \pm 1.223544\mathrm{i},\ 1.066424 \pm 0.127992\mathrm{i}\}, \tag{21}$$
$$\mathcal{Z}_2 = \{-0.908806,\ -0.857417,\ 0.663092 \pm 0.246471\mathrm{i}\}, \tag{22}$$
$$\mathcal{Z}_3 = \{-0.598555,\ 0.474890,\ -0.502433 \pm 0.914034\mathrm{i}\}, \tag{23}$$
$$\mathcal{Z}_4 = \{-1.272319,\ -0.907339,\ -0.046490 \pm 0.594179\mathrm{i}\}, \tag{24}$$

and the resulting pole sets are

$$\mathcal{P}_1 = \{0,\ \pm 0.994400\mathrm{i},\ 0.976368 \pm 0.179176\mathrm{i}\}, \tag{25}$$
$$\mathcal{P}_2 = \{0,\ -0.990691,\ -0.461592,\ 0.026116 \pm 0.966347\mathrm{i}\}, \tag{26}$$
$$\mathcal{P}_3 = \{0,\ -0.761948,\ 0.718261,\ -0.493190 \pm 0.848475\mathrm{i}\}, \tag{27}$$
$$\mathcal{P}_4 = \{0,\ -0.932245,\ 0.838485,\ -0.496357 \pm 0.839896\mathrm{i}\}. \tag{28}$$

Each channel therefore contains a complex-conjugate pole pair, ensuring that the constructed true transfer function contains poles not constrained to the real axis, and every pole lies strictly inside the unit circle, with $\max_{z\in\mathcal{P}_c}|z|$ approximately equal to $0.994400$, $0.990691$, $0.981400$ and $0.975600$ for the four channels.

**Target Signal.** We generate the target signal by applying the true transfer function to the input signal, such that $Y_c(z) = H_c(z)X_c(z)$. Here, the input is zero outside $1 \le t \le T$, and $Y_c(z)$ transforms the full causal zero-state response, whose first $T$ samples form the target sequence. Equivalently,

$$Y_c(z) = z^{-1}\left(\sum_{j=0}^{4} b_{c,j}z^{-j}\right)X_c(z) + \left(\sum_{j=1}^{4} a_{c,j}z^{-j}\right)Y_c(z). \tag{29}$$

The corresponding direct causal recurrence is

$$y_{t,c} = \sum_{j=0}^{4} b_{c,j}x_{t-1-j,c} + \sum_{j=1}^{4} a_{c,j}y_{t-j,c}, \qquad y_{t,c} = x_{t,c} = 0 \quad \text{for } t \le 0. \tag{30}$$

### A.2 Constructing Cayley Map

For any skew-symmetric $S$, the matrix $I - S$ is invertible and the Cayley map (Cayley, 1846)

$$R = (I + S)(I - S)^{-1} \tag{31}$$

is a rotation. Orthogonality follows from $S^{\mathsf{T}} = -S$ together with the commutation of $I + S$ and $I - S$,

$$R^{\mathsf{T}}R = (I - S)^{-\mathsf{T}}(I + S)^{\mathsf{T}}(I + S)(I - S)^{-1} = (I + S)^{-1}(I - S)(I + S)(I - S)^{-1} = I,$$

and $\det R = 1$ by continuity from $S = 0$. The map therefore parameterizes rotations without trigonometric evaluation, and its image is the set of rotations that do not have $-1$ as an eigenvalue.

In two dimensions every skew-symmetric matrix carries a single free parameter,

$$S = \tau J, \qquad J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad S = \begin{pmatrix} 0 & -\tau \\ \tau & 0 \end{pmatrix},$$

so that a planar rotation is determined by $\tau$ alone. Evaluating equation (31) with $\det(I - S) = 1+\tau^2$ gives

$$R = \frac{1}{1+\tau^2}\begin{pmatrix} 1 & -\tau \\ \tau & 1 \end{pmatrix}\begin{pmatrix} 1 & -\tau \\ \tau & 1 \end{pmatrix} = \frac{1}{1+\tau^2}\begin{pmatrix} 1-\tau^2 & -2\tau \\ 2\tau & 1-\tau^2 \end{pmatrix},$$

and therefore

$$\cos\theta = \frac{1-\tau^2}{1+\tau^2}, \qquad \sin\theta = \frac{2\tau}{1+\tau^2}, \qquad \tau = \tan\frac{\theta}{2}, \tag{32}$$

which is the general Cayley parameterization of a planar rotation.

The reflection coefficient matrix of MIRA is the special case in which $\tau$ is taken to be the Möbius transform (Möbius, 1855) of the projected coefficient,

$$\tau_t^{(k)} = \frac{1 - \alpha_t^{(k)}}{1 + \alpha_t^{(k)}}.$$

Substituting this into equation (32), with $1 - \tau^2 = {}^{4\alpha}/_{(1+\alpha)^2}$ and $1 + \tau^2 = {}^{2(1+\alpha^2)}/_{(1+\alpha)^2}$, yields

$$\cos\left(\theta_t^{(k)}\right) = \frac{2\alpha_t^{(k)}}{1 + \left(\alpha_t^{(k)}\right)^2}, \qquad \sin\left(\theta_t^{(k)}\right) = \frac{1 - \left(\alpha_t^{(k)}\right)^2}{1 + \left(\alpha_t^{(k)}\right)^2},$$

the parameterization used in equation (2). The rotation angle is thus obtained from $\alpha_t^{(k)}$ by rational arithmetic alone, requiring neither a trigonometric function nor an arctangent, and the composition of the Cayley map with the Möbius transform is what exchanges the roles of $\sin$ and $\cos$ relative to the parameterization in $\tau$.

### A.3 Weight Initialization

**Characteristic Horizon.** Consider a stable continuous-time oscillatory mode

$$\frac{\mathrm{d}V(\tau)}{\mathrm{d}\tau} = \left(-\frac{1}{H} + \mathrm{i}\omega\right)V(\tau). \tag{33}$$

Its evolution over one token interval is

$$V(\tau + 1) = \exp\left(-\frac{1}{H}\right)\exp(\mathrm{i}\omega)\,V(\tau). \tag{34}$$

Therefore, the corresponding discrete-time pole is $\lambda = q\exp(\mathrm{i}\omega)$, where its radius and horizon satisfy

$$q = \exp\left(-\frac{1}{H}\right), \qquad H = -\frac{1}{\log q}. \tag{35}$$

Here $q$ denotes the radius of an assembled recurrent pole rather than the local contraction coefficient $r_c^{(k)}$.

