RICCATI STATE SPACE MODELS: NON-ITERATIVE PARALLELIZATION FOR NONLINEAR SEQUENCE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

State space models (SSMs) achieve efficient sequence processing because their affine state updates are closed under composition and can therefore be evaluated with an associative parallel scan. Nonlinear recurrent models can provide richer, state-dependent dynamics, but generally lose this compositional structure: parallel evaluation then requires iterative methods that repeatedly linearize and scan the recurrence. We ask, what state-dependent nonlinear dynamics can be designed to remain exactly composable? We answer by introducing RiccatiSSM, a nonlinear SSM, in which each state dimension follows an input-conditioned Riccati differential equation. Its quadratic state dependence makes the local Jacobian explicitly state-dependent, while its exact per-step flow under piecewise-constant inputs is a Möbius transformation. Since Möbius maps are closed under composition and compose through 2×2 matrix multiplication, the complete nonlinear state trajectory can be evaluated exactly with a single associative parallel scan, without iterative linearization. We further derive a constrained parameterization that ensures bounded, contractive dynamics, and avoids poles in the fractional-linear state update. Across long-sequence classification, regression, and forecasting tasks, RiccatiSSM achieves competitive predictive performance while reducing runtime by $22-33\%$ compared to the nonlinear LrcSSM under matched architectures. These results demonstrate that state-dependent nonlinear dynamics can retain exact composability and be evaluated efficiently within a single parallel scan.

1 INTRODUCTION

State space models (SSMs) have emerged as efficient sequence models for long-context tasks, combining recurrent state updates with highly parallel training (Gu et al., 2022; Smith et al., 2023; Gu & Dao, 2024). A key source of this efficiency is the structure of their recurrence: although the transition parameters may depend on the input, the state update remains affine in the previous state. Affine maps are closed under composition, allowing an entire sequence to be evaluated using an associative parallel scan with linear work and logarithmic parallel depth.

This computational structure, however, restricts the dynamics that can be represented within the recurrent state update. In particular, the local state Jacobian of an affine recurrence does not depend on the current state. Nonlinear recurrent models can instead make the dynamics themselves state-dependent, allowing their local contraction rates and effective timescales to change along a trajectory. Liquid time-constant neural networks (LTC) provide a prominent example of this behavior by placing the current state inside the continuous-time neuron dynamics (Hasani et al., 2021). Such state dependence increases the flexibility of the dynamics, but in general destroys the compositional structure of the model, that enables a single parallel scan.

Recent methods recover parallelism for nonlinear recurrences by treating sequence evaluation as a nonlinear trajectory-solving problem. DEER (Lim et al., 2024), for example, applies Newton iterations on the state trajectory, where each iteration linearizes the recurrence around the current estimate and solves the resulting affine recurrence in parallel. LrcSSM (Farsang & Grosu, 2025) applies this approach to biologically plausible liquid-resistance, liquid-capacitance recurrent dynamics, while designing the neuron dynamics such that they have a diagonal Jacobian, substantially

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.

reducing the cost of each iteration. Nevertheless, evaluating the nonlinear recurrence still requires repeated parallel scans, with the number of iterations determined by convergence.

In this work, we take a different approach: rather than starting from a general nonlinear recurrence and recovering parallelism through iterative linearization, we ask *whether the nonlinear dynamics themselves can be chosen to remain exactly composable*. This requires a nonlinear family whose time-step maps are closed under composition with a fixed-size representation. We show that input-conditioned Riccati dynamics provide such a construction. Their exact per-step flow maps under piecewise-constant inputs are fractional-linear, or Möbius transformations, which compose through multiplication of 2×2 matrices. Consequently, the complete nonlinear state trajectory can be evaluated within a single associative parallel scan, without Newton or fixed-point iterations.

This leads us to RiccatiSSM, a diagonal nonlinear state space model, in which each state dimension follows an input-conditioned Riccati differential equation. The quadratic state term makes the local Jacobian explicitly state-dependent, preserving the central dynamical property motivating liquid recurrent models. Moreover, the Riccati structure makes the exact flow compositionally closed. Since unrestricted Riccati equations can exhibit finite-time divergence, we further derive a constrained parameterization that provides stable, bounded dynamics while preserving the exact Möbius flow. RiccatiSSM therefore occupies a middle ground between affine SSMs, which admit direct parallel scans but lack state-dependent local dynamics, and general nonlinear recurrent models, which provide such dynamics but require iterative methods for parallel evaluation.

Our main contributions in this paper are as follows:

- *We introduce RiccatiSSM*, a nonlinear state-dependent sequence model whose discrete-time flow maps are Möbius transformations. Their closure under composition enables exact, non-iterative sequence evaluation with a single associative scan.
- *We derive a constrained Riccati parameterization* that makes these Riccati dynamics particularly suitable for sequence modeling by ensuring a bounded invariant state interval, contractive transitions, and a pole-free projective readout.
- *We evaluate RiccatiSSM* on long-sequence classification, regression, and forecasting tasks, where it achieves competitive predictive performance while reducing runtime by $22-33\%$ relative to iterative nonlinear LrcSSM under matched architectures.

2 Background and Motivation

2.1 Parallel scans and state-dependent dynamics

Modern SSMs achieve efficient sequence evaluation because their discrete-time recurrence is affine in the previous state,

$$x_t = \Lambda_t x_{t-1} + b_t, \tag{1}$$

where Λ_t and b_t may depend on the current input u_t but not on x_{t-1}. Each time step therefore defines an affine map $f_t(x) = \Lambda_t x + b_t$. Affine maps are closed under composition:

$$f_2(f_1(x)) = \Lambda_2\Lambda_1 x + \Lambda_2 b_1 + b_2, \tag{2}$$

so a pair (Λ_t, b_t) provides a fixed-size representation of each update and can be combined with an associative operator. The complete sequence can consequently be evaluated using a parallel prefix scan with $O(T)$ work and $O(\log T)$ parallel depth.

Allowing the dynamics to depend nonlinearly on the current state changes the recurrent update to:

$$x_t = f_t(x_{t-1}), \tag{3}$$

where f_t is nonlinear. For a general nonlinear family, composing two updates does not yield another map with the same fixed-size representation. For example, composing two degree-k polynomials generally produces a degree-k^2 polynomial. The representation therefore grows under composition, preventing the recurrence from being evaluated by a single fixed-size associative scan.

Methods such as DEER (Lim et al., 2024) recover parallelism for general nonlinear recurrences through Newton iterations over the full state trajectory. At each iteration, the nonlinear recurrence

Table 1: RiccatiSSM compared with DEER/ELK-style (Lim et al., 2024; Gonzalez et al., 2024) parallelization of general nonlinear recurrences. Both evaluate state-dependent dynamics with associative scans, but DEER/ELK approximate the general nonlinear update by Newton iteration updates until convergence, whereas RiccatiSSM restricts the dynamics to a family that composes exactly in a single step. T is the sequence length, D the number of channels, and K the number of sweeps.

	DEER/ELK-based models	RiccatiSSM
Dynamics	any $f(x_{t-1}, u_t)$	Riccati: $\alpha + \beta x + \gamma x^2$
Scan element	affine $((J_t, c_t)$, J_t the state Jacobian)	fractional-linear, $M_t \in \mathbb{R}^{2\times 2}$
Composition	approximate (per iteration)	exact
Sweeps per layer	until convergence (K, data-dependent)	1, fixed
Depth per layer	$\mathcal{O}(K \log T)$	$\mathcal{O}(\log T)$
Work per layer	$\mathcal{O}(K\,TD)$	$\mathcal{O}(TD)$
Within-step integration	based on the model	exact (matrix exponential)
Convergence	data-dependent, may stall	not applicable

is linearized, which is then evaluated by a parallel scan, but several such iterations may be required. Thus, general nonlinear state dependence can be parallelized, but at the cost of repeated scans. Lrc-SSM (Farsang & Grosu, 2025) applies this approach to liquid recurrent dynamics while designing the model to have a diagonal Jacobian, reducing the cost of each iteration.

