FRACTIONAL-ORDER STATE SPACE MODELS WITH LEARNABLE MEMORY DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured state space models (SSMs) combine parallel sequence processing with compact recurrent inference, encoding memory through matrix-exponential responses. We introduce a broader state space framework with a learnable Caputo derivative order $\alpha \in (0, 1]$. The continuous formulation includes ordinary first-order SSM dynamics as the special case $\alpha = 1$. For $0 < \alpha < 1$, the state evolution explicitly incorporates nonlocal dependence on past states, with α controlling the history weighting and the algebraic long-time decay of stable relaxation modes. This formulation extends integer-order SSMs with an explicit, learnable mechanism for incorporating state history. Direct evaluation of this history dependence requires a growing cache and quadratic total work. We address these costs through two computational realizations derived from a shared convolution-quadrature discretization. For parallel training, contour sampling of the discrete transfer function and an inverse FFT construct an approximate kernel, which is applied by FFT convolution. For causal streaming inference, a recent-state buffer is combined with a sum-of-exponentials approximation of the discrete history tail. Both realizations approximate the same discrete fractional operator. For fixed model dimensions and compression settings, the streaming cache size is independent of the number of processed tokens, with the tail approximation fitted over a specified horizon. Experiments on Long Range Arena, raw-waveform Speech Commands classification, and BIDMC heart-rate estimation demonstrate the applicability of the framework across diverse sequence domains. Within this framework, controlled comparisons with $\alpha = 1$ test whether introducing fractional history can enhance the representational capacity of the corresponding integer-order model.

1 INTRODUCTION

Long-sequence modeling requires integrating information across widely separated time scales while remaining computationally efficient. This requirement arises in language modeling (Vaswani et al., 2017; Gu & Dao, 2023; Dao & Gu, 2024), speech processing (Warden, 2018; Goel et al., 2022), and physiological-signal modeling (Pimentel et al., 2016). Some predictions are dominated by recent observations, whereas others benefit from retaining information over substantially longer temporal horizons. The mechanism by which a model retains and discounts past information therefore provides an important inductive bias for sequence learning.

State space models (SSMs) (Gu et al., 2021b;a), provide an efficient foundation for long-sequence modeling. Structured linear time-invariant (LTI) SSMs support parallel sequence evaluation through convolution and streaming inference through compact recurrent states (Gu et al., 2021a). Their development combines structured initialization of state-space parameters with efficient kernel construction (Gu et al., 2020; 2021a). Diagonal parameterizations and multi-input–multi-output formulations offer further simplifications (Gu et al., 2022a; Smith et al., 2022). Related recurrent and convolutional architectures have expanded the range of efficient sequence models (Orvieto et al., 2023; Fu et al., 2022; Poli et al., 2023), while selective SSMs introduce input-dependent state updates (Gu & Dao, 2023; Dao & Gu, 2024).

Our starting point is the finite-dimensional, first-order LTI SSM. Its state equation is local in time: the preceding history is summarized by the current latent state. For a standard continuous-time first-order LTI SSM, the memory kernel is governed by a matrix exponential. Learning the state-

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.

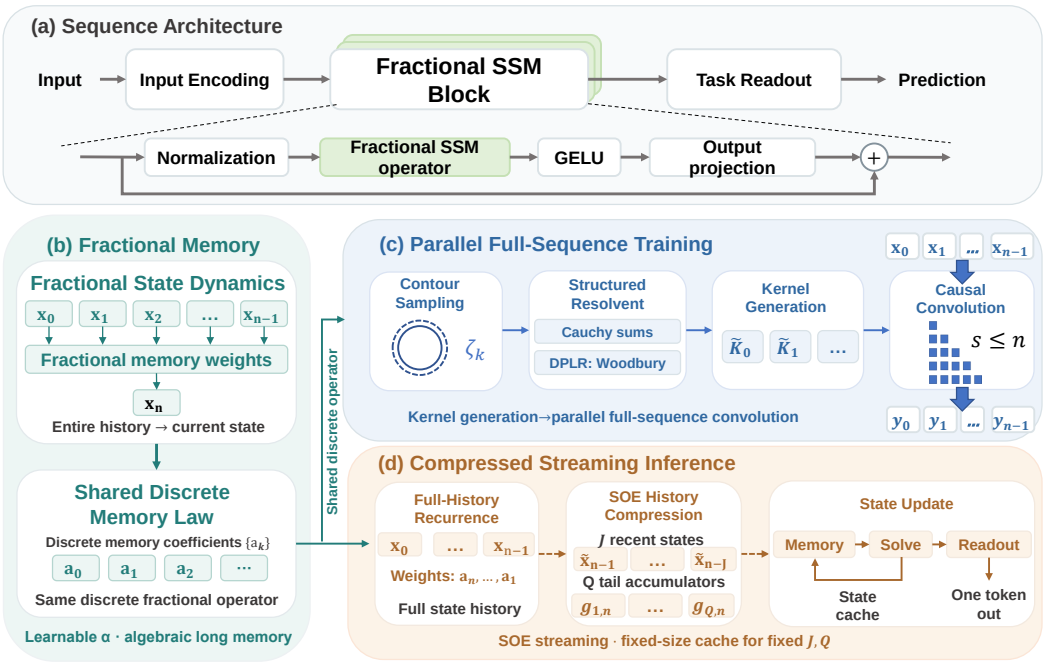

Figure 1: **Overview of Fractional SSM.** (a) Sequence architecture and a representative residual block. (b) Learnable fractional orders control history dependence; discretization yields the shared memory coefficients. (c) Structured resolvent evaluation and Fourier methods construct an approximate causal kernel, which is applied by FFT convolution for parallel full-sequence training. (d) A recent-state buffer and sum-of-exponentials (SOE) tail accumulators approximate the full-history recurrence for streaming inference.

space parameters adapts the time scales and combinations of the resulting responses, which can approximate slowly decaying dependencies over finite intervals. A complementary approach is to encode algebraically weighted history dependence directly in the state equation.

Fractional calculus provides a way to introduce this nonlocal history dependence. The first-order derivative can be generalized to a Caputo derivative of order $0 < \alpha \leq 1$ (Podlubny, 1998; Mainardi, 2022). For $0 < \alpha < 1$, the resulting state dynamics depend on the preceding state trajectory, and stable fractional relaxation modes exhibit Mittag–Leffler responses with algebraic long-time decay (Haubold et al., 2011). The order α controls the asymptotic decay exponent of these responses. At $\alpha = 1$, the Caputo derivative becomes the ordinary first-order derivative, recovering a continuous-time first-order SSM. This provides a broader state-space framework in which integer-order dynamics are retained as a special case and fractional order introduces explicit nonlocal history dependence. A central question is whether this additional history dependence can enrich the sequence representations learned by the corresponding integer-order model.

Fractional operators have been studied in neural differential equations and graph dynamics (Coelho et al., 2025; Kang et al., 2024). LongSpike incorporates fractional-order state-space modeling into spiking sequence models (He et al., 2026). FRACTAL (Li et al., 2026) introduces a fractional measure into the HiPPO construction while retaining first-order state dynamics and a preconfigured bank of singularity indices. These approaches illustrate different ways to incorporate fractional structure into learned dynamical models.

Directly adopting a nonlocal state equation introduces a computational challenge. Each update in a direct discretization accumulates contributions from the preceding history, so a naive implementation uses memory that grows with sequence length and incurs quadratic total work. Fast numerical methods can approximate power-law kernels using sums of exponentials (Beylkin & Monzón, 2010; Jiang et al., 2017). For sequence learning, these numerical tools must be integrated with differen-

tiable parameter learning, batched parallel evaluation, and accurate causal streaming. The challenge is to make the fractional history dependence computationally tractable in both evaluation settings.

This work develops a broader framework for LTI state-space sequence modeling by introducing a learnable Caputo derivative order. The continuous formulation includes ordinary first-order SSM dynamics as the special case $\alpha = 1$. A shared discretization defines the memory coefficients underlying both computational routes, as summarized in Figure 1. For parallel training, a spectral construction uses contour-based resolvent evaluation to approximate the causal kernel, which is applied to complete sequences using FFT convolution. For causal streaming inference, a recent-state buffer handles the most recent history directly, while an SOE approximation compresses the tail of the discrete history coefficients into auxiliary states. For fixed model dimensions, buffer length J, and number of tail modes Q, the streaming cache does not grow with the number of processed tokens. The tail approximation is constructed over a specified fitting horizon. Both routes approximate the same discrete fractional model, with numerical errors arising from their respective kernel construction and tail approximation.

Experiments on Long Range Arena, raw-waveform classification on Speech Commands, and heart-rate estimation on BIDMC demonstrate competitive predictive performance. Ablation studies examine the effect of the fractional-order configuration, including the choice between fixed and learnable orders.

Our contributions are as follows:

- We formulate a state-space sequence model with a learnable fractional derivative that introduces nonlocal history dependence. The continuous formulation recovers ordinary first-order SSM dynamics at $\alpha = 1$.
- We derive parallel and streaming approximations from a shared discrete fractional operator. A spectral kernel construction enables parallel sequence evaluation, while a recent-state buffer and an SOE tail approximation enable causal streaming with memory independent of the number of processed tokens for fixed model dimensions and approximation sizes.
- We evaluate the model on Long Range Arena, Speech Commands, and BIDMC heart-rate estimation, and use ablation studies to examine the effect of fractional-order configurations.

2 Related Work

State Space Models. State space sequence models capture temporal dependencies through latent-state dynamics (Gu et al., 2020; 2021b). Structured models such as S4 support efficient recurrent updates and parallel convolutional computation (Gu et al., 2021a; 2022b). Diagonal and simplified recurrent formulations further emphasize efficient parameterization, initialization, and normalization for long-range modeling (Gupta et al., 2022; Gu et al., 2022a; Smith et al., 2022; Orvieto et al., 2023). Other architectures introduce gating or input-dependent dynamics for content-dependent information propagation (Mehta et al., 2022; Fu et al., 2022; Gu & Dao, 2023; Dao & Gu, 2024). We investigate the differential order of state evolution as a complementary source of memory structure, while preserving a time-invariant convolutional core.

Fractional Dynamics and Memory Modeling. Fractional calculus provides history-dependent temporal operators, and stable scalar fractional relaxation exhibits algebraic tails for $0 < \alpha < 1$. Neural Fractional Differential Equations combine fractional dynamics with neural networks and support joint optimization of derivative orders and network parameters (Kang et al., 2024). Within SSMs, FRACTAL derives projection operators from fractional history measures to initialize a first-order SSM, using fixed singularity indices in its reported experiments (Li et al., 2026). LongSpike also employs Caputo fractional SSMs and exponential kernel approximations (He et al., 2026). Our focus is on the discretization of learnable fractional state dynamics and its parallel and streaming realizations. We learn the Caputo derivative order jointly with the state-space parameters and use convolution quadrature to define the discrete history operator. The spectral training kernel and the full-history causal recurrence are derived from this same discrete operator. When preparing a trained model for streaming inference, an SOE approximation is fitted to the discrete history weights beyond a recent-state window. This keeps the SOE approximation outside the training parameterization, allowing the streaming approximation to be configured after training without changing the learned model parameters.