**Schur Initialization.** We define logarithmically spaced horizons

$$H_c \in [10, 2000] \tag{36}$$

and

$$q_c = \exp\left(-\frac{1}{H_c}\right). \tag{37}$$

Let $J = \lfloor K/2 \rfloor$. For each expanded channel $c$, we construct $J$ complex-conjugate target-pole pairs

$$\lambda_{c,j}^{\pm} = q_c^{1+(j-1)/J} e^{\pm \mathrm{i}\pi\xi_j}, \qquad 1 \le j \le J, \tag{38}$$

where $(\xi_j)_{j\ge1}$ is the breadth-first dyadic sequence obtained by first listing $m/4$ for $m \in \{1,2,3\}$ and then, for each $\ell \in \{3,4,\dots\}$, listing $(2m-1)/2^\ell$ for $m \in \{1,\dots,2^{\ell-1}\}$ in increasing order; the first $J$ elements are used for an order-$K$ system. We include the additional real target pole $-q_c$ when $K$ is odd. The corresponding monic polynomial $P_{c,K}(z) = \sum_{\ell=0}^{K} p_{c,K,\ell} z^{K-\ell}$ is converted into reflection coefficients through the Schur step-down recursion

$$\kappa_c^{(k)} = p_{c,k,k}, \qquad P_{c,k-1}(z) = \frac{P_{c,k}(z) - \kappa_c^{(k)} z^k P_{c,k}(z^{-1})}{\left(1 - \left(\kappa_c^{(k)}\right)^2\right) z}, \qquad 1 \le k \le K. \tag{39}$$

We initialize

$$r_c^{(k)} = 1 - 2^{-13}\left(1 - \left|\kappa_c^{(k)}\right|\right), \tag{40}$$

and initialize the biases of the $\alpha$ and $\beta$ projections as

$$b_{\alpha,c}^{(k)} = \epsilon_{\alpha,c}^{(k)} \sqrt{\frac{1 - \kappa_c^{(k)}/r_c^{(k)}}{1 + \kappa_c^{(k)}/r_c^{(k)}}}, \qquad b_{\beta,c}^{(k)} = \epsilon_{\beta,c}^{(k)} \sqrt{\frac{1}{r_c^{(k)}} - 1}, \tag{41}$$

where $\epsilon_{\alpha,c}^{(k)}, \epsilon_{\beta,c}^{(k)} \sim \{-1,+1\}$ are independent random signs.

**Initialization Distribution.** Each entry $W_{i,j}$ in the corresponding row of $W_\alpha$ or $W_\beta$ associated with bias $b_i$ is initialized independently as

$$W_{i,j} = \epsilon u, \qquad \epsilon \sim \{-1,+1\}, \qquad u \sim \mathcal{U}\left(\frac{s(b_i)}{4}, s(b_i)\right), \tag{42}$$

where

$$s(b_i) = \min\left\{\frac{(K+1)^{-1/4}}{\sqrt{d_r}}, \frac{\max\{|b_i|, 10^{-6}\}}{4\sqrt{d_r}}\right\}. \tag{43}$$

Thus, $b_\alpha$ and $b_\beta$ establish the structured initial pole geometry, while the nonzero uniformly initialized $W_\alpha$ and $W_\beta$ provide input-conditioned perturbations around that geometry.

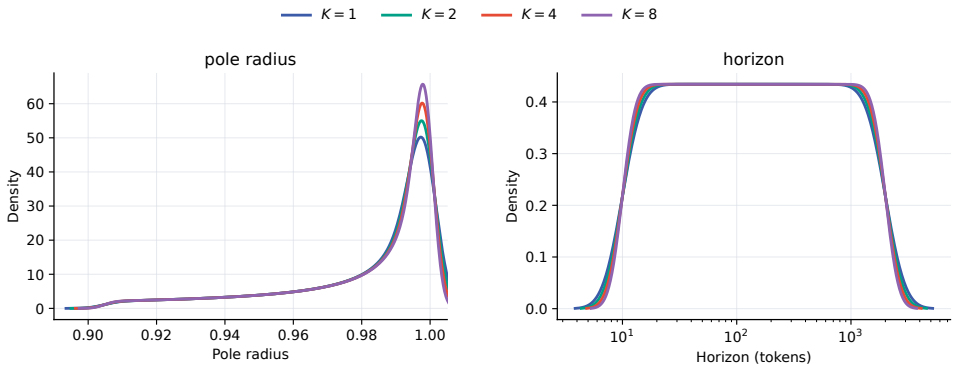


Figure 8: Distribution of pole radius and horizon.

### A.4 Poles of Recurrent Systems

We characterize the poles induced by the recurrent dynamics of each sequence mixer. For methods with input-dependent coefficients, we fix the coefficients at token position $t$ and analyze the resulting frozen system. The denominators below are unreduced; actual transfer poles may be removed by cancellation.

**Theorem 1** (Poles of First-Order Systems)**.** *Let $x_t \in \mathbb{R}^{d_x}$, $h_t \in \mathbb{R}^n$, and $y_t \in \mathbb{R}^{d_y}$ denote the input, recurrent state, and output at token position $t$, respectively. Consider the first-order system*

$$h_t = Ah_{t-1} + Bx_t, \qquad y_t = Ch_t + Dx_t, \tag{44}$$

*where $A \in \mathbb{R}^{n\times n}$ is the recurrent transition matrix and $B$, $C$, and $D$ are the input, readout, and feedthrough matrices of compatible dimensions. With $h_0 = 0$, the corresponding transfer function is*

$$H(z) = \frac{A(z)}{B(z)}, \tag{45}$$

*where*

$$A(z) = C\,\mathrm{adj}\big(I - Az^{-1}\big)B + D\det\big(I - Az^{-1}\big), \tag{46}$$

$$B(z) = \det\big(I - Az^{-1}\big) = \prod_{j=1}^{n}\big(1 - \lambda_j z^{-1}\big). \tag{47}$$

*Here, $z \in \mathbb{C}$ is the $z$-transform variable, $I \in \mathbb{R}^{n\times n}$ is the identity matrix, $\mathrm{adj}(\cdot)$ and $\det(\cdot)$ denote the adjugate and determinant, respectively, and $\lambda_1, \ldots, \lambda_n \in \mathrm{spec}(A)$ are the eigenvalues of $A$. The finite poles of the transfer function form a subset of $\mathrm{spec}(A)$, after clearing negative powers of $z$ and cancelling common factors.*

*Proof.* Let $X(z)$ and $H_z(z)$ denote the $z$-transforms of the input sequence and recurrent-state sequence, respectively. Taking the $z$-transform of the state equation gives

$$H_z(z) = Az^{-1}H_z(z) + BX(z), \tag{48}$$

and therefore

$$H_z(z) = \big(I - Az^{-1}\big)^{-1}BX(z). \tag{49}$$

Substitution into the output equation yields

$$H(z) = C\big(I - Az^{-1}\big)^{-1}B + D = zC(zI - A)^{-1}B + D. \tag{50}$$

Using

$$\big(I - Az^{-1}\big)^{-1} = \frac{\mathrm{adj}\big(I - Az^{-1}\big)}{\det\big(I - Az^{-1}\big)} \tag{51}$$

gives the stated numerator and denominator. Moreover,

$$\det\big(I - Az^{-1}\big) = z^{-n}\det(zI - A) = \prod_{j=1}^{n}\big(1 - \lambda_j z^{-1}\big), \tag{52}$$

Thus, $H(z) = \big[zC\,\mathrm{adj}(zI - A)B + D\det(zI - A)\big]/\det(zI - A)$, so every finite transfer pole, including any pole at the origin, is an eigenvalue of $A$; numerator cancellation can remove such poles. □

For token-dependent coefficients evaluated along the input sequence, $h_t = A_t h_{t-1} + B_t x_t$, $y_t = C_t h_t + D_t x_t$, and $h_0 = 0$, repeated substitution gives

$$y_t = D_t x_t + \sum_{\tau=1}^{t} C_t\,(A_t \cdots A_{\tau+1})\,B_\tau x_\tau, \tag{53}$$

where $\tau$ indexes the input position and the product is $I$ for $\tau = t$. This exact response retains the coefficient history. When the coefficients are held constant, the product becomes $A^{t-\tau}$ and the rational transfer function above applies.

