Learning to Dissipate Energy in Oscillatory State-Space Models

Jared Boyer
MIT CSAIL
jaredb@mit.edu

T. Konstantin Rusch
MIT CSAIL

Daniela Rus
MIT CSAIL

Abstract

State-space models (SSMs) are a class of networks for sequence learning that benefit from fixed state size and linear complexity with respect to sequence length, contrasting the quadratic scaling of typical attention mechanisms. Inspired from observations in neuroscience, Linear Oscillatory State-Space models (LinOSS) are a recently proposed class of SSMs constructed from layers of discretized forced harmonic oscillators. Although these models perform competitively, leveraging fast parallel scans over diagonal recurrent matrices and achieving state-of-the-art performance on tasks with sequence length up to 50k, LinOSS models rely on rigid energy dissipation ("forgetting") mechanisms that are inherently coupled to the timescale of state evolution. As forgetting is a crucial mechanism for long-range reasoning, we demonstrate the representational limitations of these models and introduce Damped Linear Oscillatory State-Space models (D-LinOSS), a more general class of oscillatory SSMs that learn to dissipate latent state energy on multiple timescales. We analyze the spectral distribution of the model's recurrent matrices and prove that the SSM layers exhibit stable dynamics under simple, flexible parameterizations. D-LinOSS consistently outperforms previous LinOSS methods on long-range learning tasks, without introducing additional complexity, and simultaneously reduces the hyperparameter search space by 50%.

1 Introduction

State-space models (SSMs) Gu et al. [2021], Smith et al. [2023], Gu and Dao [2023], Hasani et al. [2022], Rusch and Rus [2025] have emerged as a powerful deep learning architecture for sequence modeling, demonstrating strong performances across various domains, including natural language processing Gu and Dao [2023], audio generation Goel et al. [2022], reinforcement learning Lu et al. [2024], and scientific and engineering applications Hu et al. [2024].

Despite the abundance of neural network architectures for sequence modeling, SSMs have gained significant attention due to their fundamental advantages over both Recurrent Neural Networks (RNNs) and Transformer architectures based on self-attention mechanisms Vaswani [2017]. Built upon layers of sequence-to-sequence transformations defined by linear dynamical systems, SSMs integrate principles from control theory with modern deep learning techniques, making them highly effective across multiple modalities. While recent SSM architectures are often formulated as linear RNNs Orvieto et al. [2023], they introduce notable improvements over their predecessors, offering enhanced speed, accuracy, and the ability to capture long-range dependencies more effectively.

In this work, we focus on and further extend the recently introduced linear oscillatory state-space model (LinOSS) Rusch and Rus [2025]. LinOSS is based on a system of second-order ordinary differential equations (ODEs) that model forced harmonic oscillators and is discretized using fast associative parallel scans. The structure of the underlying oscillatory dynamics allows LinOSS to learn long-range interactions over arbitrary time scales without enforcing any constraints on the SSM

LinOSS single scale damping   **Damped LinOSS** multi-scale damping

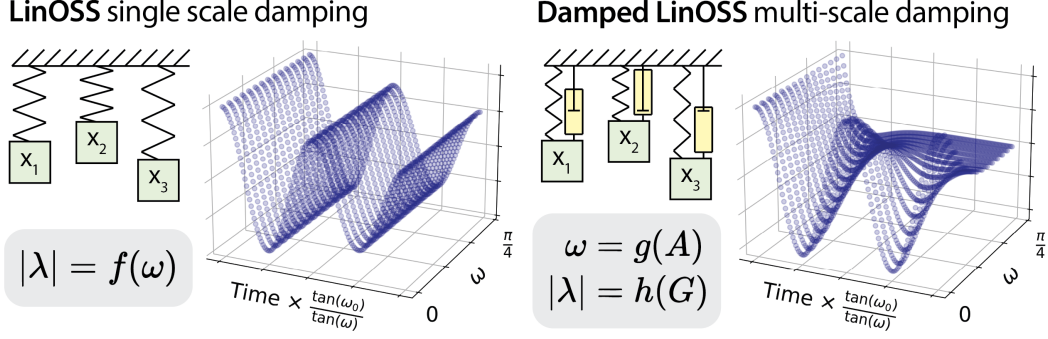$$|\lambda| = f(\omega)$$

$$\omega = g(A)$$
$$|\lambda| = h(G)$$

Figure 1: Previous LinOSS models directly couple the frequency and magnitude of discretized system eigenvalues, reducing latent state energy dissipation to a single scale when normalizing time by frequency. Through a simple, complexity-free parametric extension, D-LinOSS learns system damping on all scales, independent of frequency, expanding the range of expressible internal dynamics. The specific damping behavior depicted in the right diagram is selected arbitrarily.

state matrix. However, as we will subsequently show, LinOSS inherently couples frequency and damping, reducing latent state energy dissipation to a single scale and limiting the model's expressive power. To overcome this, we propose *Damped Linear Oscillatory State-Space Models (D-LinOSS)*, which enhance the LinOSS architecture by incorporating learnable damping across all scales.