3 PRELIMINARIES

3.1 STATE SPACE MODELS

State space models connect continuous dynamics with recurrent and convolutional sequence processing (Gu et al., 2021b). For one feature channel with latent state $x(t) \in \mathbb{R}^N$ and scalar input $u(t)$ and output $y(t)$, a linear time-invariant (LTI) SSM is

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{1}$$

The time-independent maps $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times 1}$, $C \in \mathbb{R}^{1\times N}$, and $D \in \mathbb{R}$ specify the state dynamics, input injection, readout, and direct feedthrough, respectively.

Discretization with step $\Delta > 0$ gives

$$x_n = \bar{A}x_{n-1} + \bar{B}u_n, \qquad y_n = Cx_n + Du_n, \tag{2}$$

where $\bar{A}, \bar{B}$ depend on the discretization rule. Common choices include the bilinear transform and zero-order hold (ZOH) (Gu et al., 2022a). For invertible A, ZOH gives

$$\bar{A} = \exp(\Delta A), \qquad \bar{B} = A^{-1}\big(\exp(\Delta A) - I\big)B. \tag{3}$$

With zero history before the first input, unrolling the recurrence yields

$$y_n = \sum_{j=0}^{n} K_j^{(1)} u_{n-j} + Du_n, \qquad K_j^{(1)} = C\bar{A}^j\bar{B}. \tag{4}$$

Once the first L kernel coefficients are available, a length-L sequence can be processed by FFT convolution in $O(L \log L)$ time per channel. Efficient kernel construction exploits diagonal or diagonal-plus-low-rank (DPLR) state matrices (Gu et al., 2021a; Gupta et al., 2022; Gu et al., 2022a). In a suitable basis, a tied DPLR state matrix is parameterized as $A = \mathrm{diag}(\Lambda) - PP^*$, where $\Lambda \in \mathbb{C}^N$, $P \in \mathbb{C}^{N\times r_A}$ with $r_A \ll N$, and P^* denotes the conjugate transpose of P. The input and output maps are expressed in the same basis. Matching conjugate pairs in the transformed parameters ensure real-valued outputs.

3.2 FRACTIONAL OPERATORS AND MEMORY

Fractional calculus extends differentiation from integer orders to non-integer orders. For absolutely continuous f and $0 < \alpha < 1$, the Caputo derivative is (Diethelm, 2010)

$${}^{\mathrm{C}}D_t^{\alpha} f(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} f'(\tau)\,\mathrm{d}\tau, \tag{5}$$

where Γ is the Gamma function. The operator weights past changes over $[0, t]$ by a power law controlled by α, with the integer endpoint defined as ${}^{\mathrm{C}}D_t^1 f = f'$. For $f(0) = 0$, the Caputo derivative satisfies

$$\mathcal{L}\big[{}^{\mathrm{C}}D_t^{\alpha} f(t)\big]\,(s) = s^{\alpha}\widehat{f}(s), \tag{6}$$

when the Laplace transform of f: $\widehat{f} = \mathcal{L}[f]$ exists.

Caputo dynamics use ordinary state values as initial conditions. Under suitable regularity, the initial-value problem ${}^{\mathrm{C}}D_t^{\alpha} z(t) = F(z(t), u(t))$, $z(0) = z_0$, for $0 < \alpha \leq 1$, is equivalent to the Volterra equation (Diethelm, 2010; Garrappa, 2018)

$$z(t) = z_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} F(z(\tau), u(\tau))\,\mathrm{d}\tau. \tag{7}$$

This form makes the dependence on the past trajectory explicit.

4 A MOTIVATING CAPACITY-CONTROLLED STUDY

Finite-dimensional LTI SSMs represent memory through finite sums of exponential modes. Increasing the state dimension introduces additional timescales, but does not change this exponential form. We therefore ask whether increasing exponential capacity is sufficient to reproduce an algebraic memory law beyond the training horizon. To isolate this question, we use the fractional-integration operator underlying ARFIMA$(0, d, 0)$ (Hosking, 1981; Wikipedia contributors, 2026). For white-noise input u_n, the target operator is

$$y_n^\star = (1-\mathcal{B})^{-d} u_n = \sum_{j=0}^{n} \psi_j(d) u_{n-j}, \qquad \psi_j(d) = \frac{\Gamma(j+d)}{\Gamma(d)\Gamma(j+1)} \sim \frac{j^{d-1}}{\Gamma(d)}. \tag{8}$$

We set $d = 0.4$, yielding an algebraic tail proportional to $j^{-0.6}$.

We compare matched single-channel diagonal integer- and fractional-order SSMs with $N \in \{8, 16, 32, 64\}$. Both variants use the same state parameterization, initialization, data, and optimization budget; the fractional variant additionally learns one scalar order α, while the integer variant fixes $\alpha = 1$. Models are trained only at $L_{\text{train}} = 1{,}024$ and evaluated up to $L_{\text{eval}} = 65{,}536$ ($64\times$ longer). Additional model, discretization, and training details are provided in Appendix B.

Table 1: Controlled ARFIMA operator learning. (a) State-dimension scaling at $L_{\text{eval}} = 65{,}536$. (b) Length extrapolation at $N = 64$.

(a) Capacity at $L_{\text{eval}} = 65{,}536$

	NMSE ↓		Relative gap
N	Fractional	Integer	$E_{\text{int}}/E_{\text{frac}}$
8	1.01×10^{-2}	4.79×10^{-2}	4.7
16	2.16×10^{-3}	8.50×10^{-2}	39.4
32	1.50×10^{-3}	4.63×10^{-2}	30.8
64	8.94×10^{-4}	4.72×10^{-2}	52.8

(b) Length extrapolation at $N = 64$

	NMSE ↓		Relative gap
L_{eval}	Fractional	Integer	$E_{\text{int}}/E_{\text{frac}}$
1,024	5.55×10^{-7}	2.70×10^{-5}	49.1
8,192	1.05×10^{-5}	9.84×10^{-3}	935.6
32,768	2.86×10^{-4}	3.65×10^{-2}	127.4
65,536	8.94×10^{-4}	4.72×10^{-2}	52.8

At $N = 64$, the integer model already attains a low NMSE of 2.70×10^{-5} at the training length, yet its error increases substantially under length extrapolation. At $L_{\text{eval}} = 65{,}536$, the fractional model obtains 8.94×10^{-4} NMSE compared with 4.72×10^{-2} for the integer model, a $52.8\times$ relative gap. The fractional model also achieves lower error across all tested state dimensions. These results suggest that enlarging a finite exponential basis is not equivalent to learning the form of an algebraic memory law, motivating the fractional state dynamics developed in Section 5.

5 FRACTIONAL STATE SPACE MODELS

We extend state-space dynamics with a learnable fractional order and define a shared discrete memory operator. Its convolutional form enables parallel FFT evaluation, while a sum-of-exponentials (SOE) approximation of its history weights yields a compact recurrence for streaming inference. Both computational forms approximate the same learned discrete operator.

5.1 FRACTIONAL MEMORY DYNAMICS

We replace the ordinary derivative in Eq. 1 with the Caputo derivative of order $0 < \alpha \leq 1$, assuming a zero continuous initial state $x(0) = 0$:

$$\begin{aligned} {}^{\text{C}}D_t^\alpha x(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{9}$$

Recall that the Volterra representation in Eq. 7 expresses the current state as a weighted integral of past dynamics $Ax(\tau) + Bu(\tau)$, and the order α controls this temporal weighting. To derive the Volterra representation of fractional analog dynamics, taking the Laplace transform of Eq. 9 gives

$$\begin{aligned} \mathcal{H}_c(s) &= C(s^\alpha I - A)^{-1} B, \\ Y(s) &= \big[\mathcal{H}_c(s) + D\big] U(s), \end{aligned} \tag{10}$$

where U, Y are the Laplace transforms of u, y. The dynamic convolution kernel ($\mathcal{L}^{-1}(\mathcal{H}_c(s))$) is

$$\mathcal{K}_\alpha(t) = Ct^{\alpha-1}E_{\alpha,\alpha}(At^\alpha)B, \qquad t > 0, \tag{11}$$

where $E_{\alpha,\beta}(Z) = \sum_{q\geq 0} Z^q/\Gamma(\alpha q + \beta)$ is the matrix Mittag–Leffler function. Thus, $y(t) = \int_0^t \mathcal{K}_\alpha(t-\tau)u(\tau)\,\mathrm{d}\tau + Du(t)$ remains a causal convolution. At $\alpha = 1$, the resolvent and kernel reduce to $(sI - A)^{-1}$ and $Ce^{tA}B$, recovering the standard continuous-time integer-order SSM. Fractional orders extend these exponential responses to a Mittag–Leffler family of memory kernels (Diethelm, 2010). The derivation of this input–output response is provided in Appendix D.1.

Orders are learned through a bounded sigmoid parameterization, with either a shared parameter or one parameter per channel. Each order remains fixed over time and common to all state coordinates within its channel. Each channel has a positive step Δ. The formulation supports the diagonal and DPLR state matrices introduced in Section 3.1 (Gu et al., 2021a; 2022a). The fixed-$\alpha = 1$ control is defined separately from the bounded learned parameterization.

Shared discrete operator. To process discrete sequences, we discretize the continuous-time transfer function in Eq. 10 using convolution quadrature (Lubich, 1988), with the substitution $s = \delta(\zeta)/\Delta$. Here ζ denotes a one-step delay, and δ specifies the discretization rule. The Tustin rule corresponds to $\delta(\zeta) = 2(1-\zeta)/(1+\zeta)$. Using the principal logarithm, we define the fractional symbol and the corresponding discrete transfer function:

$$F_\alpha(\zeta) = \exp\big(\alpha \operatorname{Log}\delta(\zeta)\big) = \sum_{j\geq 0} a_j \zeta^j, \tag{12}$$

$$\mathcal{H}_d(\zeta) = C\big[F_\alpha(\zeta)I - \Delta^\alpha A\big]^{-1}\Delta^\alpha B = \sum_{j\geq 0} K_j \zeta^j. \tag{13}$$

The coefficients a_j weight the current and past states in the discrete derivative, while K_j map past inputs to outputs. Under zero history, the output is $y_n = \sum_{j=0}^n K_j u_{n-j} + Du_n$. These two coefficient sequences connect parallel convolution and history-based recurrence to the same learned dynamics. Appendix D.2 derives their relationship and specifies the initialization convention.

At $\alpha = 1$, the fractional operator reduces to the integer-order SSM discretization.

5.2 Parallel Kernel Construction

For a zero-history input of length L, only $K_0, \ldots, K_{L-1}$ affect the first L outputs. We recover this prefix from $\mathcal{H}_d$ and apply it to the input by FFT convolution.

Contour sampling. The transfer function encodes an infinite kernel, so finite sampling folds later coefficients into the recovered prefix. To suppress this aliasing, we sample on a circle of radius $0 < r < 1$:

$$\zeta_k = r\exp(-2\pi ik/M), \qquad r = \exp(-\eta/M), \tag{14}$$

where $k = 0, \ldots, M-1$, $M \geq L$, and $\eta > 0$. Substituting ζ_k into the power-series expansion in Eq. 13 gives

$$\mathcal{H}_d(\zeta_k) = \sum_{j=0}^{\infty} K_j r^j \exp(-2\pi ikj/M). \tag{15}$$

The factor r^j suppresses distant coefficients, while sampling inside the unit disk avoids the fractional symbol's boundary singularities. We require $\mathcal{H}_d$ to be analytic on a neighborhood of the closed disk $|\zeta| \leq r$. Diagonal and DPLR structures reduce each evaluation to modal sums, with a Woodbury correction for DPLR (Gu et al., 2021a). The samples can be evaluated independently; Appendix D.4 gives the structured formulas and sufficient analyticity conditions.