**Corollary 1** (Symmetric Transitions Imply Real Poles)**.** *If $A$ is symmetric, or if there exists a positive-definite diagonal matrix $D_s$ such that $D_s A$ is symmetric, then every pole of the corresponding recurrent system is real.*

*Proof.* If $A$ is symmetric, the spectral theorem gives

$$\mathrm{spec}(A) \subset \mathbb{R}. \tag{54}$$

If $D_s \succ 0$ and $D_s A$ is symmetric, then

$$D_s^{1/2} A D_s^{-1/2} = D_s^{-1/2}(D_s A) D_s^{-1/2} \tag{55}$$

is symmetric. Since similar matrices have the same eigenvalues, $A$ also has a real spectrum. The result then follows from Theorem 1. □

### A.4.1 Scaled Dot-Product Attention

Let $q_t, k_t \in \mathbb{R}^d$ and $v_t \in \mathbb{R}^{d_v}$ denote the query, key, and value at token position $t$, where $d$ is the query–key dimension and $d_v$ is the value dimension. Causal SDPA computes

$$y_t = \sum_{s=1}^{t} \frac{\exp\big(q_t^\mathsf{T} k_s/\sqrt{d}\big)}{\sum_{s'=1}^{t}\exp\big(q_t^\mathsf{T} k_{s'}/\sqrt{d}\big)} v_s, \tag{56}$$

where $s$ and $s'$ index preceding token positions. An exact recurrent representation retains the preceding keys and values,

$$h_t = (k_1, v_1, \ldots, k_t, v_t), \tag{57}$$

whose dimension grows with $t$. In particular, the softmax kernel does not admit an exact finite-dimensional factorization because

$$\exp\big(q^\mathsf{T} k/\sqrt{d}\big) = \sum_{p=0}^{\infty} \frac{(q^\mathsf{T} k)^p}{p!\,d^{p/2}}, \tag{58}$$

where $p$ indexes the polynomial degree. The kernel has infinite feature rank: even for scalar queries and keys, the functions $q \mapsto e^{qk_j}$ for arbitrarily many distinct keys $k_j$ are linearly independent. Thus, unrestricted SDPA has no exact fixed finite-dimensional linear feature-state representation and no corresponding fixed finite-order rational LTI transfer function. At a fixed finite context with attention weights frozen, its output is instead a finite weighted sum of delayed values.

### A.4.2 LINEAR ATTENTION

Let $\phi : \mathbb{R}^d \to \mathbb{R}^{d_\phi}$ denote a finite-dimensional feature map, where $d_\phi$ is the feature dimension. Linear attention maintains the matrix state $S_t \in \mathbb{R}^{d_\phi \times d_v}$ and normalization state $z_t \in \mathbb{R}^{d_\phi}$:

$$S_t = S_{t-1} + \phi(k_t)v_t^\mathsf{T}, \tag{59}$$

$$z_t = z_{t-1} + \phi(k_t), \tag{60}$$

$$y_t = \frac{\phi(q_t)^\mathsf{T} S_t}{\phi(q_t)^\mathsf{T} z_t}. \tag{61}$$

Using $\mathrm{vec}(\cdot)$ to denote column-wise matrix vectorization and $\otimes$ to denote the Kronecker product, the stacked state

$$h_t = \begin{pmatrix} \mathrm{vec}(S_t) \\ z_t \end{pmatrix} \in \mathbb{R}^{d_\phi d_v + d_\phi} \tag{62}$$

satisfies

$$h_t = \begin{pmatrix} I_{d_\phi d_v} & 0 \\ 0 & I_{d_\phi} \end{pmatrix} h_{t-1} + \begin{pmatrix} v_t \otimes \phi(k_t) \\ \phi(k_t) \end{pmatrix}, \tag{63}$$

where $I_m$ denotes the $m \times m$ identity matrix. The state transition is therefore the identity matrix, with denominator

$$B(z) = \left(1 - z^{-1}\right)^{d_\phi d_v + d_\phi}. \tag{64}$$

Thus, all poles of the recurrent state lie at $z = 1$. The normalized output remains a ratio of two linear functionals of the state; without the normalizer and with query and key coefficients frozen, the map from values to outputs is a first-order linear system.

### A.4.3 SELECTIVE STATE SPACE MODELS

Let $u_{t,d} \in \mathbb{R}$ and $y_{t,d} \in \mathbb{R}$ denote the input and output at token position $t$ and channel $d$. Let $h_t \in \mathbb{R}^N$ denote the corresponding recurrent state, where $N$ is the state dimension. A selective SSM has the frozen recurrent dynamics

$$h_t = \mathrm{diag}\left(e^{\Delta_{t,d} A_{d,1}}, \ldots, e^{\Delta_{t,d} A_{d,N}}\right) h_{t-1} + \Delta_{t,d} B_t u_{t,d}, \tag{65}$$

where $\Delta_{t,d} > 0$ is the input-dependent step size, $A_{d,n} < 0$ is the continuous-time transition coefficient of state $n$, and $B_t \in \mathbb{R}^N$ is the input coefficient. Let $C_t \in \mathbb{R}^N$ and $D_d \in \mathbb{R}$ denote the readout and feedthrough coefficients, respectively. The corresponding transfer function is

$$H(z) = \sum_{n=1}^{N} \frac{\Delta B_n C_n}{1 - \overline{a}_n z^{-1}} + D, \qquad \overline{a}_n = e^{\Delta a_n}, \tag{66}$$

with denominator

$$B(z) = \prod_{n=1}^{N} \left(1 - \overline{a}_n z^{-1}\right). \tag{67}$$

The frozen transition matrix is diagonal and therefore symmetric. By Corollary 1, all poles are real. Since $\Delta > 0$ and $a_n < 0$,

$$\overline{a}_n = e^{\Delta a_n} \in (0, 1), \tag{68}$$

so every pole lies on the positive real axis inside the unit circle.