2.2 Closure under composition

An alternative is to restrict the nonlinear update family itself. For a recurrence to admit exact evaluation by a single associative scan, we seek a family $\mathcal{F}$ whose elements have a fixed-size representation and are closed under composition:

$$f_1, f_2 \in \mathcal{F} \quad \Longrightarrow \quad f_2 \circ f_1 \in \mathcal{F}. \tag{4}$$

Affine maps satisfy this property, but they remain linear in the state. We instead seek a nonlinear family that preserves the same closure property.

For a scalar state, fractional-linear (rational-linear), or Möbius, maps provide such a family:

$$f(x) = \frac{ax + b}{cx + d}. \tag{5}$$

Each such map can be represented projectively by a 2×2 matrix:

$$M_f = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \tag{6}$$

and the composition of two Möbius maps corresponds to matrix multiplication:

$$M_{f_2 \circ f_1} = M_{f_2} M_{f_1}, \tag{7}$$

Since matrix multiplication is associative, a sequence of Möbius updates can be composed exactly with a parallel prefix scan while retaining a fixed 2×2 representation at every step.

This observation motivates RiccatiSSM. Rather than approximating a general nonlinear recurrence through repeated linearization, we construct the state dynamics such that their exact discrete-time flow is a Möbius map. Riccati differential equations generate such fractional-linear flow maps, providing the continuous-time dynamics underlying RiccatiSSM, which we introduce next in Section 3.

3 Riccati State Space Model

We now construct a nonlinear sequence layer whose state-dependent dynamics admit exact parallel evaluation. The construction has four steps: (1) We first define an input-conditioned Riccati vector field. (2) We lift this nonlinear scalar ODE to a two-dimensional linear system, whose exact zero-order-hold solution induces a Möbius state update. (3) Since Möbius maps compose through matrix multiplication, the full sequence can be evaluated with a single associative scan. (4) Finally, we constrain the coefficients to ensure bounded trajectories and contractive state transitions.

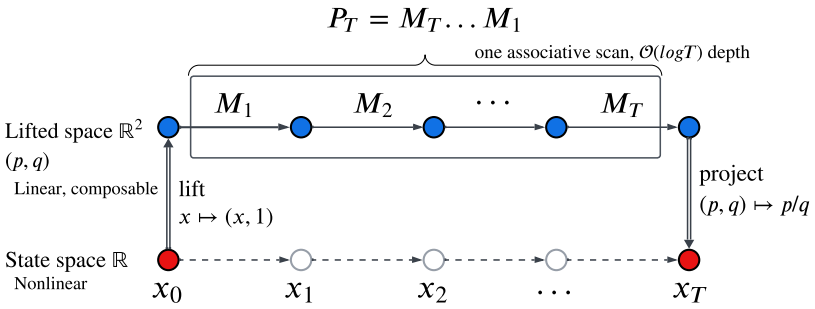


Figure 1: RiccatiSSM evaluates a nonlinear recurrence by a detour through a two-dimensional linear one. Bottom (grey, dashed): the states x_t of one channel, whose Möbius updates are nonlinear in the state. Composing such maps directly requires clearing nested fractions, so the bottom row offers no cheap merge to scan over. Top (blue, solid): the same states as homogeneous pairs (p_t, q_t), where each step is the matrix M_t. The initial state is lifted once, $x_0 \mapsto (x_0, 1)$, and states are recovered by $x_t = p_t/q_t$. Matrix multiplication is associative, so the top row collapses into prefix products $P_t = M_t \cdots M_1$ under one associative scan of depth $\mathcal{O}(\log T)$, with no fixed-point iterations; all D channels scan in parallel in the same round.

3.1 Input-dependent Riccati dynamics

Let u_t denote the input at time t, and let $x_t \in \mathbb{R}^D$ denote the hidden state. As in diagonal state space models and LrcSSM (Farsang & Grosu, 2025), RiccatiSSM uses independent scalar dynamics across state dimensions; interactions between dimensions are handled by the surrounding input and output projections. For each dimension i, we define the continuous-time dynamics as:

$$\dot{x}_i = \varepsilon_i(u) \left[\alpha_i(u) + \beta_i(u) x_i + \gamma_i(u) x_i^2\right]. \tag{8}$$

The coefficients $\alpha_i(u)$, $\beta_i(u)$, $\gamma_i(u)$, and $\varepsilon_i(u)$ are produced by input-dependent parameter heads. Here, α_i controls the input drive, β_i the linear state dependence, γ_i the quadratic curvature. Inspired by LRCs (Farsang et al., 2024), we use $\varepsilon_i \in (0, 1)$ to control the overall speed of the dynamics.

Unlike in an input-varying affine SSM, the local state Jacobian of the dynamics described in Equation 8, depends explicitly on the current state:

$$\frac{\partial \dot{x}_i}{\partial x_i} = \varepsilon_i \left(\beta_i + 2\gamma_i x_i\right). \tag{9}$$

Thus, the local contraction rate of the dynamics can vary with both the input, through the coefficients ε, β and γ, and the current state, through the quadratic term, giving the dynamics a liquid, input-and-state dependent character, like in Hasani et al. (2021); Farsang et al. (2024).

When $\gamma_i = 0$, Equation 8 reduces to an input-varying linear ODE. The quadratic term is therefore the lowest-degree polynomial term that introduces explicit state-dependence into the local Jacobian.

3.2 Projective lift and exact discretization

A scalar Riccati equation can be represented as the ratio of two coordinates evolving under a linear system. For clarity, we suppress the state index i and write

$$x = \frac{p}{q}.$$

Define the lifted state $z = (p, q)^\top$, and consider the linear system:

$$\frac{d}{dt}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = L \begin{pmatrix} p \\ q \end{pmatrix}, \qquad L = \varepsilon \begin{pmatrix} \beta/2 & \alpha \\ -\gamma & -\beta/2 \end{pmatrix}. \tag{10}$$

Applying the quotient rule to $x = p/q$ gives:

$$\dot{x} = \frac{\dot{p}q - p\dot{q}}{q^2} = \varepsilon \left(\alpha + \beta x + \gamma x^2\right), \tag{11}$$

recovering Equation 8. A full derivation of Equation 8 is given in Appendix A.2.

We apply a zero-order hold over each interval $[t\Delta t, (t+1)\Delta t)$: the coefficients are computed from u_t and held constant over the interval. The lifted system then has the exact one-step solution:

$$\begin{pmatrix} p_t \\ q_t \end{pmatrix} = M_t \begin{pmatrix} p_{t-1} \\ q_{t-1} \end{pmatrix}, \qquad M_t = \exp(\Delta t\, L_t). \tag{12}$$

Since L_t is traceless and satisfies $L_t^2 = \omega_t^2 I$, its exponential has the closed form:

$$M_t = \cosh(\omega_t \Delta t)\, I + \frac{\sinh(\omega_t \Delta t)}{\omega_t}\, L_t, \qquad \omega_t = \sqrt{\varepsilon_t^2 \left(\frac{\beta_t^2}{4} - \alpha_t \gamma_t \right)}. \tag{13}$$

Our stability parameterization that we introduce below, makes sure that $\omega_t > 0$, so the hyperbolic form in Equation 13 is always well defined. Writing:

$$M_t = \begin{pmatrix} a_t & b_t \\ c_t & d_t \end{pmatrix},$$

the induced update of the original scalar state becomes:

$$x_t = \frac{a_t x_{t-1} + b_t}{c_t x_{t-1} + d_t}. \tag{14}$$

Thus, exact zero-order-hold integration of a Riccati ODE produces a Möbius update. This is the property that makes the nonlinear recurrence scan-compatible. Although the dynamics are linear in the lifted coordinates, the projected state ($x = p/q$) follows the nonlinear, state-dependent Riccati dynamics in Equation 8. We discuss this and its relation to liquid dynamics further in Appendix B.