Kernel recovery and convolution. An inverse FFT recovers the aliased, radius-weighted coefficients. Removing the known radial weight gives the kernel approximation:

$$\widetilde{K}_n = r^{-n}\,\mathrm{IFFT}_M[\mathcal{H}_d(\zeta_k)]_n\,, \qquad 0 \leq n < L. \tag{16}$$

In exact arithmetic, finite sampling produces the aliasing error

$$\widetilde{K}_n - K_n = \sum_{q\geq 1} r^{qM} K_{n+qM} = \sum_{q\geq 1} e^{-q\eta} K_{n+qM}. \tag{17}$$

At fixed M, increasing η suppresses tail contributions but amplifies numerical errors through $r^{-n} = \exp(\eta n/M)$. At fixed η and L, increasing M moves aliased coefficients farther into the tail and reduces this amplification. Appendix D.5 derives the identity and its numerical implications.

We then zero-pad $\widetilde{K}$ and u to a separate convolution length $L_c \geq 2L-1$ and compute

$$y^{\mathrm{FFT}} = \Big[\mathrm{IFFT}_{L_c}\Big(\mathrm{FFT}_{L_c}(\widetilde{K}) \odot \mathrm{FFT}_{L_c}(u)\Big)\Big]_{0:L} + Du. \tag{18}$$

Here $\odot$ denotes elementwise multiplication and $[\,\cdot\,]_{0:L}$ retains the first L entries. The length L_c accommodates the linear convolution of two length-L sequences, preventing circular wrap-around. It is separate from the sampling count M and does not remove the kernel-extraction error.

During training, the kernel is constructed from the current parameters and shared across the mini-batch. This differentiable construction enables learning of α and Δ alongside the state-space parameters, without unrolling the full-history state recurrence.

5.3 Constant-memory Streaming Inference

Streaming evaluates the shared discrete operator as inputs arrive. Under zero history, define $S = a_0 I - \Delta^\alpha A$ and $h_n = \sum_{j=1}^{n} a_j x_{n-j}$. Assuming S is nonsingular, the exact discrete recurrence is

$$\begin{aligned} Sx_n &= \Delta^\alpha B u_n - h_n, \\ y_n &= Cx_n + Du_n. \end{aligned} \tag{19}$$

Zero history refers to states before the first input; the discrete state x_0 is computed from u_0. Direct evaluation of h_n costs $O(T^2)$ work and $O(T)$ stored states over T tokens at fixed model dimensions. We compress this growing history into a recent-state window and auxiliary states that summarize the older trajectory.

Compressing the history tail. We retain a_0 and the first J history coefficients exactly and approximate the remaining coefficients by Q signed geometric modes. Inspired by SOE history compression (Jiang et al., 2017), we fit the discrete derivative coefficients directly:

$$a_j \approx \sum_{m=1}^{Q} c_m \lambda_m^{j-J-1}, \qquad J < j < T, \quad |\lambda_m| < 1, \tag{20}$$

for a chosen horizon T and $0 \leq J < T$. Here $c_m \in \mathbb{R}$ and $\lambda_m \in (-1, 1)$. Factors with magnitude closer to one retain contributions over longer lags; negative factors represent alternating components, including those of the Tustin symbol. Their magnitudes correspond to fixed, logarithmically spaced decay times. Once the learned orders are fixed after training, the coefficients c_m are fitted by weighted least squares. Increasing J preserves more recent weights explicitly, while increasing Q enriches the tail representation; $J = 0$ compresses all nonzero lags.

Updating the compressed state. The auxiliary vectors $g_{m,n}$ summarize history beyond the recent window. With $g_{m,-1} = 0$ and $\widetilde{x}_n = 0$ for $n < 0$, the compressed updates are

$$\begin{aligned} g_{m,n} &= \lambda_m g_{m,n-1} + \widetilde{x}_{n-J-1}, \\ \widetilde{h}_n &= \sum_{j=1}^{\min(J,n)} a_j \widetilde{x}_{n-j} + \sum_{m=1}^{Q} c_m g_{m,n}, \\ S\widetilde{x}_n &= \Delta^\alpha B u_n - \widetilde{h}_n, \\ \widetilde{y}_n &= C\widetilde{x}_n + Du_n. \end{aligned} \tag{21}$$

Only the history representation is approximated, with errors propagating through subsequent state updates. For fixed J, Q, the window and auxiliary vectors form a finite-dimensional recurrent state carried across chunks. The condition $|\lambda_m| < 1$ alone does not ensure stability of the coupled recurrence; Appendix D.6 provides the derivation and a conditional trajectory-error bound. For Tustin at $\alpha = 1$, an exact finite recurrence is given in Appendix D.3.

Table 2: Results on the Long Range Arena (LRA) benchmark. Results of previous methods are taken from S4 (Gu et al., 2021a) and S4D (Gu et al., 2022a). We take the 50% accuracy for the absence of Path-X accuracy as did in the work of S4D, then compute the overall average metrics across all tasks as AVG.

Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	AVG
Transformer (Vaswani et al., 2017)	36.37	64.27	57.46	42.44	71.40	–	53.66
BigBird (Zaheer et al., 2020)	36.05	64.02	59.29	40.83	74.87	–	54.18
Linear Transformer (Katharopoulos et al., 2020)	16.13	65.90	53.09	42.34	75.30	–	50.46
Performer (Choromanski et al., 2020)	18.01	65.40	53.82	42.77	77.05	–	51.18
FNet (Lee-Thorp et al., 2022)	35.33	65.11	59.61	38.67	77.80	–	54.42
S4-LegS (Gu et al., 2021a)	59.60	86.82	90.90	88.65	94.20	**96.35**	86.09
S4D-Legs (Gu et al., 2022a)	60.47	86.18	89.46	88.19	93.06	91.95	84.89
FracSSM(ours)	**63.80**	**91.12**	**92.29**	**92.08**	**96.17**	96.21	**88.61**

5.4 Fractional SSM Architecture

An embedding or linear encoder maps a length-L input to $U^{(0)} \in \mathbb{R}^{L\times H}$, where H is the hidden width. A representative residual block is

$$U^{(\ell+1)} = U^{(\ell)} + \mathrm{Proj}\Big(\mathrm{GELU}\Big(\mathrm{FracSSM}\Big(\mathrm{Norm}(U^{(\ell)})\Big)\Big)\Big). \tag{22}$$

The fractional operator models temporal dependencies within each channel, and the projection mixes channels. The operator is LTI, while the complete residual block is nonlinear. Task-specific heads produce position-wise or aggregated sequence-level outputs. Streaming inference uses causal operators with position-wise normalization and projection.

For one block processing a single sequence, compressed history requires $O(HN(J{+}Q{+}1))$ storage, including the current state, and $O(HN(J+Q))$ history-accumulation work per step. At fixed model dimensions and $M \propto L$, parallel kernel construction and FFT convolution cost $O(L\log L)$. Preparation and state-solve costs are given in Appendix D.7.

6 Experiments

We evaluate FracSSM on LRA, BIDMC heart-rate estimation, and Speech Commands classification, covering symbolic, textual, visual, physiological, and acoustic sequences. We further conduct ablations to examine the role of fractional order in memory modeling. All experiments are conducted on NVIDIA H100 GPUs with 80 GB memory.

6.1 Results

For fair comparison with existing linear time-invariant (LTI) SSM baselines, we use time-invariant state-space dynamics throughout. Dataset and experimental details are provided in Appendix C. Results on LRA, BIDMC, and SC10 are reported in Tables 2 and 3.

Long Range Arena. Table 2 compares FracSSM with representative sequence models on LRA. FracSSM achieves the highest reported accuracy among the compared models on five of the six tasks. It exceeds the strongest listed baselines on Text and Image by 4.30 and 3.43 percentage points, respectively, and also leads on Pathfinder. On Path-X, FracSSM achieves 96.21%, only 0.14 percentage points below S4-LegS.

Table 3: BIDMC HR and SC10 results. Baselines from Gu et al. (2022a) and Gu et al. (2021a).

Model	BIDMC RMSE ↓	SC10 Acc. (%) ↑
NRDE (Morrill et al., 2021)	2.97	16.49
CKConv (Romero et al., 2021)	2.05	71.66
S4 / S4-LegS (Gu et al., 2021a)	0.332 (0.013)	98.32
FracSSM (ours)	**0.306 (0.019)**	**98.39**

BIDMC Heart-Rate Estimation. Table 3 compares FracSSM with representative sequence models on BIDMC heart-rate estimation. FracSSM achieves an RMSE of 0.306 ± 0.019, yielding the

lowest reported mean among the compared methods. This result demonstrates the effectiveness of FracSSM for continuous-valued regression from physiological signals.

Speech Commands. Table 3 reports raw-waveform classification results on SC10. FracSSM achieves a test accuracy of 98.39%, outperforming the reported S4 accuracy of 98.32% (Gu et al., 2021a). It also exceeds the other raw-waveform baseline results listed in the table. This result demonstrates competitive performance on command classification from long audio sequences.

6.2 Fractional Order Ablation

We study the effect of fractional order by varying the order configuration while keeping the model architecture and optimization settings unchanged. Results are reported in Table 4. Performance is not monotonic with respect to α: a broad fractional-order range, $\alpha \in [0.3, 0.7]$, achieves consistently strong performance, outperforming the integer-order baseline ($\alpha = 1$), while the smaller-order setting $\alpha = 0.1$ provides no additional gains. This indicates that fractional order does not simply increase memory persistence, but instead changes the decay structure of historical information. The learnable-order model also achieves strong performance, showing that the fractional order can be optimized jointly with the remaining model parameters without requiring manual selection of a single fixed order. Finally, we examine whether this improvement can be explained purely by timescale adjustment. With $\alpha = 1$ fixed, increasing the step range improves performance from 88.94% to 89.87%, while decreasing it reduces performance to 87.36%. Thus, timescale tuning explains part of the improvement, but fractional order provides an additional degree of freedom for shaping memory decay beyond conventional timescale selection.

Table 4: Fractional order ablation. All models share the same architecture and training configuration. Δ denotes the difference from the $\alpha = 1$.

Configuration	Accuracy (%)	Δ vs. $\alpha = 1$
Fixed fractional order		
$\alpha = 0.1$	88.40	-0.54
$\alpha = 0.3$	90.41	+1.47
$\alpha = 0.5$	90.19	+1.25
$\alpha = 0.7$	90.35	+1.41
$\alpha = 0.9$	89.30	+0.36
$\alpha = 1$	88.94	–
Order learning and timescale control		
Learnable α	90.30	+1.36
$\alpha = 1$, shorter timescale	87.36	-1.58
$\alpha = 1$, longer timescale	89.87	+0.93

7 Conclusion

We presented a broader state space framework that extends integer-order SSMs through a learnable fractional derivative order. The continuous formulation recovers ordinary first-order dynamics at $\alpha = 1$, while explicitly incorporating nonlocal dependence on past states for $0 < \alpha < 1$. The fractional order controls how this history enters the state evolution, enriching the memory dynamics available within the framework. Controlled comparisons with $\alpha = 1$ assess the contribution of this additional history dependence and provide empirical support for its representational benefits relative to the corresponding integer-order model. From a shared convolution-quadrature discretization, we derive FFT-based kernel construction and convolution for parallel training, and sum-of-exponentials compression of the discrete history tail for causal streaming inference. Both computational routes approximate the same discrete fractional operator. For fixed model dimensions and compression settings, the streaming cache is independent of the number of processed tokens, with the tail approximation fitted over a specified horizon. Experiments on Long Range Arena, raw-waveform Speech Commands classification, and BIDMC heart-rate estimation demonstrate the applicability of the framework across diverse sequence domains. The current framework is limited to linear time-invariant dynamics; extending fractional-order state space models to linear time-varying dynamics is an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to aid manuscript preparation, including improving writing clarity, grammar, and overall presentation. Generative AI tools were not used for developing the research methodology, conducting experiments, generating experimental results, or drawing scientific conclusions. All AI-assisted content was carefully reviewed and revised by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts presented in the manuscript.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Gregory Beylkin and Lucas Monzón. Approximation by exponential sums revisited. *Applied and Computational Harmonic Analysis*, 28(2):131–149, 2010.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Cecilia Coelho, M Fernanda P Costa, and Luis L Ferras. Neural fractional differential equations: Optimising the order of the fractional derivative. *Fractal and Fractional*, 8(9):529, 2024.