### A.4.4 RWKV-6

For each head, let $S_t \in \mathbb{R}^{S \times S}$ denote the recurrent state and let $r_t, k_t, v_t, w_t \in \mathbb{R}^S$ denote the receptance, key, value, and decay vectors, respectively, where $S$ is the head dimension and $w_t \in (0, 1)^S$. RWKV-6 updates

$$S_t = \mathrm{diag}(w_t) S_{t-1} + k_t v_t^\mathsf{T}. \tag{69}$$

Its readout is

$$y_t = r_t^\mathsf{T} \left(\mathrm{diag}(u) k_t v_t^\mathsf{T} + S_{t-1}\right), \tag{70}$$

where $u \in \mathbb{R}^S$ denotes the time-mixing bonus coefficient. Since $\mathrm{diag}(w_t)$ is invertible,

$$S_{t-1} = \mathrm{diag}(w_t)^{-1} \left(S_t - k_t v_t^\mathsf{T}\right), \tag{71}$$

and therefore

$$y_t = r_t^\mathsf{T} \operatorname{diag}(1/w_t) S_t + \left[ r_t^\mathsf{T} \big( (u - 1/w_t) \odot k_t \big) \right] v_t^\mathsf{T}, \tag{72}$$

where $\odot$ denotes element-wise multiplication. For each state column, the transition matrix is

$$A_t = \operatorname{diag}(w_t). \tag{73}$$

Its denominator polynomial is

$$B(z) = \prod_{j=1}^{S} \big(1 - w_{t,j} z^{-1}\big). \tag{74}$$

Because $A_t$ is diagonal, all poles are real and satisfy

$$w_{t,j} \in (0, 1). \tag{75}$$

### A.4.5 RWKV-7

Let $S_t \in \mathbb{R}^{N \times N}$ denote the recurrent state, where $N$ is the head dimension. Let $v_t, k_t, r_t \in \mathbb{R}^N$ denote the value, key, and receptance vectors; let $w_t \in (0,1)^N$ denote the decay vector; let $a_t \in (0,1)^N$ denote the learning-rate vector; and let $\kappa_t \in \mathbb{R}^N$ denote a normalized key direction satisfying $\|\kappa_t\|_2 = 1$. RWKV-7 updates

$$S_t = S_{t-1} \operatorname{diag}(w_t) - S_{t-1} \kappa_t \big(a_t \odot \kappa_t\big)^\mathsf{T} + v_t k_t^\mathsf{T}. \tag{76}$$

Factoring the state terms gives

$$S_t = S_{t-1} \left[ \operatorname{diag}(w_t) - \kappa_t \kappa_t^\mathsf{T} \operatorname{diag}(a_t) \right] + v_t k_t^\mathsf{T}. \tag{77}$$

After transposition, the left-acting transition matrix is

$$A_t = \operatorname{diag}(w_t) - \operatorname{diag}(a_t) \kappa_t \kappa_t^\mathsf{T}. \tag{78}$$

The matrix determinant lemma gives

$$B(z) = \prod_{j=1}^{N} \big(1 - w_{t,j} z^{-1}\big) \left[ 1 + z^{-1} \sum_{j=1}^{N} \frac{a_{t,j} \kappa_{t,j}^2}{1 - w_{t,j} z^{-1}} \right]. \tag{79}$$

Although $A_t$ is not generally symmetric, define the positive-definite diagonal matrix

$$D_s = \operatorname{diag}(a_t^{-1}). \tag{80}$$

Then

$$D_s A_t = \operatorname{diag}(w_t / a_t) - \kappa_t \kappa_t^\mathsf{T} \tag{81}$$

is symmetric. Equivalently,

$$D_s^{1/2} A_t D_s^{-1/2} = \operatorname{diag}(w_t) - \big(a_t^{1/2} \odot \kappa_t\big) \big(a_t^{1/2} \odot \kappa_t\big)^\mathsf{T} \tag{82}$$

is symmetric. Corollary 1 therefore guarantees that all poles of RWKV-7 are real.

### A.4.6 RETNET

For each head, let $S_t \in \mathbb{C}^{d_k \times d_v}$ denote the recurrent state, where $d_k$ and $d_v$ are the key and value dimensions, respectively. Let $q_t, k_t \in \mathbb{C}^{d_k}$ and $v_t \in \mathbb{R}^{d_v}$ denote the query, key, and value at token position $t$. In the original RetNet formulation, the positional rotations are incorporated into the query and key:

$$q_t = (x_t W_Q) \odot \Theta_t, \qquad k_t = (x_t W_K) \odot \overline{\Theta}_t, \tag{83}$$

where $\Theta_t$ denotes the complex positional rotation at token position $t$ and $\overline{\Theta}_t$ denotes its complex conjugate. The recurrent representation is

$$S_t = \gamma S_{t-1} + k_t v_t^\mathsf{T}, \qquad y_t = q_t^\mathsf{T} S_t, \tag{84}$$

where $\gamma \in (0,1)$ is the head-specific decay coefficient.

The recurrent transition matrix is $\gamma I$, where $I$ denotes the identity matrix. Freezing the complete query and key coefficients, including their positional rotations, the corresponding transfer function has the denominator

$$B(z) = 1 - \gamma z^{-1}. \tag{85}$$

Therefore, the only possible pole of each frozen RetNet head is

$$z = \gamma \in (0, 1). \tag{86}$$

For multi-scale retention with head-specific decay coefficients $\gamma_h$, the candidate pole set is

$$\mathcal{P} = \{\gamma_h\}_{h=1}^{H}, \tag{87}$$

where $H$ is the number of heads. Thus, all poles of this frozen representation are real and lie strictly inside the unit circle. This conclusion does not apply when the positional rotations retain their temporal progression, which can produce complex lag factors.

### A.4.7 HGRN2

Let $S_t \in \mathbb{R}^{d \times d_v}$ denote the recurrent state, where $d$ is the feature dimension and $d_v$ is the value dimension. Let $f_t \in (0, 1)^d$ denote the forget gate, $i_t \in \mathbb{R}^{d_v}$ the input value, and $o_t \in \mathbb{R}^d$ the output vector. HGRN2 updates

$$S_t = \mathrm{diag}(f_t) S_{t-1} + (1 - f_t) i_t^{\mathsf{T}}, \tag{88}$$

with readout

$$y_t = o_t^{\mathsf{T}} S_t. \tag{89}$$

For each state column, the transition matrix is

$$A_t = \mathrm{diag}(f_t). \tag{90}$$

The corresponding denominator is

$$B(z) = \prod_{j=1}^{d} \big(1 - f_{t,j} z^{-1}\big). \tag{91}$$

The transition matrix is diagonal, and its eigenvalues are

$$z = f_{t,j} \in (0, 1). \tag{92}$$

Thus, all poles of HGRN2 are real and lie strictly inside the unit circle.

### A.4.8 HYENA

Let $u_t$ and $y_t$ denote the input and output of a Hyena convolutional stage. The stage applies a finite causal convolution

$$y_t = \sum_{n=0}^{L-1} \kappa_n u_{t-n} + D u_t, \tag{93}$$

where $L$ is the filter length, $\kappa_n$ is the filter coefficient at lag $n$, and $D$ is the feedthrough coefficient. The filter coefficients are generated from position according to

$$\kappa_n = \mathrm{MLP}_{\sin}\big(z(n)\big) \odot e^{-n|\delta|}, \tag{94}$$

where $z(n)$ is the positional embedding at lag $n$, $\mathrm{MLP}_{\sin}$ is the sinusoidal filter-generating network, and $\delta$ controls exponential decay. Introducing the shift-register state

$$h_t = (u_t, u_{t-1}, \ldots, u_{t-L+1})^{\mathsf{T}} \in \mathbb{R}^L \tag{95}$$

gives

$$h_t = S_{\downarrow} h_{t-1} + e_1 u_t, \tag{96}$$

where $S_{\downarrow} \in \mathbb{R}^{L \times L}$ is the down-shift matrix and $e_1 \in \mathbb{R}^L$ is the first standard basis vector. The transfer function is

$$H(z) = D + \sum_{n=0}^{L-1} \kappa_n z^{-n}, \tag{97}$$

whose denominator is

$$B(z) = 1. \tag{98}$$

Therefore, a Hyena convolutional stage is an FIR system and has no nonzero finite poles.