Our approach constructs a deep state space model capable of capturing a wide range of temporal relationships by expanding the expressivity of individual SSM layers. Unlike previous versions of LinOSS that were constrained to a limited subset of oscillatory systems, our method allows each layer to independently learn a wider range of stable oscillatory dynamics, collectively leading to a more powerful sequence model. Our full contributions are:

- We conduct a rigorous spectral analysis of the proposed D-LinOSS model, highlighting the representational improvements enabled by learnable damping.

- We validate the theoretical expressivity improvements through a synthetic experiment of learning exponential decay.

- We derive a stable parameterization of D-LinOSS and introduce an initialization procedure to generate arbitrary eigenvalue distributions in the recurrent matrix. We perform ablations comparing different initialization techniques.

- We provide extensive empirical evaluation, showing that D-LinOSS on average outperforms state-of-the-art models across eight different challenging real-world sequential datasets. At the same time, D-LinOSS reduces the hyperparameter space of previous LinOSS models by 50% by eliminating the need for multiple discretization schemes.

- To support reproducibility and further research, we release our code and experiments at `github.com/jaredbmit/damped-linoss`.

2   Background

2.1   Underlying continuous-time dynamics

D-LinOSS layers are constructed from a system of damped, forced harmonic oscillators, represented in the following state-space formulation.

$$\mathbf{x}''(t) = -\mathbf{A}\mathbf{x}(t) - \mathbf{G}\mathbf{x}'(t) + \mathbf{B}\mathbf{u}(t),$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{1}$$

The system input $\mathbf{u} : [0, T] \rightarrow \mathbb{R}^p$, the system state $\mathbf{x} : [0, T] \rightarrow \mathbb{R}^m$, and the system output $\mathbf{y} : [0, T] \rightarrow \mathbb{R}^q$ are all vector-valued functions of continuous time t. The parameters $\mathbf{A}$ and $\mathbf{G}$ are

restricted to diagonal matrices with non-negative entries, meaning (1) is an uncoupled second-order system. The feed-forward operation $\mathbf{D}\mathbf{u}(t)$ will be omitted for the rest of the paper for concision.

The goal of a D-LinOSS layer is to model complex sequence-to-sequence transformations $\mathbf{u} \mapsto \mathbf{y}$ by learning parameters $\mathbf{A}, \mathbf{G}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$. $\mathbf{A}$ controls the natural frequency of the system's oscillation and $\mathbf{G}$ defines the damping, i.e., the energy dissipation of the latent state. The underlying dynamical system of previous LinOSS models is (1) subject to $\mathbf{G} = \mathbf{0}$; thus, D-LinOSS is constructed from a more general oscillatory state space layer with learnable damping. The additional m learnable parameters per layer are a negligible contribution to model size and have no impact on speed.

2.2 Discretization

To approximately solve the ODE system in (1), we first rewrite it as an equivalent first-order system in (2) by introducing the auxiliary state variable $\mathbf{z}(t) \in \mathbb{R}^m$. The full state $[\mathbf{z}, \mathbf{x}]^\intercal$ is denoted $\mathbf{w} \in \mathbb{R}^{2m}$. A discretization scheme is then applied to the first-order system (2), mapping the parameters $\mathbf{A}, \mathbf{G}, \mathbf{B}, \mathbf{C}$ to discrete-time counterparts $\mathbf{M}, \mathbf{F}, \mathbf{H}$ in the system (3). Importantly, this mapping introduces learnable timestep parameters $\Delta t \in \mathbb{R}^m$ that govern the integration interval for the ODE.

$$
\begin{aligned}
\mathbf{z}'(t) &= -\mathbf{A}\mathbf{x}(t) - \mathbf{G}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t), \\
\mathbf{x}'(t) &= \mathbf{z}(t), \\
\mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)
\end{aligned}
\qquad (2) \qquad \xrightarrow{\text{Discretize}} \qquad
\begin{aligned}
\mathbf{w}_{k+1} &= \mathbf{M}\mathbf{w}_k + \mathbf{F}\mathbf{u}_{k+1}, \\
\mathbf{y}_{k+1} &= \mathbf{H}\mathbf{w}_k
\end{aligned}
\qquad (3)
$$

Unlike standard first-order SSMs, LinOSS explicitly models the acceleration and velocity of the system state, resulting in smoother outputs due to the twice-integrated structure. However, this structure necessitates the use of special discretization schemes to maintain system stability.