3.3 Parallel sequence evaluation

The Möbius updates in Equation 14 compose directly through their matrix representations. Define the the following prefix product:

$$P_t = M_t M_{t-1} \cdots M_1 = \begin{pmatrix} P_t^{(11)} & P_t^{(12)} \\ P_t^{(21)} & P_t^{(22)} \end{pmatrix}. \tag{15}$$

Since matrix multiplication is associative, all prefix products $\{P_t\}_{t=1}^T$ can be computed with a single parallel scan. The state at every time step is then recovered from the initial state x_0 as:

$$x_t = \frac{P_t^{(11)} x_0 + P_t^{(12)}}{P_t^{(21)} x_0 + P_t^{(22)}}. \tag{16}$$

The forward pass therefore consists of three non-iterative stages:

1. *Coefficient generation.* Compute the coefficients $(\alpha_t, \beta_t, \gamma_t, \varepsilon_t)$ and the corresponding matrices M_t, for all time steps and state dimensions, in parallel.
2. *Parallel scan.* Compute the matrix prefix-products in Equation 15, by using the associativity property of 2×2 matrix multiplication.
3. *Projective readout.* Recover the state x_t from the readout of Equation 16.

It is important to note that for a diagonal state dynamics of dimension D and sequence length T, the recurrent computation requires $\mathcal{O}(TD)$ work and $\mathcal{O}(\log T)$ parallel depth. Under the zero-order-hold assumption, each Riccati step is integrated exactly. Hence the scan introduces neither iterative approximation nor a within-step numerical integration error.

3.4 Stable parameterization

Unfortunately, a general Riccati equation can diverge in finite time. In the fractional-linear update of Equation 14, this divergence corresponds to the denominator approaching zero (a pole). We avoid this behavior by constraining the Riccati coefficients in such a way that the dynamics admit a bounded invariant interval, and the state transition is contractive within this interval. The resulting parameterization and stability conditions are detailed in Appendix A.1.

3.5 Relation to Liquid Dynamics

RiccatiSSM and LrcSSM both use diagonal continuous-time neuron dynamics, in order to avoid the matrix-multiplication blow-up within the associative scan. However, they differ in how they represent state dependence. LrcSSM places the state inside saturating nonlinearities, making its update, leak, and speed (elastance) factor state-dependent. RiccatiSSM instead uses input-dependent coefficients and represents state dependence explicitly through a quadratic term. This structure yields an exact fractional-linear step map that can be composed with a single associative scan.

We establish a local correspondence between the two models by constructing a second-order Taylor approximation of the LRC vector field with respect to the state around its zero value. We further evaluate the accuracy of this local approximation and distinguish it from the discretization error introduced by the explicit-Euler update used in LrcSSM. The derivation and the matched-dynamics experiments are provided in Appendix B.1 and B.2.

4 Experiments

We evaluate RiccatiSSM on long-sequence classification, regression, and forecasting tasks, comparing its predictive performance with transformer-based architectures, neural differential equation models, and with linear, oscillatory, input-selective and nonlinear SSMs. We then measure the computational benefit of replacing the iterative quasi-DEER evaluation of LrcSSM with the exact single-scan evaluation of RiccatiSSM. Finally, we ablate the RiccatiSSM parameterization to study the effects of input-dependent coefficients and the connection to LRC dynamics.

4.1 Experimental setup

RiccatiSSM follows the sensory-synapse input encoding of LrcSSM (Farsang & Grosu, 2025), but uses independent learned projections of this representation to produce the Riccati coefficients $\alpha(u)$, $\beta(u)$, $\gamma(u)$, and $\varepsilon(u)$. The resulting coefficients are mapped through the constrained parameterization of Section A.1, which enforces the stability conditions of RiccatiSSM. We separately evaluate an LRC-tied parameterization in Section 4.4, where these coefficients are constrained by the Taylor-matching relations derived in Appendix B.1.

Table 2: Test accuracy comparison of different models. The performance of the models marked by † is reported from Rusch & Rus (2025), those with ‡ from Moreno-Pino et al. (2024), and ⋄ from Farsang & Grosu (2025). The same hyperparameter tuning protocol and dataset splitting over the same 5 seeds were used. Bold highlights the top 3 models.

	Heart	SCP1	SCP2	Ethanol	Motor	Worms
Sequence length	405	896	1,152	1,751	3,000	17,984
Input size	61	6	7	2	63	6
#Classes	2	2	2	4	2	5
Transformer‡	70.5 ± 0.1	84.3 ± 6.3	49.1 ± 2.5	**40.5 ± 6.3**	50.5 ± 3.0	OOM
RFormer‡	72.5 ± 0.1	81.2 ± 2.8	52.3 ± 3.7	34.7 ± 4.1	55.8 ± 6.6	**90.3 ± 0.1**
NRDE†	73.9 ± 2.6	76.7 ± 5.6	48.1 ± 11.4	31.4 ± 4.5	54.0 ± 7.8	77.2 ± 7.1
NCDE†	68.1 ± 5.8	80.0 ± 2.0	49.1 ± 6.2	22.0 ± 1.0	51.6 ± 6.2	62.2 ± 2.2
Log-NCDE†	74.2 ± 2.0	82.1 ± 1.4	54.0 ± 2.6	35.9 ± 6.1	57.2 ± 5.6	82.8 ± 2.7
LRU†	**78.1 ± 7.6**	84.5 ± 4.6	47.4 ± 4.0	23.8 ± 2.8	51.9 ± 8.6	85.0 ± 6.2
S5†	73.9 ± 3.1	87.1 ± 2.1	**55.1 ± 3.3**	25.6 ± 3.5	53.0 ± 3.9	83.9 ± 4.1
LinOSS-IMEX†	75.5 ± 4.3	**87.5 ± 4.0**	**58.9 ± 8.1**	29.9 ± 1.0	57.9 ± 5.3	80.0 ± 2.7
LinOSS-IM†	75.8 ± 3.7	**87.8 ± 2.6**	**58.2 ± 6.9**	29.9 ± 0.6	**60.0 ± 7.5**	**95.0 ± 4.4**
Mamba†	**76.2 ± 3.8**	80.7 ± 1.4	48.2 ± 3.9	27.9 ± 4.5	47.7 ± 4.5	70.9 ± 15.8
S6†	**76.5 ± 8.3**	82.8 ± 2.7	49.9 ± 9.4	26.4 ± 6.4	51.3 ± 4.7	85.0 ± 16.1
LrcSSM⋄	72.7 ± 5.7	85.2 ± 2.1	53.9 ± 7.2	**36.9 ± 5.3**	**58.6 ± 3.1**	**90.6 ± 1.4**
RiccatiSSM (ours)	73.0 ± 6.8	**87.4 ± 4.3**	53.2 ± 4.3	**36.6 ± 3.4**	**59.6 ± 3.1**	83.9 ± 6.4

Results are reported over five random seeds. Datasets, model configurations, and hyperparameter ranges are provided in Appendix C, with the complete hyperparameter settings.

4.2 LONG-SEQUENCE BENCHMARK RESULTS

Classification. We first evaluate RiccatiSSM on six multivariate time-series classification datasets from the UEA archive, spanning sequence lengths from 405 to $17{,}984$. Table 2 compares RiccatiSSM with continuous-time recurrent models, linear SSMs, Transformer-based architectures, and the nonlinear LrcSSM baseline. RiccatiSSM is competitive across datasets and closely matches LrcSSM while avoiding its iterative sequence evaluation. In particular, it improves over LrcSSM on SCP1 and MotorImagery, while obtaining similar performance on Heartbeat, SCP2, and Ethanol-Concentration. On EigenWorms, RiccatiSSM shows higher variability across seeds, with individual runs ranging from 77.8% to 91.7%, suggesting greater sensitivity to optimization on this dataset.