### A.4.9 DELTANET AND GATED DELTANET

Let $S_t \in \mathbb{R}^{d_k \times d_v}$ denote the recurrent state, where $d_k$ and $d_v$ are the key and value dimensions. Let $k_t \in \mathbb{R}^{d_k}$ and $v_t \in \mathbb{R}^{d_v}$ denote the normalized key and value, with $\|k_t\|_2 = 1$. Let $\beta_t \in (0,1)$ denote the update rate and $\alpha_t \in (0,1)$ denote the decay coefficient for Gated DeltaNet; for DeltaNet, $\alpha_t = 1$. The dynamics are

$$S'_t = \alpha_t S_{t-1}, \tag{99}$$

$$\widetilde{v}_t = \beta_t \left(v_t - S_t'^{\mathsf{T}} k_t\right), \tag{100}$$

$$S_t = S'_t + k_t \widetilde{v}_t^{\mathsf{T}}, \tag{101}$$

where $S'_t$ is the decayed state and $\widetilde{v}_t \in \mathbb{R}^{d_v}$ is the delta-rule update. Substitution gives

$$\begin{aligned} S_t &= \alpha_t S_{t-1} + \beta_t k_t v_t^{\mathsf{T}} - \alpha_t \beta_t k_t k_t^{\mathsf{T}} S_{t-1} \\ &= \alpha_t \left(I_{d_k} - \beta_t k_t k_t^{\mathsf{T}}\right) S_{t-1} + \beta_t k_t v_t^{\mathsf{T}}. \end{aligned} \tag{102}$$

The transition matrix is therefore

$$A_t = \alpha_t \left(I_{d_k} - \beta_t k_t k_t^{\mathsf{T}}\right), \tag{103}$$

with denominator

$$B(z) = \left(1 - \alpha_t z^{-1}\right)^{d_k - 1} \left(1 - \alpha_t (1 - \beta_t) z^{-1}\right). \tag{104}$$

Since

$$A_t k_t = \alpha_t (1 - \beta_t) k_t, \tag{105}$$

$$A_t z = \alpha_t z, \qquad z \perp k_t, \tag{106}$$

where $z \in \mathbb{R}^{d_k}$ denotes any vector orthogonal to $k_t$, the spectrum is

$$\operatorname{spec}(A_t) = \{\alpha_t (1 - \beta_t)\} \cup \{\alpha_t\}^{d_k - 1}. \tag{107}$$

The transition matrix is symmetric, so Corollary 1 guarantees that all poles are real.

## A.5 TRANSFER FUNCTION OF LATTICE–LADDER NETWORK

This section derives the transfer function of the parameterized lattice–ladder network. We consider a single expanded feature channel $\ell \in \{1, \ldots, d_e\}$ and omit the channel index for clarity. For the frozen-system analysis at token position $t$, we treat the input-conditioned reflection angles $\theta_{t,\ell}^{(k)}$, radius coefficients $r_{t,\ell}^{(k)}$, readout coefficients $\rho_{t,\ell}^{(k)}$, memory write projection $s_t$, and memory readout projections $\nu_t^{(k)}$ as constants, denoted by $\theta^{(k)}$, $r^{(k)}$, $\rho^{(k)}$, $s$, and $\nu^{(k)}$, respectively. Since the lattice recursion acts independently across expanded feature channels, the derivation applies to every channel.

Let $K$ denote the lattice order and $k \in \{1, \ldots, K\}$ denote the lattice stage. For token positions $t \in \{1, \ldots, T\}$, the lattice recursion is

$$U_t^{(k-1)} = r^{(k)} \left[\cos\left(\theta^{(k)}\right) U_t^{(k)} - \sin\left(\theta^{(k)}\right) V_{t-1}^{(k-1)}\right], \tag{108}$$

$$V_t^{(k)} = r^{(k)} \left[\sin\left(\theta^{(k)}\right) U_t^{(k)} + \cos\left(\theta^{(k)}\right) V_{t-1}^{(k-1)}\right], \tag{109}$$

with the boundary conditions

$$U_t^{(K)} = x_t s^{\mathsf{T}}, \qquad V_t^{(0)} = U_t^{(0)}, \qquad V_0^{(k)} = 0, \quad k \in \{0, \ldots, K-1\}. \tag{110}$$

Here, $x_t$ denotes the input signal, $U_t^{(k)}$ denotes the forward signal at stage $k$, and $V_t^{(k)}$ denotes the recurrent state at stage $k$. The ladder readout is

$$y_t = \sum_{k=0}^{K} \rho^{(k)} V_t^{(k)} \nu^{(k)}, \tag{111}$$

where $y_t$ denotes the output signal.

### A.5.1 Polynomial Recursion of the Lattice Stages

Let $X(z), U^{(k)}(z)$, and $V^{(k)}(z)$ denote the $z$-transforms of $x_t$, $U_t^{(k)}$, and $V_t^{(k)}$, respectively, with the input zero outside $1 \le t \le T$ and the frozen recursion continued causally beyond $T$. Ratios of the row-vector signals below denote scalar response factors, not vector division. Under the zero-state condition, taking the $z$-transform of the lattice recursion gives

$$U^{(k-1)}(z) = r^{(k)} \left[\cos\left(\theta^{(k)}\right) U^{(k)}(z) - \sin\left(\theta^{(k)}\right) z^{-1} V^{(k-1)}(z)\right], \tag{112}$$

$$V^{(k)}(z) = r^{(k)} \left[\sin\left(\theta^{(k)}\right) U^{(k)}(z) + \cos\left(\theta^{(k)}\right) z^{-1} V^{(k-1)}(z)\right]. \tag{113}$$

Assuming $\cos\left(\theta^{(k)}\right) \neq 0$ at every stage for the intermediate divisions, rearranging the first equation and substituting it into the second equation gives

$$\begin{aligned} U^{(k)}(z) &= \frac{1}{r^{(k)} \cos\left(\theta^{(k)}\right)} U^{(k-1)}(z) + \tan\left(\theta^{(k)}\right) z^{-1} V^{(k-1)}(z), \\ V^{(k)}(z) &= \tan\left(\theta^{(k)}\right) U^{(k-1)}(z) + \frac{r^{(k)}}{\cos\left(\theta^{(k)}\right)} z^{-1} V^{(k-1)}(z). \end{aligned} \tag{114}$$

Define the normalized stage polynomial $P_k(z)$ and ladder-basis polynomial $Q_k(z)$ by

$$\begin{aligned} P_0(z) &= 1, \qquad Q_0(z) = \prod_{i=1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right), \\ P_k(z) &= P_{k-1}(z) + \frac{r^{(k)} \sin\left(\theta^{(k)}\right)}{\prod_{i=k}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)} z^{-1} Q_{k-1}(z), \\ Q_k(z) &= r^{(k)} \sin\left(\theta^{(k)}\right) \prod_{i=k+1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right) P_{k-1}(z) + \frac{r^{(k)}}{\cos\left(\theta^{(k)}\right)} z^{-1} Q_{k-1}(z). \end{aligned} \tag{115}$$