Most SSMs discretize the underlying continuous-time dynamics using zero-order hold or the bilinear method. However, due to the second-order nature of LinOSS, both implicit-explicit (IMEX) and fully implicit (IM) methods are leveraged to ensure stability. These integrators endow different "forgetting" behaviors into the discrete-time systems; the IM integrator introduces an energy dissipation term whereas the IMEX integrator completely preserves energy over time. The selection of discretization technique is a binary hyperparameter in the original LinOSS model and gives rise to two flavors of LinOSS (LinOSS-IM and LinOSS-IMEX) that exhibit different dynamical behaviors.

We extend the use of the structure-preserving implicit-explicit method to the D-LinOSS layer, as using the implicit method would introduce additional dissipative terms that are both unnecessary and uncontrollable for the learning process. Applying the IMEX discretization to System (2) yields

$$
\begin{aligned}
\mathbf{z}_{k+1} &= \mathbf{z}_k + \Delta t\big(-\mathbf{A}\mathbf{x}_k - \mathbf{G}\mathbf{z}_{k+1} + \mathbf{B}\mathbf{u}_{k+1} \big), \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta t\mathbf{z}_{k+1},
\end{aligned}
\qquad (4)
$$

or in matrix form,

$$
\begin{bmatrix} \mathbf{I} + \Delta t\mathbf{G} & \mathbf{0} \\ -\Delta t\mathbf{I} & \mathbf{I} \end{bmatrix}
\begin{bmatrix} \mathbf{z}_{k+1} \\ \mathbf{x}_{k+1} \end{bmatrix}
=
\begin{bmatrix} \mathbf{I} & -\Delta t\mathbf{A} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}
\begin{bmatrix} \mathbf{z}_k \\ \mathbf{x}_k \end{bmatrix}
+
\begin{bmatrix} \Delta t\mathbf{B}\mathbf{u}_k \\ \mathbf{0} \end{bmatrix}.
$$

Inverting the left hand side block matrix and re-arranging terms, we arrive at the final discrete-time formulation in (3), parameterized by $\mathbf{M} \in \mathbb{R}^{2m \times 2m}$, $\mathbf{F} \in \mathbb{R}^{2m \times p}$, and $\mathbf{H} \in \mathbb{R}^{q \times 2m}$.

$$
\mathbf{M} := \begin{bmatrix} \mathbf{S}^{-1} & -\Delta t\mathbf{S}^{-1}\mathbf{A} \\ \Delta t\mathbf{S}^{-1} & \mathbf{I} - \Delta t^2\mathbf{S}^{-1}\mathbf{A} \end{bmatrix}, \quad
\mathbf{F} := \begin{bmatrix} \Delta t\mathbf{S}^{-1}\mathbf{B} \\ \Delta t^2\mathbf{S}^{-1}\mathbf{B} \end{bmatrix}, \quad
\mathbf{H} := \begin{bmatrix} \mathbf{0} & \mathbf{C} \end{bmatrix}
\qquad (5)
$$

Here, the Schur complement is the diagonal matrix $\mathbf{S} = \mathbf{I} + \Delta t\mathbf{G}$ and $\mathbf{M}$ and $\mathbf{F}$ are block matrices composed of diagonal sub-matrices.