Cecılia Coelho, M Fernanda P Costa, and Luis L Ferrás. Neural fractional differential equations. *Applied Mathematical Modelling*, 144:116060, 2025.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Kai Diethelm. *The analysis of fractional differential equations: An application-oriented exposition using differential operators of Caputo type*. springer, 2010.

Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Roberto Garrappa. Numerical solution of fractional differential equations: A survey and a software tutorial. *Mathematics*, 6(2):16, 2018.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It's raw! audio generation with state-space models. In *International conference on machine learning*, pp. 7616–7633. PMLR, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in neural information processing systems*, 35:35971–35983, 2022a.

Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022b.

Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in neural information processing systems*, 35:22982–22994, 2022.

Hans J Haubold, Arak M Mathai, and Ram K Saxena. Mittag-leffler functions and their applications. *Journal of applied mathematics*, 2011(1):298628, 2011.

Xinrui He, Qiyu Kang, Xuhao Li, and Zheng-Jun Zha. Longspike: Fractional order spiking state space models for efficient long sequence learning. *arXiv preprint arXiv:2606.12895*, 2026.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

J. R. M. Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981. ISSN 00063444. URL http://www.jstor.org/stable/2335817.

Shidong Jiang, Jiwei Zhang, Qian Zhang, and Zhimin Zhang. Fast evaluation of the caputo fractional derivative and its applications to fractional diffusion equations. *Communications in Computational Physics*, 21(3):650–678, 2017.

Qiyu Kang, Kai Zhao, Qinxu Ding, Feng Ji, Xuhao Li, Wenfei Liang, Yang Song, and Wee Peng Tay. Unleashing the potential of fractional calculus in graph neural networks with frond. In *International Conference on Learning Representations*, volume 2024, pp. 12816–12857, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

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.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, pp. 4296–4313, 2022.

Mario Lezcano-Casado and David Martınez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International conference on machine learning*, pp. 3794–3803. PMLR, 2019.

Mengqi Li, Wensheng Lin, Jinshuai Yang, and Lixin Li. Fractal: Ssm with fractional recurrent architecture for computational temporal analysis of long sequences. *arXiv preprint arXiv:2605.08833*, 2026.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2024.

Christian Lubich. Convolution quadrature and discretized operational calculus. i. *Numerische Mathematik*, 52(2):129–145, 1988.

Francesco Mainardi. *Fractional calculus and waves in linear viscoelasticity*. World Scientific Publishing Company, 2022.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*, 2022.

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.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36:43177–43201, 2023.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

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.

Marco AF Pimentel, Alistair EW Johnson, Peter H Charlton, Drew Birrenkott, Peter J Watkinson, Lionel Tarassenko, and David A Clifton. Toward a robust estimation of respiratory rate from pulse oximeters. *IEEE Transactions on Biomedical Engineering*, 64(8):1914–1923, 2016.

Igor Podlubny. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, volume 198. elsevier, 1998.

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 *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

David W Romero, Anna Kuzina, Erik J Bekkers, Jakub M Tomczak, and Mark Hoogendoorn. Ckconv: Continuous kernel convolution for sequential data. *arXiv preprint arXiv:2102.02611*, 2021.

T Konstantin Rusch and Siddhartha Mishra. Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. *arXiv preprint arXiv:2010.00951*, 2020.

T Konstantin Rusch and Siddhartha Mishra. Unicornn: A recurrent model for learning very long time dependencies. In *International Conference on Machine Learning*, pp. 9168–9178. PMLR, 2021.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.

Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12(1), 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Wikipedia contributors. Autoregressive fractionally integrated moving average — Wikipedia, the free encyclopedia, 2026. URL `https://en.wikipedia.org/w/index.php?title=Autoregressive_fractionally_integrated_moving_average&oldid=1357135086`. [Online; accessed 26-September-2026].

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

A Overview

This appendix supplements the motivating study, Preliminaries, Method, and experimental sections with additional experimental details, mathematical derivations, numerical analysis, and computational details for the proposed fractional state space model. It is organized as follows:

1. Details of the controlled ARFIMA motivating study are provided in Appendix B, including the target fractional-integration operator, capacity-controlled model setup, training and evaluation protocol, and additional diagnostic results.
2. Dataset and experimental details are provided in Appendix C.
3. The continuous-time response is derived in Appendix D.1, including the Mittag–Leffler convolution kernel and the distinction between initial-state relaxation and input–output responses.
4. The convolution-quadrature discretization is developed in Appendix D.2, which relates the discrete derivative weights to the input–output kernel and clarifies the initialization convention.
5. The integer-order limit and its exact finite Tustin recurrence are derived in Appendix D.3, together with an explanation of the role of input discretization when comparing SSM recurrences.
6. Sufficient analyticity conditions and structured resolvent formulas for diagonal and DPLR state matrices are provided in Appendix D.4.
7. The parallel kernel construction is detailed in Appendix D.5, covering Fourier coefficient recovery, finite-sampling aliasing, numerical error amplification, zero-padded causal convolution, and differentiation with respect to model parameters.
8. The SOE history approximation and compressed streaming updates are derived in Appendix D.6, along with a conditional finite-horizon trajectory-error bound and an explanation of cache continuity across consecutive chunks.
9. The preparation costs, memory requirements, and computational complexity of the parallel and streaming realizations are analyzed in Appendix D.7, including structured state solves and residual-block operations.

B Details of the ARFIMA Motivating Study

This section provides additional details for the controlled ARFIMA study in Section 4, including the target operator, matched model configurations, training protocol, evaluation metrics, and additional diagnostic results.

ARFIMA and target operator. ARFIMA is a standard long-memory time-series model based on fractional differencing (Hosking, 1981; Wikipedia contributors, 2026). It generalizes ARIMA by allowing the differencing order to take non-integer values, resulting in slowly decaying long-range temporal dependence. This makes ARFIMA a convenient controlled setting for examining whether a sequence model can learn and extrapolate an algebraically decaying memory kernel.

We use the fractional-integration component of an ARFIMA$(0, d, 0)$ process as the target linear operator. Let $u_n \sim \mathcal{N}(0, 1)$ denote an independent white-noise innovation sequence. The target output is

$$y_n^\star = (1 - \mathcal{B})^{-d} u_n = \sum_{j=0}^{n} \psi_j(d) u_{n-j}, \tag{23}$$

where $\mathcal{B}$ denotes the one-step delay operator and

$$\psi_j(d) = \frac{\Gamma(j + d)}{\Gamma(d)\Gamma(j + 1)} \sim \frac{j^{d-1}}{\Gamma(d)}, \qquad j \to \infty. \tag{24}$$

We set $d = 0.4$, yielding $\psi_j(d) \sim j^{-0.6}/\Gamma(0.4)$. The target therefore has an algebraically decaying memory tail rather than a finite collection of exponential modes. Because the innovation sequence itself is supplied as input, the task directly evaluates approximation of a deterministic causal operator rather than autoregressive forecasting.

Controlled setup and training protocol. We compare single-channel diagonal integer- and fractional-order SSMs over the complete capacity sweep $N \in \{2, 4, 8, 16, 32, 64\}$. The two variants use the same state-space parameterization, data, initialization, discretization, and optimization procedure. No nonlinear activation, normalization, feature projection, channel mixing, or residual connection is used, so the state dimension N directly controls the size of the temporal state.

Both models learn their ordinary SSM parameters, including the state dynamics, residues, direct term, and step size. The only order-level distinction is α: the fractional model learns one shared scalar order initialized at $\alpha = 0.5$, whereas the integer-order control fixes $\alpha = 1$. No pole or timescale is selected using knowledge of the ARFIMA target.

For both variants, the $N/2$ diagonal decay-rate magnitudes are initialized logarithmically between $1/L_{\mathrm{train}}$ and 1, with zero imaginary components. The residue real parts are initialized to $1/(N/2)$, their imaginary parts and the direct term are initialized to zero, and the step size is initialized at $\Delta = 1$ and remains learnable. Thus, the integer-order baseline starts from a broad multi-timescale dictionary rather than a target-specific initialization.

Both variants use the same Grünwald–Letnikov (GL) discretization. This diagnostic discretization is separate from the convolution-quadrature/Tustin realization used by the main FracSSM framework in Section 5; here, the common GL scheme is used specifically to isolate the effect of the derivative order.

Training and validation sequences have length $L_{\mathrm{train}} = 1{,}024$. For each training seed, we generate 128 training sequences and 64 validation sequences. Models are optimized for at most $1{,}500$ steps with AdamW and batch size 32. The initial learning rates are 3×10^{-3} for the remaining model parameters, 1×10^{-3} for the SSM dynamics and step-size parameters, and 5×10^{-3} for α in the fractional model. Weight decay is zero. The learning rate is warmed up for 100 steps and then follows a cosine schedule with minimum ratio 0.05; gradients are clipped to norm 1. Validation NMSE is evaluated every 100 steps, the best checkpoint is retained, and early stopping uses a patience of 12 validation evaluations. We use two independent training seeds, $s \in \{0, 1\}$.

Evaluation and diagnostics. After training, checkpoints are frozen and evaluated without further optimization or adaptation. We generate 32 independent test sequences once at the maximum length $65{,}536$ and use prefixes of the same sequences for all shorter evaluation horizons. Since the target operator is causal, each prefix is identical to the target that would be generated independently at that length. The complete evaluation set is $L_{\mathrm{eval}} \in \{1{,}024,\ 2{,}048,\ 4{,}096,\ 8{,}192,\ 16{,}384,\ 32{,}768,\ 65{,}536\}$, so the maximum evaluation horizon is $64\times$ longer than the training horizon.

The primary metric is global normalized mean-squared error,

$$\mathrm{NMSE} = \frac{\|\widehat{y} - y^\star\|_2^2}{\|y^\star\|_2^2}, \tag{25}$$

where $\widehat{y}$ is the model output and $y^\star$ is the exact ARFIMA target. The main-text results are means over the two training seeds.

We additionally inspect the learned effective kernel by applying each frozen model to a unit impulse. The diagnostic measures include NMSE over the final $1{,}024$ output positions, the relative kernel error

$$E_K = \frac{\|\widehat{K} - K^\star\|_2}{\|K^\star\|_2}, \tag{26}$$

relative kernel errors within dyadic lag intervals $[2^b, 2^{b+1})$, and the low-frequency log–log slope of the learned transfer response. For the ARFIMA target, the corresponding low-frequency magnitude has asymptotic slope $-d$.