Inductively, $Q_{k-1}(z)$ contains the factor $\prod_{i=k}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)$. Cancelling this factor in the recursion for $P_k$ and its $k$-th factor in the recursion for $Q_k$ makes every coefficient polynomial in $r^{(i)} \sin\left(\theta^{(i)}\right)$, $(r^{(i)})^2$, and $r^{(i)} \cos\left(\theta^{(i)}\right)$. The final polynomial transfer formula therefore extends continuously to parameters with $\cos\left(\theta^{(k)}\right) = 0$. We first establish the stage-response identities

$$\begin{aligned} \frac{U^{(k)}(z)}{U^{(0)}(z)} &= \frac{P_k(z)}{\prod_{i=1}^{k} r^{(i)} \cos\left(\theta^{(i)}\right)}, \\ \frac{V^{(k)}(z)}{U^{(0)}(z)} &= \frac{Q_k(z)}{\prod_{i=1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)}. \end{aligned} \tag{116}$$

For $k = 0$, the identity follows immediately from $V^{(0)}(z) = U^{(0)}(z)$, $P_0(z) = 1$, and $Q_0(z) = \prod_{i=1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)$. Suppose that it holds at stage $k-1$. Substituting equation (116) into the first equation of equation (114) gives

$$\begin{aligned} \frac{U^{(k)}(z)}{U^{(0)}(z)} &= \frac{P_{k-1}(z)}{\prod_{i=1}^{k} r^{(i)} \cos\left(\theta^{(i)}\right)} + \frac{\tan\left(\theta^{(k)}\right) z^{-1} Q_{k-1}(z)}{\prod_{i=1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)} \\ &= \frac{P_k(z)}{\prod_{i=1}^{k} r^{(i)} \cos\left(\theta^{(i)}\right)}. \end{aligned} \tag{117}$$

Similarly, the second equation of equation (114) gives

$$\begin{aligned}\frac{V^{(k)}(z)}{U^{(0)}(z)} &= \frac{\tan\left(\theta^{(k)}\right) P_{k-1}(z)}{\prod_{i=1}^{k-1} r^{(i)} \cos\left(\theta^{(i)}\right)} + \frac{r^{(k)} z^{-1} Q_{k-1}(z)}{\cos\left(\theta^{(k)}\right) \prod_{i=1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)} \\ &= \frac{Q_k(z)}{\prod_{i=1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)},\end{aligned} \tag{118}$$

which proves equation (116) for every $k \in \{1, \ldots, K\}$.

Write the order-$k$ stage polynomials as

$$P_k(z) = \sum_{j=0}^{k} p_{k,j} z^{-j}, \qquad Q_k(z) = \sum_{j=0}^{k} q_{k,j} z^{-j}, \tag{119}$$

where $p_{k,j}$ and $q_{k,j}$ denote the coefficients of $z^{-j}$. The initial coefficients are

$$p_{0,0} = 1, \qquad q_{0,0} = \prod_{i=1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right). \tag{120}$$

Matching coefficients in equation (115) gives

$$\begin{aligned} p_{k,j} &= \begin{cases} 1, & j = 0, \\ p_{k-1,j} + \dfrac{r^{(k)} \sin\left(\theta^{(k)}\right)}{\prod_{i=k}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)} q_{k-1,j-1}, & 1 \le j \le k, \end{cases} \\ q_{k,j} &= \begin{cases} r^{(k)} \sin\left(\theta^{(k)}\right) \prod_{i=k+1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right) p_{k-1,0}, & j = 0, \\ r^{(k)} \sin\left(\theta^{(k)}\right) \prod_{i=k+1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right) p_{k-1,j} + \dfrac{r^{(k)}}{\cos\left(\theta^{(k)}\right)} q_{k-1,j-1}, & 1 \le j \le k, \end{cases} \end{aligned} \tag{121}$$

where coefficients whose second index exceeds the polynomial degree are zero. After $K$ lattice stages,

$$P_K(z) = 1 + \sum_{j=1}^{K} p_{K,j} z^{-j}. \tag{122}$$

Since equation (116) gives

$$\frac{U^{(K)}(z)}{U^{(0)}(z)} = \frac{P_K(z)}{\prod_{i=1}^{K} r^{(i)} \cos\left(\theta^{(i)}\right)}, \tag{123}$$

every ladder-stage response relative to $U^{(K)}(z)$ has the common denominator $P_K(z)$. Therefore, the denominator polynomial induced by the lattice recursion is

$$B(z) = P_K(z) = 1 + \sum_{j=1}^{K} b_j z^{-j}, \tag{124}$$

where

$$b_j = p_{K,j}. \tag{125}$$

### A.5.2 Ladder Readout and Numerator Polynomial

From the stage-response identity in equation (116), the intermediate lattice states satisfy

$$V^{(k)}(z) = \frac{Q_k(z)}{P_K(z)} X(z) s^{\mathsf{T}}, \qquad 1 \le k \le K, \tag{126}$$

with

$$V^{(0)}(z) = \frac{Q_0(z)}{P_K(z)} X(z) s^{\mathsf{T}}. \tag{127}$$

Let $Y(z)$ denote the $z$-transform of the output sequence $y_t$. The ladder readout becomes

$$\begin{aligned} Y(z) &= \sum_{k=0}^{K} \rho^{(k)} V^{(k)}(z) \nu^{(k)} \\ &= \frac{\sum_{k=0}^{K} \rho^{(k)} \left( s^{\mathsf{T}} \nu^{(k)} \right) Q_k(z)}{P_K(z)} X(z). \end{aligned} \tag{128}$$

The ladder-basis polynomial satisfies

$$Q_k(z) = P_K(z) \frac{V^{(k)}(z)}{U^{(K)}(z)}, \qquad 0 \le k \le K. \tag{129}$$

Substituting

$$\frac{V^{(k)}(z)}{U^{(K)}(z)} = \frac{Q_k(z)}{P_K(z)} \tag{130}$$

gives

$$Y(z) = \frac{\sum_{k=0}^{K} \rho^{(k)} \left( s^{\mathsf{T}} \nu^{(k)} \right) Q_k(z)}{P_K(z)} X(z). \tag{131}$$

Therefore, the transfer function is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{A(z)}{B(z)}, \tag{132}$$

where the numerator and denominator polynomials are

$$A(z) = \sum_{k=0}^{K} \rho^{(k)} \left( s^{\mathsf{T}} \nu^{(k)} \right) Q_k(z), \tag{133}$$

$$B(z) = P_K(z). \tag{134}$$

Expanding the numerator as

$$A(z) = \sum_{m=0}^{K} a_m z^{-m}, \tag{135}$$

where $a_m$ denotes the coefficient of $z^{-m}$, gives

$$\sum_{m=0}^{K} a_m z^{-m} = \sum_{k=0}^{K} \rho^{(k)} \left( s^{\mathsf{T}} \nu^{(k)} \right) Q_k(z). \tag{136}$$

Hence, the order-$K$ lattice–ladder network realizes the transfer function

$$H(z) = \frac{\sum_{m=0}^{K} a_m z^{-m}}{1 + \sum_{j=1}^{K} b_j z^{-j}}, \tag{137}$$

where the numerator coefficients $a_m$ and denominator coefficients $b_j$ are induced by the reflection angles $\theta^{(1:K)}$, radius coefficients $r^{(1:K)}$, readout coefficients $\rho^{(0:K)}$, memory write projection $s$, and memory readout projections $\nu^{(0:K)}$ through the lattice recursion.