Long-range Regression and Forecasting We next evaluate regression on PPG-DaLiA (Reiss et al., 2019). Following the same tuning protocol, RiccatiSSM achieves an MSE of $7.15 \pm 1.01 \times 10^{-2}$ (Table 3), improving over LrcSSM ($10.89 \pm 0.96 \times 10^{-2}$) and remaining competitive with the strongest linear SSM baseline, the LinOSS.

Table 3: Mean squared error (MSE × 10^{-2}) for different models on the PPG-DaLiA dataset. The performance of the models marked by † is reported from Rusch & Rus (2025) and ⋄ from Farsang & Grosu (2025). All results are averaged over 5 seeds.

Model	MSE $\times 10^{-2}$ (↓)
NRDE† (Morrill et al., 2021)	9.90 ± 0.97
NCDE† (Kidger et al., 2020)	13.54 ± 0.69
Log-NCDE† (Walker et al., 2024)	9.56 ± 0.59
LRU† (Orvieto et al., 2023)	12.17 ± 0.49
S5† (Smith et al., 2023)	12.63 ± 1.25
LinOSS-IMEX† (Rusch & Rus, 2025)	$\mathbf{7.50 \pm 0.46}$
LinOSS-IM† (Rusch & Rus, 2025)	$\mathbf{6.40 \pm 0.23}$
S6† (Gu & Dao, 2024)	12.88 ± 2.05
Mamba† (Gu & Dao, 2024)	10.65 ± 2.20
LrcSSM⋄ (Farsang & Grosu, 2025)	10.89 ± 0.96
RiccatiSSM (ours)	$\mathbf{7.15 \pm 1.13}$

To evaluate long-horizon forecasting, we use the Weather dataset with an input context of 720 time steps and predict the subsequent 720 steps, following Zhou et al. (2021). As shown in Table 4, RiccatiSSM obtains a mean absolute error of 0.5681, improving over LrcSSM (0.5888) and several recurrent, Transformer-based, and state-space baselines, while remaining close to LinOSS.

Table 4: Mean absolute error on the weather dataset predicting 720 future time steps based on 720 past time steps. The performance of the models marked by † is reported from Rusch & Rus (2025).

Model	Mean Absolute Error (↓)
Informer† (Zhou et al., 2021)	0.731
LogTrans† (Li et al., 2019)	0.773
Reformer† (Kitaev et al., 2020)	1.575
LSTMa† (Bahdanau et al., 2014)	1.109
LSTnet† (Lai et al., 2018)	0.757
S4† (Gu et al., 2022)	0.5783
LinOSS-IMEX† (Rusch & Rus, 2025)	$\mathbf{0.5081}$
LinOSS-IM† (Rusch & Rus, 2025)	$\mathbf{0.5282}$
LrcSSM (Farsang & Grosu, 2025)	0.5888
RiccatiSSM (ours)	$\mathbf{0.5681}$

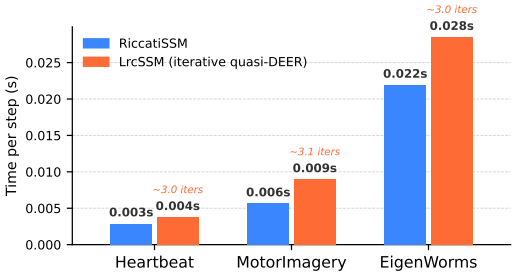


Figure 2: Time per step (seconds) for RiccatiSSM and LrcSSM across three UEA time-series classification datasets using the same architecture on an NVIDIA A100 GPU. RiccatiSSM evaluates each layer with a single parallel scan, whereas LrcSSM requires approximately three quasi-DEER iterations. The runtime gap is largest for the longer EigenWorms sequences.

4.3 Runtime and parallel-evaluation cost

We next evaluate whether eliminating iterative nonlinear sequence evaluation translates into an end-to-end computational benefit. We compare RiccatiSSM with LrcSSM using the same 6-layer architecture and a state size of 64. While RiccatiSSM evaluates its recurrent dynamics with a single associative scan, LrcSSM requires approximately three quasi-DEER iterations, each involving a scan of a linearized recurrence.

Figure 2 shows that eliminating these iterative solves consistently reduces wall-clock runtime. The reduction is smaller than the approximately $3\times$ difference in the number of scans because sequence evaluation constitutes only part of the complete architecture: embeddings, MLPs, normalization layers, and other operations are evaluated once in both models. Across the three datasets, RiccatiSSM requires $67-78\%$ of the LrcSSM runtime, corresponding to a $22-33\%$ reduction in end-to-end computation. The difference is largest on EigenWorms, which has the longest sequences. Additional runtime measurements are provided in Table 8 in the Appendix.

4.4 Ablations

Input-dependent coefficients. We ablate several components of the RiccatiSSM parameterization using PPG-DaLiA. First, we remove the input-dependent speed factor $\varepsilon(u)$, which increases the test MSE by 1.3×10^{-2}. We then restrict β, γ, and ε to be input-independent, leaving only $\alpha(u)$ to inject the input into the state dynamics. Note that at least one of the coefficients should remain input dependent, such that the input can influence the state dynamics. The results indicate that input-dependent modulation of the full Riccati dynamics is beneficial.

Free vs. LRC-tied parameterization. The local correspondence derived in Appendix B.1 provides a second way to parameterize RiccatiSSM. Rather than learning the Riccati coefficients independently, we constrain them according to the Taylor-matching relations derived from the LrcSSM vector field. This LRC-tied parameterization obtains an MSE of $8.47\pm0.47\times10^{-2}$, compared with $7.15\pm1.01\times10^{-2}$ for the freely parameterized RiccatiSSM. Thus, retaining the LRC-derived coupling is not necessary for RiccatiSSM and is more restrictive than learning the Riccati coefficients independently subject to the stability constraints.The reachable coefficient spaces induced by the free and LRC-tied parameterizations are visualized in Figure 4 in the Appendix.

4.5 Limitations

Compared to general nonlinear systems, RiccatiSSM is restricted to a specific second-order polynomial form in which the coefficients can be input-dependent but not state-dependent, i.e., α, β, and γ depend only on the input. State-dependent nonlinearity instead enters explicitly through the quadratic curvature term $\gamma(u)x^2$. Additional constraints are required to ensure stable dynamics. Despite these restrictions, RiccatiSSMs define an interesting new family of nonlinear SSMs that can be parallelized exactly with a single scan without iterative methods.

Table 5: Ablation of the RiccatiSSM parameterization on PPG-DaLiA. Test MSE ($\times 10^{-2}$, lower is better), reported as mean $\pm$ standard deviation over five seeds.

Model	Input-dependent coefficients	MSE (↓)
RiccatiSSM (ours)	$\alpha(u), \beta(u), \gamma(u), \varepsilon(u)$	$\mathbf{7.15 \pm 1.01}$
ε ablation	$\alpha(u), \beta(u), \gamma(u)$	8.45 ± 0.38
Reduced input dependence	$\alpha(u), \beta, \gamma, \varepsilon$	8.71 ± 1.10
RiccatiSSM (LRC-tied)	LRC Taylor-matched	8.47 ± 0.47

5 Related Work

Parallel evaluation of nonlinear recurrences. Linear and input-dependent linear SSMs evaluate long sequences with associative scans because affine state updates are closed under composition (Smith et al., 2023; Gu & Dao, 2024). Several recent methods extend parallel evaluation to nonlinear recurrences through iterative linearization. DEER (Lim et al., 2024) formulates sequence evaluation as a fixed-point problem and applies Newton-style iterations, where each iteration scans a linearized recurrence. Quasi-DEER and ELK (Gonzalez et al., 2024) develop related iterative schemes intended to improve stability and scalability. Fixed-Point RNNs (Movahedi et al., 2025) express dense linear recurrences as fixed points of parallelizable diagonal systems, while Gonzalez et al. (2026) study Newton, Picard, and Jacobi approaches within a common successive-linearization framework. Gonzalez et al. (2025) further relate the convergence of these methods to properties such as contractivity of the underlying dynamics. Here, we explore the boundary of associative scans.