Additional results and interpretation. Table 5 reports the complete capacity sweep at the longest evaluation horizon. The main text omits $N = 2$ and $N = 4$ only for compactness.

The fractional model has lower long-horizon NMSE than the matched integer model at every tested state dimension, although the dependence on N is not monotonic. At $N = 64$, the integer model already attains a validation NMSE of 2.65×10^{-5} at the training horizon and approximately $2.70\times$

Table 5: Complete ARFIMA capacity sweep. NMSE is reported at the training horizon $L = 1{,}024$ and the longest evaluation horizon $L = 65{,}536$. Results are averaged across independent training runs.

	$L = 1{,}024$			$L = 65{,}536$		
N	Frac.	Int.	α	Frac.	Int.	$E_{\text{int}}/E_{\text{frac}}$
2	1.10×10^{-4}	1.84×10^{-1}	0.376	6.52×10^{-3}	6.11×10^{-1}	93.6
4	1.12×10^{-4}	4.47×10^{-2}	0.454	3.04×10^{-2}	3.45×10^{-1}	11.3
8	1.21×10^{-6}	1.16×10^{-4}	0.461	1.01×10^{-2}	4.79×10^{-2}	4.7
16	3.18×10^{-6}	2.53×10^{-5}	0.472	2.16×10^{-3}	8.50×10^{-2}	39.4
32	1.02×10^{-6}	1.73×10^{-5}	0.478	1.50×10^{-3}	4.63×10^{-2}	30.8
64	5.55×10^{-7}	2.65×10^{-5}	0.482	8.94×10^{-4}	4.72×10^{-2}	52.8

10^{-5} on the independent test set at the same length. Its long-horizon error therefore cannot be attributed simply to failure to fit the observed interval. After extrapolating to $L_{\text{eval}} = 65{,}536$, its NMSE increases to 4.72×10^{-2}, whereas the fractional model obtains 8.94×10^{-4}, corresponding to an error ratio of approximately 52.8.

The kernel diagnostics show the same qualitative behavior. At $N = 64$ and $L_{\text{eval}} = 65{,}536$, both models retain small relative error at short lags, whereas in the farthest dyadic lag interval the mean relative kernel error is approximately 0.404 for the fractional model and 1.000 for the integer model. Thus, neither model reproduces the target tail perfectly, but the fractional model preserves substantially more of the long-lag structure.

Across the complete sweep, the mean learned fractional orders range from approximately 0.376 to 0.482; for the configurations shown in the main-text capacity table, $N \in \{8, 16, 32, 64\}$, they range from approximately 0.461 to 0.482. These values need not equal the data-generating parameter $d = 0.4$, because α is optimized jointly with the state dynamics and step size over a finite training horizon rather than serving as a direct estimator of d.

This experiment should therefore be interpreted as a controlled diagnostic rather than a comprehensive statistical benchmark. It does not imply that finite sums of exponentials cannot approximate an algebraic kernel over a bounded interval. Instead, within the tested state dimensions, optimization setting, and training horizon, enlarging the finite exponential basis does not recover the same long-horizon behavior as the learned fractional-order model. This distinction motivates the fractional memory dynamics used by FracSSM.

C Experimental Details

This appendix provides dataset details and experimental settings for the six tasks of the Long Range Arena (LRA) benchmark, BIDMC heart-rate prediction, and raw-waveform classification on the ten-class Speech Commands subset (SC10). Training hyperparameters are summarized in Table 6.

C.1 Datasets Details

LRA. The Long Range Arena (LRA) benchmark (Tay et al., 2020) evaluates sequence models on long-context inputs with sequence lengths ranging from approximately 1K to 16K. We consider six tasks: ListOps, Text, Retrieval, Image, Pathfinder, and Path-X. These tasks span symbolic, textual, and visual modalities, covering hierarchical reasoning, text classification, document matching, image classification, and visual connectivity reasoning.

BIDMC Heart-Rate Estimation. The BIDMC dataset contains physiological recordings from critically ill patients. We use the heart-rate (HR) regression task considered in prior SSM work (Gu et al., 2022a), which estimates HR from electrocardiogram (ECG) and photoplethysmogram (PPG) sequences of length $4{,}000$. Each example maps a multichannel signal window to a single HR target, yielding a sequence-to-one regression problem.

Speech Commands. Speech Commands (Warden, 2018) contains short recordings of spoken English keywords collected from thousands of speakers. We use SC10, the ten-class subset adopted in prior sequence modeling work (Kidger et al., 2020). Following the raw-waveform input setting of S4 (Gu et al., 2021a), each example is represented as a one-second signal sampled at $16\,\mathrm{kHz}$, yielding a sequence of $16{,}000$ time steps. The task is to classify each waveform into one of the ten command categories, and performance is measured by test accuracy.

C.2 Hyperparameter settings

The hyperparameter settings for all experiments are summarized in Table 6. These settings cover the six tasks of the Long Range Arena (LRA) benchmark, BIDMC heart-rate prediction, and raw-waveform classification on the ten-class Speech Commands subset (SC10).

Table 6: Hyperparameters used to obtain the best FracSSM results on each dataset. H and N denote the model width and state size, respectively. BN and LN denote batch and layer normalization. LR is the base learning rate.

Dataset	Depth	H	N	Norm	Dropout	LR	Batch size	Epochs	Weight decay
ListOps	4	256	4	BN	0.02	0.001	32	40	0.05
Text	6	256	4	BN	0.09	0.0045	16	32	0.05
Retrieval	4	384	8	BN	0.05	0.004	64	30	0.05
Image	6	512	64	BN	0.20	0.005	50	200	0.10
Pathfinder	6	256	64	BN	0.00	0.004	64	200	0.05
Path-X	6	256	64	BN	0.10	0.002	16	50	0.05
SC10	8	128	64	BN	0.15	0.005	16	40	0.05
BIDMC (HR)	6	128	256	LN	0.00	0.005	32	500	0.05

D Derivations and Numerical Details

This appendix develops the memory law, parallel kernel evaluation, and compressed streaming realization in the order used in the main text. It includes the transform and coefficient derivations, the numerical conditions for the two evaluation routes, and their error analysis. Unless stated otherwise, equations describe one channel, parameters are assumed constant with respect to time, and norms are Euclidean vector norms and their induced matrix norms. Conjugate-paired complex coordinates are equivalent to the real realization in the main text.

D.1 Continuous Dynamics and Memory Responses

For an absolutely continuous trajectory and $0 < \alpha < 1$, Eq. 5 expresses the fractional derivative as a weighted integral of the past derivative. To distinguish the initial-state response from the input response, temporarily write $x(0) = x_{\mathrm{init}}$; the model in the main text sets this value to zero. The fractional integral $I^\alpha f(t) = \Gamma(\alpha)^{-1} \int_0^t (t-\tau)^{\alpha-1} f(\tau)\,\mathrm{d}\tau$ satisfies $I^\alpha {}^{\mathrm{C}}D_t^\alpha x = x - x_{\mathrm{init}}$. Indeed, substituting Eq. 5 and exchanging the two integrals reduces the inner integral to $\int_0^1 (1-v)^{\alpha-1} v^{-\alpha}\,\mathrm{d}v = \Gamma(\alpha)\Gamma(1-\alpha)$. Applying this identity to the state equation gives

$$x(t) = x_{\mathrm{init}} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \big(Ax(\tau) + Bu(\tau)\big)\,\mathrm{d}\tau. \tag{27}$$

This form makes the dependence on previous state trajectories explicit. It also distinguishes the derivative weight $(t-\tau)^{-\alpha}$ from the integral weight $(t-\tau)^{\alpha-1}$; neither is generally the input–output impulse response.

Laplace-domain representation. Let $X(s)$ and $U(s)$ denote Laplace transforms, under the usual regularity and growth assumptions for their existence. The Caputo operator is a convolution of x' with $t^{-\alpha}/\Gamma(1-\alpha)$, whose transform is $s^{\alpha-1}$. Combining this with $\mathcal{L}[x'] = sX - x_{\mathrm{init}}$ gives

$$\mathcal{L}\big[{}^{\mathrm{C}}D_t^\alpha x\big](s) = s^\alpha X(s) - s^{\alpha-1} x_{\mathrm{init}}. \tag{28}$$

Consequently,

$$X(s) = (s^\alpha I - A)^{-1}\big(BU(s) + s^{\alpha-1}x_{\text{init}}\big). \tag{29}$$

Our model uses $x_{\text{init}} = 0$, so multiplication by C and addition of $DU(s)$ give Eq. 10. The direct term D is excluded from both $\mathcal{H}_c$ and $\mathcal{H}_d$ throughout. Fractional powers use the principal branch, and the transform identities are first understood in a right half-plane where the transforms exist (Diethelm, 2010).

Deriving the matrix impulse response. For sufficiently large real s, the Mittag–Leffler series gives

$$\mathcal{L}\big[t^{\alpha-1}E_{\alpha,\alpha}(At^\alpha)\big](s) = \sum_{q=0}^{\infty} A^q s^{-\alpha(q+1)} = (s^\alpha I - A)^{-1}. \tag{30}$$

The identity extends within their common domain of analyticity. Multiplication by C and B proves Eq. 11. For nonzero initial state, the additional homogeneous term is $E_{\alpha,1}(At^\alpha)x_{\text{init}}$. Thus the homogeneous relaxation and the forced impulse response involve different Mittag–Leffler functions. For the zero-initialized model, the output is

$$y(t) = \int_0^t \mathcal{K}_\alpha(t-\tau)u(\tau)\,\mathrm{d}\tau + Du(t). \tag{31}$$

The direct term acts instantaneously and is kept separate from the dynamic kernel.

A scalar illustration of algebraic memory. Consider $A = -\beta$ with $\beta > 0$ and $0 < \alpha < 1$. Define $E_{\alpha,\nu}(z) = \sum_{k=0}^{\infty} z^k/\Gamma(\alpha k + \nu)$ and $E_\alpha = E_{\alpha,1}$. The unforced relaxation from a nonzero initial value is

$$x(t) = E_\alpha(-\beta t^\alpha)x_{\text{init}}, \qquad E_\alpha(-\beta t^\alpha) \sim \frac{t^{-\alpha}}{\beta\Gamma(1-\alpha)} \quad (t \to \infty). \tag{32}$$

This fractional relaxation law (Diethelm, 2010) illustrates the memory family; the nonzero initial value is used only in this example. For the zero-initialized, unit-input/unit-readout system, the impulse response instead satisfies

$$\begin{aligned} k_\alpha(t) &= t^{\alpha-1}E_{\alpha,\alpha}(-\beta t^\alpha) = -\frac{1}{\beta}\frac{\mathrm{d}}{\mathrm{d}t}E_\alpha(-\beta t^\alpha), \\ k_\alpha(t) &\sim \frac{\alpha}{\beta^2\Gamma(1-\alpha)}t^{-1-\alpha} \quad (t \to \infty). \end{aligned} \tag{33}$$

The derivative identity follows term by term from the Mittag–Leffler series; the displayed long-time powers follow from its asymptotic expansions. Thus relaxation and impulse responses have different powers. The $t^{-\alpha}$ relaxation tail and the $t^{-1-\alpha}$ impulse-response tail refer to different response functions and should not be conflated. At $\alpha = 1$ the scalar response is exponential. These scalar statements do not imply a single decay exponent for every learned multi-mode output: readout combinations can cancel leading terms.