**Complex-Conjugate Poles.** For $K = 2$, take $r^{(1)} = r^{(2)} = 1/2$, $\theta^{(1)} = 0$, and $\theta^{(2)} = \pi/6$. The polynomial recursion gives $P_2(z) = 1 + z^{-2}/16$ and $Q_0(z) = \sqrt{3}/8$. Choosing $\rho^{(0)}(s^\mathsf{T}\nu^{(0)}) = 1$ and the remaining ladder contributions zero yields $H(z) = (\sqrt{3}/8)/(1+z^{-2}/16)$, with uncancelled complex-conjugate poles $z = \pm \mathrm{i}/4$.

**Stability of the Carried State.** For homogeneous propagation, set $U_t^{(K)} = 0$ and let $q = \max_{1\le k\le K} r^{(k)} \le 1$. Orthogonality of each stage rotation gives $\|U_t^{(k-1)}\|_F^2 + \|V_t^{(k)}\|_F^2 \le q^2(\|U_t^{(k)}\|_F^2 + \|V_{t-1}^{(k-1)}\|_F^2)$. Summing over stages and using $V_t^{(0)} = U_t^{(0)}$ yields

$$\sum_{k=0}^{K-1} \|V_t^{(k)}\|_F^2 + \|V_t^{(K)}\|_F^2 + (1-q^2)\sum_{k=1}^{K-1} \|U_t^{(k)}\|_F^2 \le q^2 \sum_{k=0}^{K-1} \|V_{t-1}^{(k)}\|_F^2. \tag{138}$$

Thus, the complete carried-state transition is nonexpansive for $q \le 1$ and Schur stable for $q < 1$, so all poles of the frozen transfer function lie strictly inside the unit circle under the latter sufficient condition. For time-varying coefficients, the same bound holds at each token with $q_t$; a uniform bound $\sup_t q_t < 1$ guarantees uniform exponential decay.

### A.6 Algebraic Structure for Parallelism

We define the transition $(k, t)$ as

$$(k,t) : \big(U_t^{(k)}, V_{t-1}^{(k-1)}\big) \longmapsto \big(U_t^{(k-1)}, V_t^{(k)}\big), \qquad k \in \{K, \dots, 1\}, \quad t \in \{1, \dots, T\}.$$

Its component-wise update is given by equation (2), as illustrated in Figure 3. The transitions within token $t$ form a dependent cascade from $k = K$ to $k = 1$. Since the recurrence acts independently across channels, we define the carried state as

$$\mathbf{V}_t = \big(V_t^{(0)}, \dots, V_t^{(K-1)}\big)^\mathsf{T} \in \mathbb{R}^{Kd_e \times d_m}.$$

For compactness, we define

$$D_t^{(k)} = \operatorname{diag}\left(r_t^{(k)} \odot \cos\left(\theta_t^{(k)}\right)\right), \qquad E_t^{(k)} = \operatorname{diag}\left(r_t^{(k)} \odot \sin\left(\theta_t^{(k)}\right)\right).$$

Let $\mathbf{V}_t = \mathcal{T}_t(\mathbf{V}_{t-1})$ denote the state transition, where $\mathcal{T}_t$ is the corresponding transition operator. We now derive its explicit form using the closure $V_t^{(0)} = U_t^{(0)}$. By equation (2), recursively expanding the forward signal gives

$$U_t^{(k)} = \left(\prod_{i=k+1}^{K} D_t^{(i)}\right) \widetilde{x}_t s_t^\mathsf{T} - \sum_{j=k+1}^{K} \left(\prod_{i=k+1}^{j-1} D_t^{(i)}\right) E_t^{(j)} V_{t-1}^{(j-1)},$$

and substituting $U_t^{(k)}$ into the update of $V_t^{(k)}$ yields

$$V_t^{(k)} = E_t^{(k)} \left(\prod_{i=k+1}^{K} D_t^{(i)}\right) \widetilde{x}_t s_t^\mathsf{T} + D_t^{(k)} V_{t-1}^{(k-1)} - E_t^{(k)} \sum_{j=k+1}^{K} \left(\prod_{i=k+1}^{j-1} D_t^{(i)}\right) E_t^{(j)} V_{t-1}^{(j-1)}.$$

Because $D_t^{(k)}$, $E_t^{(k)}$, $\widetilde{x}_t$, and $s_t$ depend only on $x_t$, the complete cascade $\{(k,t)\}_{k=K}^{1}$ defines the affine operator

$$\mathcal{T}_t = \big(\Phi_t, \mathbf{S}_t\big), \qquad \mathcal{T}_t(\mathbf{V}) = \Phi_t \mathbf{V} + \mathbf{S}_t, \qquad \mathbf{V}_t = \mathcal{T}_t\big(\mathbf{V}_{t-1}\big), \tag{139}$$

where the transition matrix $\Phi_t \in \mathbb{R}^{Kd_e \times Kd_e}$ and input contribution $\mathbf{S}_t \in \mathbb{R}^{Kd_e \times d_m}$ are

$$\Phi_t = \begin{pmatrix} -E_t^{(1)} & \cdots & -\left(\prod_{i=1}^{k-1} D_t^{(i)}\right) E_t^{(k)} & \cdots & -\left(\prod_{i=1}^{K-1} D_t^{(i)}\right) E_t^{(K)} \\ D_t^{(1)} & \cdots & -E_t^{(1)}\left(\prod_{i=2}^{k-1} D_t^{(i)}\right) E_t^{(k)} & \cdots & -E_t^{(1)}\left(\prod_{i=2}^{K-1} D_t^{(i)}\right) E_t^{(K)} \\ \vdots & & \vdots & & \vdots \\ 0_{d_e} & \cdots & D_t^{(k)} & \cdots & -E_t^{(k)}\left(\prod_{i=k+1}^{K-1} D_t^{(i)}\right) E_t^{(K)} \\ \vdots & & \vdots & \ddots & \vdots \\ 0_{d_e} & \cdots & 0_{d_e} & D_t^{(K-1)} & -E_t^{(K-1)} E_t^{(K)} \end{pmatrix},$$

$$\mathbf{S}_t = \begin{pmatrix} \prod_{i=1}^{K} D_t^{(i)} \\ E_t^{(1)} \prod_{i=2}^{K} D_t^{(i)} \\ \vdots \\ E_t^{(k)} \prod_{i=k+1}^{K} D_t^{(i)} \\ \vdots \\ E_t^{(K-1)} D_t^{(K)} \end{pmatrix} \widetilde{x}_t s_t^{\mathsf{T}},$$

respectively. Therefore, for any $t_1 < t_2 < t_3$, temporal composition of the affine operators is associative

$$\begin{aligned} \big(\mathcal{T}_{t_3} \odot \mathcal{T}_{t_2}\big) \odot \mathcal{T}_{t_1} &= (\Phi_{t_3}\Phi_{t_2}\Phi_{t_1}, \Phi_{t_3}\Phi_{t_2}\mathbf{S}_{t_1} + \Phi_{t_3}\mathbf{S}_{t_2} + \mathbf{S}_{t_3}) \\ &= \mathcal{T}_{t_3} \odot \big(\mathcal{T}_{t_2} \odot \mathcal{T}_{t_1}\big). \end{aligned} \tag{140}$$