Liquid and continuous-time models. LTC (Hasani et al., 2021) makes each neuron's effective time constant depend on it own state, by using a liquid conductance. CfC (Hasani et al., 2022) removes the ODE solver by integrating a single step in closed form, but the sequence recursion remains sequential and its coefficients remain state-dependent, so parallelization over T still requires DEER-style iteration. LRC and LrcSSM (Farsang et al., 2024; Farsang & Grosu, 2025) extend this family with a liquid capacitance, and scale it to long sequences with a diagonal Jacobian and a contraction-based gradient-stability guarantee. While not fully following the biological constraints anymore, RiccatiSSM keeps both the state-dependent time constant (Equation 9) and the contraction guarantee, and additionally makes the *within-step* discretization exact rather than explicit Euler.

Möbius maps and parallel prefix. That fractional-linear maps compose thruough a 2×2 matrix multiplication, and that first-order rational recurrences therefore admit parallel-prefix evaluation, is classical (Kogge & Stone, 1973; Kogge, 1974); we claim no novelty for the primitive itself. In machine learning, Möbius transformations have been used as feature maps in attention (Halacheva et al., 2024), but as static transformations of representations, not as temporal dynamics composed over time. The closest use of the primitive as a sequence recurrence is Kalman Linear Attention (Shaj et al., 2026), where the information-form *precision* of a linear-Gaussian latent follows a Möbius recursion composed by an associative scan. The hidden *state* evolves linearly and the fractional-linear map acts on an uncertainty statistic. In RiccatiSSM the map acts on the state itself (Equation 8), which is what yields a state-dependent time constant and places the model in the liquid family.

6 Conclusion

We introduced RiccatiSSM, a nonlinear sequence layer whose input-conditioned Riccati dynamics yield exact fractional-linear state updates. Because these updates are closed under composition through 2×2 matrix multiplication, RiccatiSSM evaluates the full sequence with a single associative parallel scan, avoiding the iterative Newton updates required by general nonlinear recurrences. A constrained parameterization keeps the dynamics bounded and contractive while retaining state-dependent timescales. Across long-sequence classification, regression, and forecasting tasks, RiccatiSSM achieves competitive predictive performance, while reducing end-to-end runtime by $22 - 33\%$ relative to the iterative LrcSSM implementation. More broadly, RiccatiSSM shows that state-dependent nonlinear dynamics and exact parallel evaluation need not be mutually exclusive: restricting the dynamics to a nonlinear family closed under composition provides a middle ground between linear SSMs and general nonlinear recurrent models.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code development, the writing of theoretical sections, and improvements to the overall readability of the manuscript. All AI-assisted content was reviewed by the authors, and AI-generated code was independently verified. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

The code required to reproduce our experiments is included in the supplementary materials. The Appendix provides further details on the datasets, model configurations, training procedure, and hyperparameter settings.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Mónika Farsang and Radu Grosu. Parallelization of non-linear state-space models: Scaling up liquid-resistance liquid-capacitance networks for efficient sequence modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ENYvdnyhLl.

Mónika Farsang, Sophie A Neubauer, and Radu Grosu. Liquid resistance liquid capacitance networks. In *The First Workshop on NeuroAI@ NeurIPS2024*, 2024.

Xavier Gonzalez, Andrew Warrington, Jimmy Smith, and Scott Linderman. Towards scalable and stable parallelization of nonlinear rnns. *Advances in Neural Information Processing Systems*, 37: 5817–5849, 2024.

Xavier Gonzalez, Leo Kozachkov, David M. Zoltowski, Kenneth L. Clarkson, and Scott Linderman. Predictability enables parallelization of nonlinear state space models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=7AGXSlXcK6.

Xavier Gonzalez, E. Kelly Buchanan, Hyun Dong Lee, Jerry Weihong Liu, Ke Alexander Wang, David M. Zoltowski, Leo Kozachkov, Chris Ré, and Scott W. Linderman. A unifying framework for parallelizing sequential models with linear dynamical systems. *Transactions on Machine Learning Research (TMLR)*, 2026.

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.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022. URL https://arxiv.org/abs/2111.00396. doi:10.48550/arXiv.2111.00396.

Anna-Maria Halacheva, Mojtaba Nayyeri, and Steffen Staab. Expanding expressivity in transformer models with m\" obiusattention. *arXiv preprint arXiv:2409.12175*, 2024.

Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 35(9), pp. 7657–7666, 2021. doi:10.1609/aaai.v35i9.16936.

Ramin Hasani, Mathias Lechner, Alexander Amini, Lucas Liebenwein, Aaron Ray, Max Tschaikowski, Gerald Teschl, and Daniela Rus. Closed-form continuous-time neural networks. *Nature Machine Intelligence*, 4(11):992–1003, 2022.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Peter M Kogge. Parallel solution of recurrence problems. *IBM Journal of Research and Development*, 18(2):138–148, 1974.

Peter M Kogge and Harold S Stone. A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE transactions on computers*, 100(8):786–793, 1973.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

Yi Heng Lim, Qi Zhu, Joshua Selfridge, and Muhammad Firmansyah Kasim. Parallelizing non-linear sequential models over the sequence length. In *The Twelfth International Conference on Learning Representations*, 2024.

Fernando Moreno-Pino, Álvaro Arroyo, Harrison Waldon, Xiaowen Dong, and Álvaro Cartea. Rough transformers: Lightweight and continuous time series modelling through signature patching. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 106264–106294. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/bfe167fee3be2d862a56af82dee77720-Paper-Conference.pdf.

James Morrill, Cristopher Salvi, Patrick Kidger, and James Foster. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pp. 7829–7838. PMLR, 2021.

Sajad Movahedi, Felix Sarnthein, Nicola Muca Cirone, and Antonio Orvieto. Fixed-point RNNs: Interpolating from diagonal to dense. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KT8y9pFgJE.

Philipp* Nazari, Francesco Maria* Ruscio, Benedict* Armstrong, and T. Konstantin Rusch. Discretax: A lightweight collection of state space models in jax, 2025. URL https://github.com/camail-official/discretax.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698. PMLR, 2023.

Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19(14):3079, 2019.

T Konstantin Rusch and Daniela Rus. Oscillatory state-space models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Vaisakh Shaj, Cameron Barker, Aidan Scannell, Andras Szecsenyi, Elliot J Crowley, and Amos Storkey. Kalman linear attention: Parallel bayesian filtering for efficient language modelling and state tracking. *arXiv preprint arXiv:2602.10743*, 2026.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. In *ICLR*, 2023.

Benjamin Walker, Andrew Donald McLeod, Tiexin Qin, Yichuan Cheng, Haoliang Li, and Terry Lyons. Log neural controlled differential equations: The lie brackets make a difference. In *Forty-first International Conference on Machine Learning*, 2024.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

A Additional Theoretical Results

This appendix proves the algebraic and stability properties underlying RiccatiSSM. We first show the parametrization, then that a Riccati equation is the projective image of a two-dimensional linear system. We then derive the exact matrix exponential used for zero-order-hold discretization, establish closure of the resulting Möbius updates under composition, and prove the guarantees of the constrained parameterization.

A.1 Stable parameterization

A general Riccati equation can diverge in finite time. In the fractional-linear update of Equation 14, such a divergence corresponds to the denominator approaching zero. We avoid this behavior by parameterizing the coefficients such that a fixed interval $[-B, B]$, with $B > 0$, is forward-invariant and the dynamics are contractive within this interval.