D.2 Discrete Operator, Initialization, and Coefficients

Convolution-quadrature definition. Using the symbol $F_\alpha(\zeta) = \sum_{j\ge 0} a_j\zeta^j$ in Eq. 12, substitute $s = \delta(\zeta)/\Delta$ into Eq. 10. Since $\Delta > 0$ and the same logarithm branch is used,

$$\left(\frac{\delta(\zeta)}{\Delta}\right)^\alpha = \Delta^{-\alpha}F_\alpha(\zeta). \tag{34}$$

Multiplying the resolvent by Δ^α gives exactly Eq. 13. This is the convolution-quadrature definition of the discrete operator (Lubich, 1988). It defines the target for both numerical realizations in the main text; it does not identify the discrete states with exact continuous-time samples. The bounds below concern approximating this discrete target, rather than a convergence rate as $\Delta \to 0$. Such a rate additionally depends on the discretization rule, regularity, and treatment of the initial time.

Define the unilateral generating functions $X_d(\zeta) = \sum_{n\ge 0} x_n\zeta^n$ and $U_d(\zeta) = \sum_{n\ge 0} u_n\zeta^n$. The corresponding discrete state equation is

$$\big[F_\alpha(\zeta)I - \Delta^\alpha A\big]X_d(\zeta) = \Delta^\alpha BU_d(\zeta). \tag{35}$$

These identities can first be read as formal power series when the constant matrix coefficient is invertible. Analytic convergence is an additional requirement for numerical contour extraction.

Coefficient matching and the first state. The coefficient of ζ^n in Eq. 35 is

$$\sum_{j=0}^{n} a_j x_{n-j} = \Delta^\alpha A x_n + \Delta^\alpha B u_n. \tag{36}$$

Separating $j = 0$ gives

$$\begin{aligned} S = a_0 I - \Delta^\alpha A, \qquad h_n = \sum_{j=1}^{n} a_j x_{n-j}, \\ S x_n = \Delta^\alpha B u_n - h_n, \qquad y_n = C x_n + D u_n, \end{aligned} \tag{37}$$

which is Eq. 19. Conversely, multiplying this recurrence by ζ^n and summing recovers Eq. 35; applying the readout gives the dynamic transfer in Eq. 13. The convolution and full-history recurrence are therefore exact representations of the same discrete operator before numerical approximation. Under the discrete zero-prehistory convention, the first output state is

$$x_0 = (a_0 I - \Delta^\alpha A)^{-1} \Delta^\alpha B u_0. \tag{38}$$

Hence x_0 denotes the state produced by processing the first discrete input. The prescribed continuous value $x_{\mathrm{init}} = 0$ has already been used to remove the initial-condition term in Eq. 29. The discrete definition does not impose $x_0 = x_{\mathrm{init}}$. Changing this convention, the input discretization, or startup corrections changes the discrete input–output operator and must be applied consistently to both evaluation routes.

Computing the Tustin coefficients. For the Tustin symbol in Section 5.1,

$$F_\alpha(\zeta) = 2^\alpha (1-\zeta)^\alpha (1+\zeta)^{-\alpha}, \qquad |\zeta| < 1. \tag{39}$$

The factors use branches analytic in the unit disk and agree with the principal-power definition at $\zeta = 0$; hence the factorization holds throughout that disk. Logarithmic differentiation gives

$$(1-\zeta^2) F'_\alpha(\zeta) = -2\alpha F_\alpha(\zeta). \tag{40}$$

Matching powers of ζ yields

$$\begin{aligned} a_0 = 2^\alpha, \qquad a_1 = -2\alpha a_0, \\ a_{j+1} = \frac{-2\alpha a_j + (j-1) a_{j-1}}{j+1}, \qquad j \geq 1. \end{aligned} \tag{41}$$

Thus the first T convolution-quadrature coefficients require $O(T)$ arithmetic operations per distinct order. Their accuracy in floating-point arithmetic also depends on the horizon and precision. As a comparison, choosing backward Euler, $\delta(\zeta) = 1 - \zeta$, gives $a_j = (-1)^j \binom{\alpha}{j}$, with $a_0 = 1$ and $a_j = (j - 1 - \alpha) a_{j-1}/j$ for $j \geq 1$. The coefficient sequence must always match the symbol used for spectral kernel construction.

A direct reference for the input–output kernel. The derivative weights and the output kernel are different sequences. To connect them explicitly, define state impulse coefficients $V_j \in \mathbb{C}^{N \times 1}$ by

$$V_0 = S^{-1} \Delta^\alpha B, \qquad V_n = -S^{-1} \sum_{j=1}^{n} a_j V_{n-j} \quad (n \geq 1), \qquad K_n = C V_n. \tag{42}$$

These equations follow by applying Eq. 19 to the unit impulse $u_0 = 1$, $u_n = 0$ for $n > 0$. They give an exact finite-prefix reference in exact arithmetic, albeit with quadratic history work. In particular, $K_0 = C S^{-1} \Delta^\alpha B$, and the full instantaneous output coefficient is $K_0 + D$.

Order and step parameterization. A bounded learned order can be written as

$$\alpha_h = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \sigma(\theta_h), \qquad 0 < \alpha_{\min} < \alpha_{\max} \leq 1. \tag{43}$$

A shared order ties the corresponding θ_h values. Each step satisfies $\Delta_h > 0$. Order bounds, initialization, parameter sharing, and trainable versus frozen dynamics are configuration choices. The logistic function takes values strictly between zero and one, so the fixed-$\alpha = 1$ control is defined separately even if $\alpha_{\max} = 1$. A differentiable positive parameterization supports either learning Δ or fixing it as a model configuration. Derivatives with respect to α and Δ are given in Appendix D.5.

D.3 Integer-Order Limit and Discretization Conventions

At $\alpha = 1$, the continuous dynamics become the ordinary SSM in Eq. 1. The corresponding discrete realization also depends on how the input is discretized. This subsection makes that choice explicit and derives the exact finite recurrence for the Tustin endpoint.

The recurrence in the preliminaries. For the ordinary differential equation, holding u_n constant over an interval of length Δ and sampling the state at its end gives

$$\bar{A}_{\mathrm{ZOH}} = e^{\Delta A}, \qquad \bar{B}_{\mathrm{ZOH}} = \int_0^{\Delta} e^{\tau A} B \, \mathrm{d}\tau. \tag{44}$$

For invertible A, integrating the matrix exponential gives Eq. 3; the integral remains valid when A is singular. With the initial state indexed as $x_{-1} = 0$, induction in Eq. 2 yields $x_n = \sum_{j=0}^{n} \bar{A}^j \bar{B} u_{n-j}$, which proves Eq. 4. The same unrolling applies to any specified pair $(\bar{A}, \bar{B})$ with that indexing convention.

Exact Tustin endpoint. Set $\alpha = 1$ and multiply Eq. 35 by $1 + \zeta$. Then

$$\big[(2I - \Delta A) - (2I + \Delta A)\zeta\big] X_d(\zeta) = \Delta B(1 + \zeta) U_d(\zeta). \tag{45}$$

Coefficient matching gives

$$(2I - \Delta A)x_n = (2I + \Delta A)x_{n-1} + \Delta B(u_n + u_{n-1}), \qquad n \geq 0, \tag{46}$$

with $x_{-1} = 0$ and $u_{-1} = 0$. Define

$$R = (2I - \Delta A)^{-1}(2I + \Delta A), \qquad G = (2I - \Delta A)^{-1}\Delta B. \tag{47}$$

The update is $x_n = Rx_{n-1} + G(u_n + u_{n-1})$ and $y_n = Cx_n + Du_n$. It needs one state vector and the previous input, with no growing history or SOE fit. Its dynamic impulse coefficients are

$$K_0 = CG, \qquad K_n = CR^{n-1}(R + I)G \quad (n \geq 1). \tag{48}$$

Relation to a common bilinear SSM recurrence. A common bilinear parameterization, including that used in S4, is $\bar{A} = R$ and $\bar{B} = (I - \Delta A/2)^{-1}\Delta B = 2G$ (Gu et al., 2021a; 2022a). The recurrence $z_n = Rz_{n-1} + 2Gu_n$ has dynamic transfer $\mathcal{H}_{\mathrm{rec}}(\zeta) = C(I - R\zeta)^{-1}2G$. By Eq. 45, our Tustin substitution gives

$$\mathcal{H}_d(\zeta)|_{\alpha=1} = \frac{1 + \zeta}{2} \, \mathcal{H}_{\mathrm{rec}}(\zeta). \tag{49}$$

Thus the two dynamic recurrences agree when the latter is driven by $(u_n + u_{n-1})/2$. The direct output term remains Du_n. The integer-order limit recovers a standard first-order SSM discretization under the chosen input convention; identifying it with a particular existing implementation also requires matching the input convention, state indexing, and readout.

Finally, the Tustin derivative weights at this endpoint are $a_0 = 2$ and $a_j = 4(-1)^j$ for $j \geq 1$, from the geometric expansion of $2(1 - \zeta)/(1 + \zeta)$. Their nondecaying alternation does not prevent the exact finite recurrence in Eq. 46. A fixed-$\alpha = 1$ control can therefore use that recurrence rather than approximate the derivative tail by strictly decaying modes.

D.4 Analyticity and Structured Resolvents

To extract Taylor coefficients on a circle of radius r, we require $\mathcal{H}_d$ to be analytic on a neighborhood of the closed disk $|\zeta| \leq r$. Invertibility at the sampling points, or even everywhere on the circle, does not by itself exclude a pole inside that disk. The following gives a sufficient condition; it is not a necessary restriction on every admissible realization.

The fractional symbol. For the Tustin rule and $|\zeta| < 1$,

$$\mathrm{Re}\, \delta(\zeta) = \frac{2(1 - |\zeta|^2)}{|1 + \zeta|^2} > 0. \tag{50}$$

Hence δ has no zero or pole inside the unit disk, and its image avoids the branch cut of the principal logarithm. It follows that F_α is analytic there and $|\arg F_\alpha(\zeta)| < \alpha\pi/2 \leq \pi/2$ for $0 < \alpha \leq 1$, so $\mathrm{Re}\, F_\alpha(\zeta) > 0$. Backward Euler satisfies the same positive-real property because $\mathrm{Re}(1 - \zeta) > 0$ in the disk. A different discretization symbol requires its own branch and resolvent checks.

A sufficient matrix condition. Suppose the Hermitian part obeys $\frac{1}{2}(A + A^*) \preceq -\gamma I$ for some $\gamma \geq 0$. For any nonzero vector v,

$$\mathrm{Re}(v^*[F_\alpha(\zeta)I - \Delta^\alpha A]v) \geq \big(\mathrm{Re}\, F_\alpha(\zeta) + \Delta^\alpha\gamma\big)\|v\|_2^2 > 0. \tag{51}$$

The matrix is therefore nonsingular throughout $|\zeta| < 1$, including at $\zeta = 0$, where it equals S. Cauchy–Schwarz also gives

$$\big\|[F_\alpha(\zeta)I - \Delta^\alpha A]^{-1}\big\|_2 \leq \frac{1}{\mathrm{Re}\, F_\alpha(\zeta) + \Delta^\alpha\gamma}. \tag{52}$$

For every $r < 1$, analyticity on the open unit disk therefore supplies an analytic neighborhood of the closed extraction disk. For the tied DPLR realization $A = \mathrm{diag}(\Lambda) - PP^*$ with $P \in \mathbb{C}^{N\times r_A}$,

$$\tfrac{1}{2}(A + A^*) = \mathrm{diag}(\mathrm{Re}\,\Lambda) - PP^*, \tag{53}$$

so $\mathrm{Re}\,\Lambda_i \leq -\gamma$ is sufficient. The main-text contour assumption can also hold for matrices outside this sufficient class. The sufficient condition is conservative and is used only to guarantee analyticity of the exact discrete operator. The bound here concerns the exact discrete operator; it does not establish stability of an arbitrary fitted history approximation.