In general, they are non-commutative:

$$\mathcal{T}_{t_2} \odot \mathcal{T}_{t_1} \neq \mathcal{T}_{t_1} \odot \mathcal{T}_{t_2}.$$

Iterating equation (140), the composition of $\mathcal{T}_s, \ldots, \mathcal{T}_q$ for any $1 \leq s \leq q \leq T$ remains affine,

$$\mathcal{T}_q \odot \cdots \odot \mathcal{T}_s = \big(\Phi_{s\to q}, \mathbf{S}_{s\to q}\big), \qquad \Phi_{s\to q} = \Phi_q \Phi_{q-1} \cdots \Phi_s, \qquad \mathbf{S}_{s\to q} = \sum_{j=s}^{q} \Phi_{j+1\to q}\, \mathbf{S}_j,$$

with the empty product $\Phi_{q+1\to q} = I_{Kd_e}$. This follows by induction on $q - s$. At $q = s$ the composition is $\mathcal{T}_s$ itself, for which $\Phi_{s\to s} = \Phi_s$ and $\mathbf{S}_{s\to s} = \mathbf{S}_s$. Composing $\mathcal{T}_q$ with the hypothesis at $q-1$ gives $\Phi_q \Phi_{s\to q-1} = \Phi_{s\to q}$ together with

$$\Phi_q\, \mathbf{S}_{s\to q-1} + \mathbf{S}_q = \sum_{j=s}^{q-1} \Phi_q \Phi_{j+1\to q-1}\, \mathbf{S}_j + \mathbf{S}_q = \sum_{j=s}^{q} \Phi_{j+1\to q}\, \mathbf{S}_j.$$

Applying this operator to $\mathbf{V}_{s-1}$ gives equation (5), in which $\Phi_{s\to q}$ transports the preceding memory $\mathbf{V}_{s-1}$ across the segment, whereas setting $\mathbf{V}_{s-1} = 0$ identifies $\mathbf{S}_{s\to q}$ as the segment dynamics generated from a zero entry state, independent of all tokens preceding $s$.

### A.7 CUDA Implementation

**Chunk Partition.** The token axis is partitioned into $C$ contiguous chunks, chunk $c$ spanning $t_c, t_c + 1, \ldots, t_c + L_c - 1$ with $t_1 = 1$, $t_{c+1} = t_c + L_c$ and $\sum_{c=1}^{C} L_c = T$. Writing the carried state in stacked form

$$\mathbf{V}_t = \left(V_t^{(0)}, \ldots, V_t^{(K-1)}\right)^{\mathsf{T}} \in \mathbb{R}^{Kd_e \times d_m},$$

each chunk is delimited by an entry state $\mathbf{V}_{t_c-1}$ and an exit state $\mathbf{V}_{t_c+L_c-1}$, related across chunks by $\mathbf{V}_{t_c-1} = \mathbf{V}_{t_{c-1}+L_{c-1}-1}$. This relation orders the chunks strictly, and the following three phases remove it by exploiting the affine structure of equation (3) and its associativity in equation (4).

**Phase (i): Chunk Summarization.** All $C$ chunks are summarized concurrently, each from a zero entry state. With

$$D_t^{(k)} = \operatorname{diag}\left(r_t^{(k)} \odot \cos\left(\theta_t^{(k)}\right)\right), \qquad E_t^{(k)} = \operatorname{diag}\left(r_t^{(k)} \odot \sin\left(\theta_t^{(k)}\right)\right),$$

the token transition of equation (2) is

$$\begin{aligned} U_t^{(K)} &= \widetilde{x}_t s_t^{\mathsf{T}}, \\ U_t^{(k-1)} &= D_t^{(k)} U_t^{(k)} - E_t^{(k)} V_{t-1}^{(k-1)}, \qquad k = K, \dots, 1, \\ V_t^{(k)} &= E_t^{(k)} U_t^{(k)} + D_t^{(k)} V_{t-1}^{(k-1)}, \qquad k = K, \dots, 1, \\ V_t^{(0)} &= U_t^{(0)}, \end{aligned}$$

where $\Phi_t \in \mathbb{R}^{Kd_e \times Kd_e}$ of equation (3) is determined by $D_t^{(1)}, \dots, D_t^{(K)}$ and $E_t^{(1)}, \dots, E_t^{(K)}$. Chunk $c$ retains the pair of equation (5) at $s = t_c$ and $q = t_c + L_c - 1$,

$$\begin{aligned} \Phi_{t_c \to t_c+L_c-1} &= \Phi_{t_c+L_c-1} \Phi_{t_c+L_c-2} \cdots \Phi_{t_c}, \\ \mathbf{S}_{t_c \to t_c+L_c-1} &= \sum_{j=t_c}^{t_c+L_c-1} \Phi_{j+1 \to t_c+L_c-1} \, \mathbf{S}_j = \mathbf{V}_{t_c+L_c-1}|_{\mathbf{V}_{t_c-1}=0}, \end{aligned}$$

the transition being obtained by homogeneous propagation of the $K$ standard basis states and shared across memory components. Since the token transition is affine in $\mathbf{V}$, this pair characterizes the chunk for an arbitrary entry state,

$$\mathbf{V}_{t_c+L_c-1} = \Phi_{t_c \to t_c+L_c-1} \, \mathbf{V}_{t_c-1} + \mathbf{S}_{t_c \to t_c+L_c-1}, \tag{141}$$

so that no chunk requires the output of another. The states traversed under the zero entry are therefore auxiliary for $c \geq 2$ and are not retained.

**Phase (ii): Boundary Composition.** Applying equation (141) for $c = 1, \dots, C$ in token order from the sequence reset $\mathbf{V}_0 = 0$ recovers the entry state of every chunk in $C$ steps. This is the only exchange of information between chunks. It resolves the chunk boundaries alone: no interior state is evaluated, and its cost is independent of $L_c$.

**Phase (iii): Chunk Replay.** The ladder readout is defined at every token position, so the boundary states of phase (ii) are not sufficient and the interior states must be restored. All $C$ chunks are replayed concurrently, chunk $c$ repeating the token transition from the entry state $\mathbf{V}_{t_c-1}$ obtained in phase (ii). This recovers $V_t^{(k)}$ at every token $t$ and every stage $k = 0, \dots, K$ and yields

$$y_t = W_{\text{out}} \left( \left( \sum_{k=0}^{K} \rho_t^{(k)} \odot \left( V_t^{(k)} \nu_t^{(k)} \right) \right) \odot \gamma_t \right), \qquad t = t_c, \dots, t_c + L_c - 1.$$