For readability, we suppress the state-dimension and time indices in this subsection. At each time step, the coefficient network produces four unconstrained real-valued outputs

$$(\hat{\alpha}, \hat{\beta}, \hat{\gamma}, \hat{\varepsilon}) \in \mathbb{R}^4. \tag{17}$$

The hats indicate raw network outputs before applying the constraints. We first define

$$\varepsilon = \sigma(\hat{\varepsilon}), \qquad s = s_{\min} + \mathrm{softplus}(\hat{\beta}), \qquad \gamma = \hat{\gamma}, \tag{18}$$

where $s_{\min} > 0$ is a fixed minimum contraction margin. The remaining Riccati coefficients are then parameterized as

$$\alpha = \left(sB + |\gamma|B^2\right) \tanh(\hat{\alpha}), \qquad \beta = -\left(s + 2|\gamma|B\right). \tag{19}$$

Thus, $\hat{\alpha}$ determines the bounded drive coefficient α, $\hat{\beta}$ determines the positive contraction margin s, $\hat{\gamma}$ determines the curvature γ, and $\hat{\varepsilon}$ determines the speed factor $\varepsilon \in (0, 1)$.

In particular, $\hat{\beta}$ does not directly parameterize the linear Riccati coefficient β. Instead, it determines the positive margin s, from which β is constructed to offset the largest possible state-dependent contribution $2|\gamma|x$ to the Jacobian over $x \in [-B, B]$, while retaining a contraction margin of at least s. The resulting coefficients satisfy,

$$|\alpha| \le sB + |\gamma|B^2, \qquad \beta \le -s \le -s_{\min}. \tag{20}$$

These constraints give three properties.

Bounded trajectories. Consider the scalar Riccati vector field without its positive speed factor,

$$g(x) = \alpha + \beta x + \gamma x^2. \tag{21}$$

At the upper boundary $x = B$, using Equation 19, the coefficient bounds in Equation 20 imply

$$\begin{aligned} g(B) &= \alpha - \left(s + 2|\gamma|B\right) B + \gamma B^2 \\ &\le \left(sB + |\gamma|B^2\right) - sB - 2|\gamma|B^2 + \gamma B^2 \\ &= \left(\gamma - |\gamma|\right) B^2 \le 0. \end{aligned} \tag{22}$$

At the lower boundary $x = -B$,

$$\begin{aligned} g(-B) &= \alpha + \left(s + 2|\gamma|B\right) B + \gamma B^2 \\ &\ge -\left(sB + |\gamma|B^2\right) + sB + 2|\gamma|B^2 + \gamma B^2 \\ &= \left(|\gamma| + \gamma\right) B^2 \ge 0. \end{aligned} \tag{23}$$

Since $\varepsilon > 0$, the full vector field $\dot{x} = \varepsilon g(x)$ points inward or is tangent at both boundaries. Hence, $[-B, B]$ is forward-invariant: if $x_0 \in [-B, B]$, then $x_t \in [-B, B]$ for all subsequent times. The trajectory therefore remains finite, and the corresponding projective representation does not encounter a pole along the trajectory.

Contractive state transitions. For every $x \in [-B, B]$, the state Jacobian satisfies

$$\begin{aligned}\frac{\partial \dot{x}}{\partial x} &= \varepsilon\left(\beta + 2\gamma x\right) \\ &= \varepsilon\left(-s - 2|\gamma|B + 2\gamma x\right) \\ &\leq -\varepsilon s \leq -\varepsilon s_{\min}.\end{aligned} \tag{24}$$

Thus, the continuous-time dynamics are strictly contractive with respect to the state throughout the invariant interval. Under the zero-order hold used in Equation 12, ε_t and s_t are constant within each step, and the exact one-step flow therefore satisfies

$$0 < \frac{\partial x_t}{\partial x_{t-1}} \leq \exp(-\varepsilon_t s_t \Delta t) \leq \exp(-\varepsilon_t s_{\min} \Delta t) < 1. \tag{25}$$

Hence, the recurrent state transition is contractive within $[-B, B]$.

Real matrix-exponential branch. Finally, the discriminant in Equation 13 is strictly positive:

$$\begin{aligned}\frac{\beta^2}{4} - \alpha\gamma &\geq \frac{\left(s + 2|\gamma|B\right)^2}{4} - \left(sB + |\gamma|B^2\right)|\gamma| \\ &= \frac{s^2}{4} > 0.\end{aligned} \tag{26}$$

It follows that

$$\omega^2 = \varepsilon^2\left(\frac{\beta^2}{4} - \alpha\gamma\right) \geq \frac{\varepsilon^2 s^2}{4} > 0. \tag{27}$$

This ω is real and strictly positive, and the exact matrix exponential in Equation 13 therefore always takes the real hyperbolic branch. In summary, the proposed parameterization provides a bounded invariant state interval, contractive state transitions within that interval, pole-free trajectories, and a real-valued closed-form discretization.

A.2 Projective lift of Riccati dynamics

For this subsection, suppress the state-dimension and time indices. Consider the scalar Riccati equation

$$\dot{x} = \varepsilon\left(\alpha + \beta x + \gamma x^2\right). \tag{28}$$

Projective lift. Let $p(t)$ and $q(t)$ satisfy

$$\frac{d}{dt}\begin{pmatrix} p \\ q \end{pmatrix} = L\begin{pmatrix} p \\ q \end{pmatrix}, \qquad L = \varepsilon\begin{pmatrix} \beta/2 & \alpha \\ -\gamma & -\beta/2 \end{pmatrix}. \tag{29}$$

If $q(t) \neq 0$ on an interval I, then $x(t) = p(t)/q(t)$ satisfies Equation 28 on I.

Proof. Expanding Equation 29 componentwise gives

$$\dot{p} = \varepsilon\left(\frac{\beta}{2}p + \alpha q\right), \qquad \dot{q} = \varepsilon\left(-\gamma p - \frac{\beta}{2}q\right). \tag{30}$$

Since $x = p/q$, the quotient rule gives

$$\begin{aligned}\dot{x} &= \frac{\dot{p}q - p\dot{q}}{q^2} \\ &= \frac{\varepsilon\left[\left(\frac{\beta}{2}p + \alpha q\right)q - p\left(-\gamma p - \frac{\beta}{2}q\right)\right]}{q^2} \\ &= \varepsilon\left(\alpha + \beta\frac{p}{q} + \gamma\frac{p^2}{q^2}\right) \\ &= \varepsilon\left(\alpha + \beta x + \gamma x^2\right).\end{aligned} \tag{31}$$

which recovers Equation 28.

The projective representation is unchanged under a common nonzero scaling of p and q. More generally, adding a scalar multiple of the identity to L only introduces a common multiplicative factor in the lifted coordinates and therefore leaves the ratio $x = p/q$ unchanged. We choose the traceless representative in Equation 29, for which $\mathrm{tr}(L) = 0$. This choice yields the convenient identity $L^2 = \omega^2 I$ used in Equation 13.

A.3 Exact matrix exponential

Under the zero-order hold, the coefficients $(\alpha, \beta, \gamma, \varepsilon)$ are constant over one time interval of duration Δt. The lifted state is therefore updated by

$$z_t = \exp(\Delta t L_t) z_{t-1}, \qquad z_t = \begin{pmatrix} p_t \\ q_t \end{pmatrix}.$$

Closed-form step matrix. Let L be the generator in Equation 29, and define the characteristic rate

$$\omega = \sqrt{\varepsilon^2 \left(\frac{\beta^2}{4} - \alpha\gamma \right)}. \tag{32}$$

If $\omega \neq 0$, the exact step matrix is

$$\exp(\Delta t L) = \cosh(\omega \Delta t) I + \frac{\sinh(\omega \Delta t)}{\omega} L. \tag{33}$$

Proof. Direct multiplication gives

$$L^2 = \varepsilon^2 \begin{pmatrix} \beta/2 & \alpha \\ -\gamma & -\beta/2 \end{pmatrix}^2 = \varepsilon^2 \left(\frac{\beta^2}{4} - \alpha\gamma \right) I = \omega^2 I. \tag{34}$$

Thus, even powers of L are scalar multiples of I, and odd powers are scalar multiples of L. Substituting these identities into the power-series definition of the matrix exponential gives

$$\begin{aligned} \exp(\Delta t L) &= \sum_{k=0}^{\infty} \frac{(\Delta t L)^k}{k!} \\ &= \sum_{k=0}^{\infty} \frac{(\omega \Delta t)^{2k}}{(2k)!} I + \frac{1}{\omega} \sum_{k=0}^{\infty} \frac{(\omega \Delta t)^{2k+1}}{(2k+1)!} L \\ &= \cosh(\omega \Delta t) I + \frac{\sinh(\omega \Delta t)}{\omega} L, \end{aligned} \tag{35}$$

which proves Equation 33.