Frequency-domain evaluation. Let $d = \Delta^\alpha$, $w = F_\alpha(\zeta)/d$, and $R(w) = (wI - \mathrm{diag}(\Lambda))^{-1}$. When the inverses exist, Woodbury gives

$$\mathcal{H}_d(\zeta) = CR(w)B - CR(w)P\big(I + P^*R(w)P\big)^{-1}P^*R(w)B. \tag{54}$$

For diagonal dynamics the second term vanishes and the first is $\sum_{i=1}^N C_iB_i/(w - \Lambda_i)$. For low-rank P, the correction requires only a small matrix inverse at each frequency. This is the structured-resolvent principle used in S4 (Gu et al., 2021a), applied here at fractional frequency arguments. Conjugate pairing of A, B, C yields real coefficients and the symmetry $\mathcal{H}_d(\overline{\zeta}) = \overline{\mathcal{H}_d(\zeta)}$. At a complex contour argument, omitted modal partners must be evaluated at the same w; taking twice the real part of a half-mode sum is generally incorrect. The formulas here use tied factors PP^*; for an untied DPLR matrix, the Woodbury factors and the sufficient dissipativity condition must be adjusted accordingly.

D.5 Contour Extraction, Aliasing, and Causal Convolution

Coefficient identity. Under the analyticity assumption in Appendix D.4, the Taylor series converges absolutely on the extraction circle. Let $\zeta_k = r\exp(-2\pi ik/M)$ as in Eq. 14. Substituting these points into Eq. 13 gives Eq. 15. The coefficients are weighted by r^j before Fourier sampling. Using the inverse-FFT normalization $1/M$, for $0 \leq n < M$,

$$\begin{aligned}\mathrm{IFFT}_M[\mathcal{H}_d(\zeta_k)]_n &= \frac{1}{M}\sum_{k=0}^{M-1}\sum_{\ell\geq 0} K_\ell r^\ell \exp\left(\frac{2\pi ik(n-\ell)}{M}\right)\\ &= \sum_{q\geq 0} K_{n+qM}r^{n+qM}.\end{aligned} \tag{55}$$

The second equality uses discrete Fourier orthogonality. Dividing by r^n gives Eq. 16; subtracting the $q = 0$ term proves Eq. 17, since $r^{qM} = e^{-q\eta}$. This error comes from recovering coefficients using finitely many frequency samples. Cropping to the first L exact coefficients would introduce no error in the first L causal outputs for a zero-prehistory input of length L.

Damping and oversampling. For $r = \exp(-\eta/M)$, each alias order is weighted by $r^{qM} = \exp(-q\eta)$, whereas radial rescaling at index n amplifies a transform error by $e^{\eta n/M}$. Increasing η suppresses alias terms but raises this amplification. At fixed η, increasing M moves the aliased coefficients farther into the tail and reduces de-damping over the retained prefix; it does not change the factor $\exp(-q\eta)$ itself. Decay and conditioning of the learned kernel determine the resulting numerical tradeoff; an increase in M does not, by itself, imply a prescribed error reduction.

A quantitative bound follows from analyticity. If $r < R < 1$ and the transfer is analytic on a neighborhood of $|\zeta| \leq R$, set $\mathcal{B}_R = \max_{|\zeta|=R}|\mathcal{H}_d(\zeta)|$. Cauchy's coefficient estimate gives

$$|\widetilde{K}_n - K_n| \leq \mathcal{B}_R R^{-n}\frac{(r/R)^M}{1 - (r/R)^M}, \qquad 0 \leq n < L. \tag{56}$$

The bound follows by inserting $|K_j| \le \mathcal{B}_R R^{-j}$ into Eq. 17 and summing a geometric series. No analyticity beyond the unit circle is assumed. This sufficient bound may be conservative. If each sampled transfer value has an additional perturbation of magnitude at most $\varepsilon_{\mathrm{samp}}$, exact inverse transformation and de-damping add at most $r^{-n}\varepsilon_{\mathrm{samp}}$ to coefficient n. Roundoff in the FFT and convolution contributes separately.

Applying the recovered kernel. For two length-L sequences, the complete linear convolution is

$$v_n = \sum_{j=\max(0,n-L+1)}^{\min(n,L-1)} \widetilde{K}_j u_{n-j}, \qquad 0 \le n \le 2L-2. \tag{57}$$

There are $2L-1$ potentially nonzero entries because the largest sum of two input indices is $(L-1)+(L-1) = 2L-2$. An FFT product instead computes a circular convolution: for a transform length L_c, it adds together coefficients whose indices differ by a multiple of L_c. Choosing $L_c \ge 2L-1$ prevents that folding, giving the linear convolution in Eq. 18 after padding. We then keep only $v_0, \dots, v_{L-1}$ and add Du.

Even when only these first L outputs are needed, a direct length-L FFT product would add v_{n+L} to v_n wherever that later term exists. Those discarded linear-convolution entries can therefore corrupt the retained prefix if padding is omitted. Specialized blocked convolution algorithms can use other transform lengths with suitable overlap handling; Eq. 18 uses a single padded product. The sampling count M controls kernel extraction, whereas L_c prevents convolution wrap-around. Padding does not remove the error in Eq. 17.

For a fixed input with $U_{\max} = \max_{0\le n<L} |u_n|$, exact convolution with the approximate kernel satisfies

$$\max_{0\le n<L} |y_n^{\mathrm{FFT}} - y_n| \le U_{\max} \sum_{j=0}^{L-1} |\widetilde{K}_j - K_j|. \tag{58}$$

This bound excludes floating-point error in the subsequent convolution. It separates the coefficient-extraction approximation from the exact finite causal prefix and from numerical FFT error.

Differentiating the parallel operator. At points where the chosen logarithm is analytic, and holding the other independent parameters fixed,

$$\partial_\alpha F_\alpha = F_\alpha \operatorname{Log}\delta, \qquad \partial_\alpha \Delta^\alpha = \Delta^\alpha \log\Delta, \qquad \partial_\Delta \Delta^\alpha = \alpha \Delta^{\alpha-1}. \tag{59}$$

Both the symbol and step scaling contribute to the order gradient. To see their combined effect, let $w = (\delta(\zeta)/\Delta)^\alpha$ and $Z = (wI - A)^{-1}$, so $\mathcal{H}_d = CZB$. Differentiating the inverse with A, B, C held fixed gives

$$\begin{aligned} \partial_\alpha \mathcal{H}_d(\zeta) &= -w \operatorname{Log}\left(\frac{\delta(\zeta)}{\Delta}\right) CZ^2B, \\ \partial_\Delta \mathcal{H}_d(\zeta) &= \frac{\alpha w}{\Delta} CZ^2B. \end{aligned} \tag{60}$$

For fixed (M, η), coefficient recovery is a linear map of the sampled transfer values. The resulting derivatives therefore pass through the inverse FFT and radial rescaling, followed by the convolution. The bounded order parameterization adds the usual sigmoid chain rule. These identities describe differentiation of the finite spectral computation used during training.

D.6 History Compression and Trajectory Error

Finite-horizon fit. Let T be the number of intended inference steps, $0 \le J < T$, and let $\lambda_m \in (-1, 1)$ be fixed geometric factors. A logarithmic decay-time grid uses $\lambda_m = \varepsilon_m \exp(-1/\tau_m)$, where $\varepsilon_m \in \{-1, +1\}$ and the positive decay times τ_m are logarithmically spaced. For positive fit weights w_j, the residues solve

$$(c_1, \dots, c_Q) \in \operatorname*{argmin}_{(v_1,\dots,v_Q)\in\mathbb{R}^Q} \sum_{j=J+1}^{T-1} w_j \left| a_j - \sum_{m=1}^{Q} v_m \lambda_m^{j-J-1} \right|^2. \tag{61}$$

Equivalently, form $V_{j,m} = \lambda_m^{j-J-1}$ and $W = \mathrm{diag}(w_{J+1}, \ldots, w_{T-1})$. The problem is $\min_c \|W^{1/2}(Vc - a_{\mathrm{tail}})\|_2^2$. A QR or singular-value decomposition solves this problem without forming normal equations; if the minimizer is nonunique, the minimum-norm least-squares solution specifies one choice. When the same factors and weights are used for several channels, their design-matrix factorization can be reused. Only α and the selected symbol determine a_j; A, B, C, and Δ do not enter this dimensionless coefficient fit. For $J = T - 1$, the tail is empty and no fit is needed on the specified horizon. More generally, $J = 0$ compresses every nonzero lag. This construction is inspired by exponential-history compression (Jiang et al., 2017), but fits the coefficients of the selected discrete symbol. The continuous-kernel approximation and PDE-specific error guarantees in that work do not directly establish guarantees for this signed discrete fit. A negative λ_m introduces an alternating factor at successive integer lags, not an unstable positive growth rate. Both the rate grid and the fit weights affect finite-horizon accuracy; residues are allowed to have either sign. The decay times are measured in token lags: $|\lambda_m|^\ell = e^{-\ell/\tau_m}$. The number of positive and negative factors, the endpoints of the logarithmic grids, and the weights w_j specify the fitting configuration. The analysis does not require particular numerical values for them.

For Tustin, the singularity of $(1+\zeta)^{-\alpha}$ at $\zeta = -1$ produces alternating derivative coefficients. Expanding the other factor at $\zeta = -1$ gives, for $0 < \alpha < 1$,

$$a_j = (-1)^j \frac{2^{2\alpha}}{\Gamma(\alpha)} j^{\alpha-1}(1 + o(1)), \qquad j \to \infty. \tag{62}$$

Indeed, $F_\alpha(\zeta) \sim 2^{2\alpha}(1+\zeta)^{-\alpha}$ at this singularity, whose coefficients are $(-1)^j\Gamma(j+\alpha)/(\Gamma(\alpha)\Gamma(j+1))$; the singularity at $\zeta = 1$ contributes the smaller order $O(j^{-1-\alpha})$. These are derivative weights, not the input–output coefficients K_j. Their alternating tail decays especially slowly as α approaches one. Positive geometric factors alone do not directly encode that alternating component. The symbol remains analytic strictly inside the unit disk, so this boundary behavior is compatible with Appendix D.4.

Deriving the auxiliary-state update. For the approximate trajectory, define

$$g_{m,n} = \sum_{j=J+1}^{n} \lambda_m^{j-J-1} \widetilde{x}_{n-j}, \tag{63}$$

where an empty sum is zero. Reindexing the sum for $g_{m,n-1}$ gives

$$\begin{aligned} \lambda_m g_{m,n-1} &= \sum_{j=J+2}^{n} \lambda_m^{j-J-1} \widetilde{x}_{n-j}, \\ g_{m,n} &= \lambda_m g_{m,n-1} + \widetilde{x}_{n-J-1}. \end{aligned} \tag{64}$$

Substituting Eq. 20 into the tail sum yields

$$\sum_{j=J+1}^{n} a_j \widetilde{x}_{n-j} \approx \sum_{m=1}^{Q} c_m g_{m,n}, \tag{65}$$

which gives Eq. 21. At step n, the recent sum uses lags $1, \ldots, J$, while the accumulators begin at lag $J+1$; there is no overlap or omitted lag. A ring-buffer implementation transfers a state into the accumulators as it exits the recent window. For $J = 0$, one previous state is sufficient for the transfer, in addition to the accumulators.