The stable parameterization in Section A.1 guarantees $\beta^2/4 - \alpha\gamma > 0$, and since $\varepsilon \in (0, 1)$, it follows that $\omega^2 > 0$. Thus the constrained RiccatiSSM always uses the real hyperbolic form in Equation 33.

A.4 Composition of Möbius updates

Möbius composition. For a matrix

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

define the associated fractional-linear map

$$f_M(x) = \frac{ax + b}{cx + d}, \tag{36}$$

whenever $cx + d \neq 0$. For any two matrices M_1 and M_2,

$$f_{M_2}\left(f_{M_1}(x)\right) = f_{M_2 M_1}(x) \tag{37}$$

whenever the corresponding fractional-linear maps are defined.

Proof. Let

$$M_1 = \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}, \qquad M_2 = \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix}.$$

Substituting $f_{M_1}(x)$ into f_{M_2} gives

$$\begin{aligned} f_{M_2}\left(f_{M_1}(x)\right) &= \frac{a_2\left(\frac{a_1x+b_1}{c_1x+d_1}\right)+b_2}{c_2\left(\frac{a_1x+b_1}{c_1x+d_1}\right)+d_2} \\ &= \frac{(a_2a_1+b_2c_1)x+(a_2b_1+b_2d_1)}{(c_2a_1+d_2c_1)x+(c_2b_1+d_2d_1)}. \end{aligned} \tag{38}$$

These coefficients are exactly the entries of M_2M_1, which proves Equation 37.

Repeated application of Equation 37 therefore represents the composed updates up to time t by the ordered prefix product

$$P_t = M_tM_{t-1}\cdots M_1$$

as defined in Equation 15. Since matrix multiplication is associative, all prefix products can be computed with a single associative parallel scan.

B Relation to Liquid Dynamics

Remark: Linear lift and nonlinear state dynamics. Equation 10 shows that in the lifted coordinates (p, q) each channel evolves as a two-dimensional linear time-varying system with input-only coefficients, followed by the projective state $x = p/q$. This structure is what allows the sequence to be evaluated with a single scan. The nonlinearity enters through the projective representation, but x is the state exposed to the next layer and follows the nonlinear Riccati dynamics in Equation 8. In x, the dynamics are bounded, contractive, exactly integrated, and have a state-dependent local contraction rate, and hence a state-dependent effective time constant, shown in Equation 9, as in liquid models. What the lift cannot provide is nonlinear feedback into the coefficients, since making α, β, γ or ε depend on x would break closure under composition.

B.1 Second-order local correspondence

For a fixed input u, write the continuous-time vector field of an LRC neuron as

$$f_i(x;u) = \sigma\big(\varepsilon_i^*(x,u)\big)\left[-\sigma\big(f_i^*(x,u)\big)x + \tau\big(z_i^*(x,u)\big)e_i^{leak}\right], \tag{39}$$

where f_i^*, z_i^*, and ε_i^* are the LRC pre-activations, and σ and τ are its nonlinearities. With u fixed, $f_i(\cdot;u)$ is a scalar function of the state x.

Expanding the LRC vector field around the reference state $x = 0$ gives

$$f_i(x;u) = a_i(u) + b_i(u)x + c_i(u)x^2 + \mathcal{O}(|x|^3), \tag{40}$$

where

$$a_i(u) = f_i(0;u), \qquad b_i(u) = f_i'(0;u), \qquad c_i(u) = \frac{1}{2}f_i''(0;u). \tag{41}$$

The corresponding quadratic field

$$g_i(x;u) = a_i(u) + b_i(u)x + c_i(u)x^2 \tag{42}$$

is the unique polynomial of degree at most two that matches the LRC field's value, slope, and curvature at $x = 0$.

This quadratic field is a Riccati vector field with speed factor $\varepsilon_i(u) = 1$. More generally, for any chosen positive speed factor $\varepsilon_i(u)$, the equivalent Riccati coefficients are

$$\alpha_i(u) = \frac{a_i(u)}{\varepsilon_i(u)}, \qquad \beta_i(u) = \frac{b_i(u)}{\varepsilon_i(u)}, \qquad \gamma_i(u) = \frac{c_i(u)}{\varepsilon_i(u)}. \tag{43}$$

Only the products $\varepsilon_i\alpha_i$, $\varepsilon_i\beta_i$, and $\varepsilon_i\gamma_i$ determine the vector field. Thus, the decomposition into a speed factor and the remaining coefficients is a parameterization choice.

Table 6: Relative RMSE over $T = 512$ steps, receiving white-noise input.

Factor isolated	Comparison	Relative RMSE
Truncation effect (Exact integration)	RiccatiSSM-ZOH vs. LrcSSM-RK4	1.73×10^{-5}
Truncation effect (Euler integration)	RiccatiSSM-Euler vs. LrcSSM-Euler	1.73×10^{-5}
Integrator choice LrcSSM	LrcSSM-Euler vs. LrcSSM-RK4	7.16×10^{-3}

The correspondence is local and does not imply global agreement between the models. In particular, the LRC field is globally bounded by its saturating nonlinearities, whereas an unconstrained quadratic field can develop a pole outside the local operating region. The trained RiccatiSSM uses the constrained parameterization described in Section A.1 to prevent this behavior.

The next section empirically evaluates the accuracy and range of validity of this local correspondence, including the error introduced by the explicit-Euler discretization used by LrcSSM.

B.2 Matched-dynamics experiment

We next compare the continuous-time dynamics of the two models while holding their local vector fields fixed. At every time step t, we compute the Taylor coefficients $(a_{i,t}, b_{i,t}, c_{i,t})$ from Equation 41 using the current input frame u_t. We obtain the derivatives by automatic differentiation of the implemented LRC vector field. The Riccati systems in this analysis use the resulting matched quadratic field

$$\dot{x}_i = a_{i,t} + b_{i,t} x_i + c_{i,t} x_i^2 \tag{44}$$

instead of the learned RiccatiSSM coefficients. With this design, we aim to isolate the difference between the LRC and quadratic vector fields from differences in parameterization and training. So with these experiments, we aim to answer whether the two models express the same dynamics.

We compare four systems on identical input sequences:

- **LrcSSM-RK4** integrates the full LRC vector field Equation 39 using fourth-order Runge–Kutta with 64 substeps per model time step. We use this high-accuracy numerical solution as a reference trajectory.
- **LrcSSM-Euler** integrates the same LRC vector field with one explicit-Euler step per model time step, matching the discretization used by the original model.
- **RiccatiSSM-ZOH** integrates the matched quadratic field Equation 44. The coefficients are held constant within each time step, and the resulting Riccati flow is evaluated exactly using Equation 13.
- **RiccatiSSM-Euler** integrates the same matched quadratic field with one explicit-Euler step per model time step.