The operator that streaming evaluates. Define $\widehat{a}_0 = a_0$, $\widehat{a}_j = a_j$ for $1 \le j \le J$, and $\widehat{a}_j = \sum_m c_m \lambda_m^{j-J-1}$ for $j > J$. Its generating function is rational:

$$\widehat{F}(\zeta) = a_0 + \sum_{j=1}^{J} a_j \zeta^j + \sum_{m=1}^{Q} \frac{c_m \zeta^{J+1}}{1 - \lambda_m \zeta}. \tag{66}$$

The streaming equations are an exact realization, in exact arithmetic, of $[\widehat{F}(\zeta)I - \Delta^\alpha A]\widetilde{X}_d = \Delta^\alpha B U_d$. Replacing F_α by $\widehat{F}$ is the operator change introduced by history compression; its finite-horizon error is analyzed below. Although $|\lambda_m| < 1$ makes each isolated accumulator stable under

bounded forcing, it does not alone rule out zeros of $\det[\widehat{F}(\zeta)I - \Delta^\alpha A]$ on or inside the unit disk. The polynomial and rational terms in Eq. 66 therefore define a finite-dimensional LTI realization for each channel. Writing its zero-input cache update as $z_n = \mathsf{A}_{\mathrm{aug}} z_{n-1}$, the condition $\rho(\mathsf{A}_{\mathrm{aug}}) < 1$ is sufficient for internal stability. It concerns the full coupled update, including the state solve and delay buffer, rather than only the factors λ_m.

Cache update and chunk continuity. Immediately before processing token n, a sufficient cache consists of $\widetilde{x}_{n-1}, \ldots, \widetilde{x}_{n-J-1}$ and the Q vectors $g_{m,n-1}$, with negative-index states initialized to zero. One step consists of the following operations:

1. Read the outgoing state $\widetilde{x}_{n-J-1}$ and update every accumulator using Eq. 64.
2. Accumulate the recent contribution from $\widetilde{x}_{n-1}, \ldots, \widetilde{x}_{n-J}$ and add $\sum_m c_m g_{m,n}$.
3. Solve Eq. 21 for $\widetilde{x}_n$, compute its output, and insert $\widetilde{x}_n$ into the circular buffer.

This ordering uses each outgoing state before its buffer entry is replaced. For $J = 0$, the buffer contains only the previous state and the recent-history sum is empty. Consecutive chunks of one sequence reuse the whole cache; a new independent sequence starts with a zero cache. For fixed parameters, preserving that cache gives the same compressed recurrence as processing the concatenated sequence. Resetting it at a chunk boundary discards represented history. The fit horizon limits the accuracy guarantee, rather than the ability to execute further updates with the same cache.

Conditional finite-horizon trajectory bound. Compare the exact and approximate recurrences under the same input and zero prehistory. Set $e_n = \widetilde{x}_n - x_n$ and

$$r_n = \sum_{j=1}^{n} (\widehat{a}_j - a_j)\widetilde{x}_{n-j}. \tag{67}$$

Subtracting their equations yields

$$S e_n + \sum_{j=1}^{n} a_j e_{n-j} = -r_n. \tag{68}$$

Since S is nonsingular, the formal matrix inverse exists. Let R_ℓ be the coefficients of the state resolvent $[F_\alpha(\zeta)I - \Delta^\alpha A]^{-1} = \sum_{\ell \geq 0} R_\ell \zeta^\ell$. Then $e_n = -\sum_{k=0}^{n} R_{n-k} r_k$. Define

$$G_T = \sum_{\ell=0}^{T-1} \|R_\ell\|, \qquad E_{a,T} = \sum_{j=1}^{T-1} |\widehat{a}_j - a_j|, \qquad \|x\|_{\infty,T} = \max_{0 \leq n < T} \|x_n\|. \tag{69}$$

The convolution inequality gives

$$\|e\|_{\infty,T} \leq G_T E_{a,T} \|\widetilde{x}\|_{\infty,T}. \tag{70}$$

If $\kappa_T = G_T E_{a,T} < 1$, using $\|\widetilde{x}\|_{\infty,T} \leq \|x\|_{\infty,T} + \|e\|_{\infty,T}$ further gives

$$\|e\|_{\infty,T} \leq \frac{\kappa_T}{1 - \kappa_T} \|x\|_{\infty,T}, \qquad \|\widetilde{y} - y\|_{\infty,T} \leq \|C\|\, \|e\|_{\infty,T}. \tag{71}$$

This proves the stated conditional trajectory bound. The condition $\kappa_T < 1$ is sufficient, and need not hold for every fitted model. For any finite T, the constant G_T exists whenever S is nonsingular; it may nevertheless be large. A bound uniform in T would additionally require control of the resolvent gain and accumulated coefficient error as the horizon increases. Let $\mathcal{E}_{\mathrm{LS}}$ denote the value of the weighted objective in Eq. 61. Cauchy–Schwarz gives the finite-horizon relation

$$E_{a,T} \leq \left(\sum_{j=J+1}^{T-1} w_j^{-1} \right)^{1/2} \mathcal{E}_{\mathrm{LS}}^{1/2}. \tag{72}$$

For an empty tail both sides are zero. Together with Eq. 71, this explains why a small least-squares coefficient residual alone is insufficient: error also depends on accumulation across lags and amplification by the state resolvent. Nor does a fit on $j < T$ control larger horizons. In a nonlinear network, subsequent layers and autoregressive input feedback require additional analysis beyond this single-layer, fixed-input bound.

Numerical fidelity measures. The full-history recurrence provides a reference for both approximation routes. For a fixed input and horizon, spectral fidelity can be measured by kernel and output errors relative to that recurrence; streaming fidelity can be measured by coefficient, state, and output errors as J, Q vary. Using the same supplied input separates operator approximation from changes caused by different autoregressive samples. The derivations above do not substitute for those measurements on trained parameters. In particular, for a common supplied input and $L \le T$,

$$\|y^{\mathrm{FFT}} - \widetilde{y}\|_{\infty,L} \le U_{\max} \sum_{j=0}^{L-1} |\widetilde{K}_j - K_j| + \|C\| \, \|\widetilde{x} - x\|_{\infty,L}. \tag{73}$$

Thus the two numerical routes need not agree exactly even though both target the same discrete operator. This bound concerns operator evaluation in exact arithmetic; it excludes floating-point effects.

D.7 Architecture and Computational Costs

Dimensions and scope. Let b be the number of independent sequences processed together, H the hidden width, N the state dimension per channel, and r_A the DPLR rank. The main-text storage statement uses $b = 1$. Below, L is the parallel sequence length, T the SOE fitting horizon, M the contour sample count, and L_c the convolution FFT length. Costs describe one block and distinguish its persistent streaming cache from parameters, preparation, and training activations. All operations are counted at fixed arithmetic precision.

Direct and compressed history. Direct accumulation of the history at step n requires $O(bHNn)$ work and $O(bHNn)$ stored state entries. Over T steps, the total accumulation cost is $O(bHNT^2)$, with $O(bHNT)$ peak state storage. These costs exclude the state solve. In the compressed recurrence, the J recent-history contributions and Q auxiliary vectors require $O(bHN(J+Q))$ accumulation work per token. Including the current state and circular buffer, the persistent cache is

$$\text{streaming cache} = O(bHN(J+Q+1)). \tag{74}$$

The readout additionally costs $O(bHN)$ per token. For fixed J, Q, the cache size and per-token accumulation cost do not depend on how many tokens have already been processed.

Coefficient generation and fitting. Let $H_\alpha \le H$ denote the number of distinct orders, with a common choice of discretization symbol. Equation 41 generates T coefficients in $O(H_\alpha T)$ work. For $m_{\mathrm{fit}} = T - J - 1 \ge Q$, a dense QR factorization of the weighted fitting matrix costs $O(m_{\mathrm{fit}} Q^2)$ work and $O(m_{\mathrm{fit}} Q)$ storage. Applying a prepared factorization to one new right-hand side costs $O(m_{\mathrm{fit}} Q + Q^2)$. If all orders use the same factors and weights, the factorization is shared across those fits; otherwise each fitting configuration requires its own factorization. A rank-deficient or underdetermined fit can use an SVD, whose cost is governed by the smaller matrix dimension. The dense design matrix and the T reference coefficients are preparation storage and need not remain in the token-processing cache. After training, the fitted coefficients and state-solve factors are reused while the learned operator remains fixed.

Preparing and applying the state solve. For diagonal A, preparing S^{-1} requires $O(HN)$ work and storage, and each solve costs $O(bHN)$. For tied DPLR, set $d = \Delta^\alpha$ and $S_0 = a_0 I - d\,\mathrm{diag}(\Lambda)$. When the required inverses exist, Woodbury gives

$$S^{-1} = S_0^{-1} - dS_0^{-1} P\big(I + dP^* S_0^{-1} P\big)^{-1} P^* S_0^{-1}. \tag{75}$$

The sufficient conditions in Appendix D.4 ensure these inverses exist for Tustin. Preparing the diagonal factors, the products with P, and a factorization of the $r_A \times r_A$ correction matrix costs

$$\begin{aligned} \text{preparation work} &= O\big(H(Nr_A^2 + r_A^3 + N)\big), \\ \text{stored solve factors} &= O\big(H(Nr_A + r_A^2 + N)\big). \end{aligned} \tag{76}$$

Once prepared, each solve costs $O\big(bH(Nr_A + r_A^2 + N)\big)$. These factors are shared across sequences and are separate from the history cache. A dense implementation instead requires an $O(HN^3)$ factorization and $O(bHN^2)$ solve work per token. Thus the history-accumulation bound alone does not specify the full cost of an implementation that uses dense state solves.

Parallel kernel construction and application. For diagonal dynamics, evaluating the transfer at M contour points costs $O(HMN)$ work. With a generic rank-r_A DPLR implementation, forming and solving each small Woodbury correction gives $O\big(HM(Nr_A^2 + r_A^3 + N)\big)$ work. These are direct structured-evaluation bounds; faster specialized summation schemes are not needed for the stated scaling with L. Kernel recovery costs $O(HM \log M)$. Once the kernel is constructed, applying the padded convolution to a batch costs $O(bHL_c \log L_c)$. Kernel construction is shared across the minibatch because the channel operators do not depend on the input. With $M = \Theta(L)$ and $L_c = \Theta(L)$, the total work is therefore $O(L \log L)$ at fixed b, H, N, r_A. Storing the samples, recovered kernels, and sequence transforms requires sequence-length-dependent arrays; the constant-cache statement applies to streaming inference.

Residual blocks and causal processing. In Eq. 22, each channel applies a scalar-input, scalar-output temporal operator with an N-dimensional state. The operator's direct term Du_n is distinct from the outer residual connection. A dense H-to-H projection costs $O(bLH^2)$ for a sequence batch, or $O(bH^2)$ per streaming step. Position-wise normalization and elementwise activations cost $O(bLH)$ across the sequence. For causal streaming, normalization at a token may mix its feature channels but must not use statistics from future tokens. The projection and activation are also applied at that token. Each block maintains its own temporal cache. The resulting network is nonlinear even though each temporal operator is LTI.