These four systems distinguish approximation error from integration error. The comparison between RiccatiSSM-ZOH and LrcSSM-RK4 measures the combined effect of replacing the LRC field with its local quadratic approximation while exactly integrating both held-input systems. The comparison between LrcSSM-Euler and LrcSSM-RK4 measures the explicit-Euler error of the original LRC dynamics. Comparing RiccatiSSM-Euler with LrcSSM-Euler evaluates the quadratic approximation under the same Euler integrator.

Over 512 steps, this divergence is $1.7 \cdot 10^{-5}$ in relative RMSE, roughly two orders of magnitude smaller than the $7.2 \cdot 10^{-3}$ error that LrcSSM already incurs by discretizing its own equation with an explicit Euler step. Truncating the dynamics at second order is therefore not the limiting approximation in this model: RiccatiSSM can track the continuous LRC equation more closely than LrcSSM's own update does.

B.3 Reachable constraint-coefficient space

Figure 4 visualizes the coefficient space induced by the free and LRC-tied Riccati parameterizations. In the LRC-tied variant, the Riccati coefficients are coupled through the Taylor-matching relations

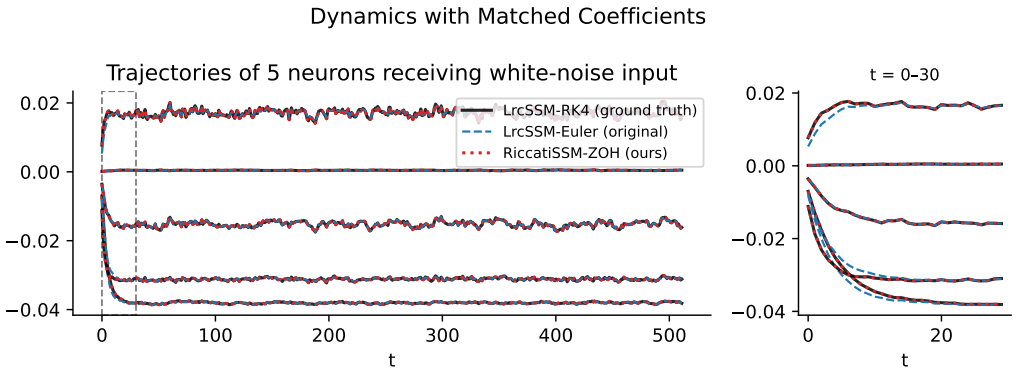


Figure 3: Hidden-state trajectories of 5 neurons under three models with matched Taylor coefficients driven by white-noise input. LrcSSM-RK4 (black, 64 RK4 substeps) serves as ground truth for the full dynamics; LrcSSM-Euler (blue dashed) is the model's single Euler step as in (Farsang & Grosu, 2025); RiccatiSSM-ZOH (red dotted) solves the quadratic (Riccati) truncation exactly via zero-order hold. The inset shows the first 30 timesteps (dashed box). Close match between RiccatiSSM-ZOH and both LrcSSM variants indicates that the dominant discretisation error comes from the Euler integrator rather than the quadratic truncation of the vector field.

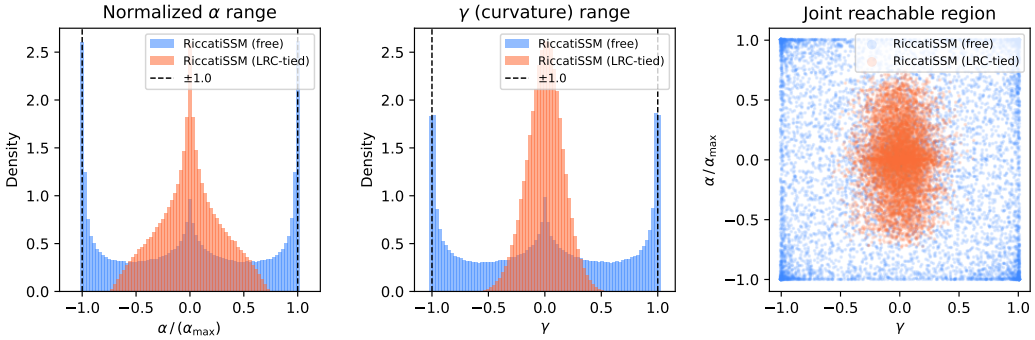


Figure 4: Reachable constraint-coefficient space under free vs. LRC-tied Riccati parameterizations, sampled over 500k random initializations. (Left) Distribution of normalized stability coefficient α/α_{max}: the free variant saturates at ± 1, while the LRC-tied variant concentrates near zero due to double-squashing through the circuit nonlinearities. (Center) Distribution of curvature coefficient γ: LRC-tied outputs are similarly compressed relative to the free range. (Right) Joint $(\gamma, \alpha/\alpha_{max})$ reachable region; the LRC-tied parameterization occupies a strictly smaller subspace, reflecting the structural constraints imposed by the underlying neural circuit dynamics.

derived in Section B.1, restricting the combinations of coefficients that can be realized. The default RiccatiSSM instead learns independent projections to the coefficients before applying the stability constraints, allowing it to explore a broader region of the admissible Riccati parameter space. This illustrates the additional flexibility of the free parameterization underlying the performance comparison in Section 4.4.

C Additional Experimental Details

Our codebase builds on the implementations of Rusch & Rus (2025) and Farsang & Grosu (2025). For the Weather dataset, we additionally build on the code from Nazari et al. (2025). All experiments were run on NVIDIA A40 and A100 GPUs.

We performed a grid search over learning rates $\{10^{-5}, 10^{-4}, 10^{-3}\}$, hidden dimensions $\{16, 64, 128\}$, state-space dimensions $\{16, 64, 256\}$, and numbers of blocks $\{2, 4, 6\}$, following the

search space of Walker et al. (2024); Rusch & Rus (2025). For each dataset, we selected the configuration with the highest mean validation accuracy across five data splits generated using the same random seeds, ensuring direct comparability with the reported baselines. For the Weather dataset, we instead performed a random hyperparameter search over the same ranges as in (Rusch & Rus, 2025).

The best-performing hyperparameter configuration selected for each dataset is reported in Table 7.

Table 7: Optimized hyperparameters used for the classification, PPG-DaLiA and Weather datasets.

	lr	hidden dim.	state-space dim.	number of layers
Heart	10^{-3}	64	64	4
SCP1	10^{-4}	64	16	6
SCP2	10^{-3}	16	16	2
Ethanol	10^{-3}	64	16	6
Motor	10^{-3}	16	256	4
Worms	10^{-3}	16	16	4
PPG-DaLiA	10^{-3}	64	64	6
Weather	$6.8 \cdot 10^{-4}$	32	64	4

C.1 RUNTIME

Table 8 reports the runtime of the best-performing configuration selected for each dataset. These configurations may differ in architecture and model size, as they correspond to the reported benchmark results. RiccatiSSM achieves low runtimes across these configurations.

RiccatiSSM is slower than LrcSSM on SCP1 and Ethanol; however, these comparisons are not architecture-matched. On both datasets, the selected RiccatiSSM configuration uses six layers, whereas the selected LrcSSM configuration uses only two. For a direct comparison with LrcSSM under an identical architecture, we refer to Section 4.3 and Figure 2.

Table 8: Runtime in seconds for the models for 1000 training steps, using the best configuration for each. Values for the models are taken from Rusch & Rus (2025) and Farsang & Grosu (2025).

	NRDE	NCDE	Log-NCDE	LRU	S5	Mamba	S6	LinOSS-IMEX	LinOSS-IM	LrcSSM	RiccatiSSM
Heart	9539	1177	826	8	11	34	4	4	7	23	21
SCP1	1014	973	635	9	17	7	3	42	38	12	33
SCP2	1404	1251	583	9	9	32	7	55	22	15	12
Ethanol	2256	2217	2056	16	9	255	4	48	8	15	26
Motor	7616	3778	730	51	16	35	34	128	11	31	27
Worms	5386	24595	1956	94	31	122	68	37	90	33	23