2.3 Diagonal equivalence

For general SSMs, evaluating the recurrence $\mathbf{w}_{k+1} = \mathbf{M}\mathbf{w}_k + \mathbf{F}\mathbf{u}_{k+1}$ with a dense transition matrix $\mathbf{M}$ is computationally expensive, which can be prohibitive for both training and inference on long sequences. However, when $\mathbf{M}$ is diagonalizable, the system can be rewritten equivalently

$$\tilde{\mathbf{w}}_{k+1} = \mathbf{\Lambda}\tilde{\mathbf{w}}_k + \tilde{\mathbf{F}}\mathbf{u}_{k+1}, \quad \mathbf{y}_{k+1} = \tilde{\mathbf{H}}\tilde{\mathbf{w}}_k, \tag{6}$$

where $\mathbf{\Lambda}$ is diagonal and the change of variables follows

$$\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}, \quad \tilde{\mathbf{w}} = \mathbf{V}^{-1}\mathbf{w}, \quad \tilde{\mathbf{F}} = \mathbf{V}^{-1}\mathbf{F}, \quad \tilde{\mathbf{H}} = \mathbf{H}\mathbf{V}. \tag{7}$$

In this formulation, the recurrence $\mathbf{\Lambda}\tilde{\mathbf{w}}_k$ becomes a vector dot-product, reducing the computational cost to $O(m)$ per step. In practice, many SSMs are learned directly in this diagonalized space, avoiding the cost of explicitly computing $\mathbf{V}$ or $\mathbf{V}^{-1}$.

In the case of D-LinOSS, the recurrent matrix $\mathbf{M}$ is composed of block matrices which are diagonal (see (5)), so computing $\mathbf{M}\mathbf{w}_k$ already requires only $O(m)$ operations and diagonalization is not strictly necessary for efficient implementation.

2.4 Associative parallel scans

Many modern SSM architectures rely on associative parallel scan algorithms Kogge and Stone [1973], Blelloch [1990] to efficiently compute recurrent operations across long sequences. The key idea is to exploit the associativity of the recurrence operator to parallelize what would otherwise be a sequential computation.

Given a binary associative operator $\bullet$, satisfying $(a \bullet b) \bullet c = a \bullet (b \bullet c)$, the cumulative product over a sequence $[a, b, c, \dots]$,

$$[a, a \bullet b, a \bullet b \bullet c, \dots],$$

can be computed in $O(\log N)$ sequential steps instead of $O(N)$, where N is the sequence length. This transformation is commonly referred to as an *operator scan*.

For SSMs, associative scans enable efficient computation of the recurrence $\mathbf{w}_{k+1} = \mathbf{M}\mathbf{w}_k + \mathbf{F}\mathbf{u}_k$ when $\mathbf{M}$ is time-invariant, acting as a key building block for scaling SSMs to long contexts.

3 Theoretical properties

Spectral analysis provides a powerful lens to examine the stability and dynamical behavior of SSMs. In the absence of bounding nonlinearities like $\tanh$, the eigenvalues of the recurrent matrix $\mathbf{M}$ fully govern how latent states evolve across time. In particular, eigenvalues with near unit norm retain energy across long time horizons, while those closer to zero rapidly dissipate energy.

In the previous LinOSS-IM and LinOSS-IMEX models, which are based on a system of harmonic oscillators, the internal system spectra are rigidly defined by the selection of discretization technique, tightly coupling frequency and damping. As shown in Figure 1, this reduces latent state energy dissipation to a single scale when normalizing frequency, limiting the range of expressible dynamics.

For D-LinOSS, the spectrum of $\mathbf{M}$ instead arises from the structure of damped harmonic oscillators, introducing a new tunable mechanism that decouples damping from frequency. Unlike the preceding models, D-LinOSS layers can represent all stable second-order systems, yielding a broader range of expressible dynamics and thus a more powerful sequence model. This is depicted in Figure 1, where the scale of energy dissipation can be arbitrarily selected regardless of oscillation frequency.

These notions are formalized in this section, where we characterize the eigenvalues of D-LinOSS, derive stability conditions, and compare the resulting spectral range to that of previous LinOSS models. In particular, we rigorously show that the set of reachable, stable eigenvalue configurations in D-LinOSS is the full complex unit disk, where that of LinOSS has zero measure in $\mathbb{C}$.

3.1 Spectral analysis and stability

Proposition 3.1. *The eigenvalues of the D-LinOSS recurrent matrix $\mathbf{M} \in \mathbb{R}^{2m \times 2m}$ are*

$$\lambda_{i_{1,2}} = \frac{1 + \frac{\Delta t_i}{2}\mathbf{G}_i - \frac{\Delta t_i^2}{2}\mathbf{A}_i}{1 + \Delta t_i \mathbf{G}_i} \pm \frac{\frac{\Delta t_i}{2}\sqrt{(\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2 - 4\mathbf{A}_i}}{1 + \Delta t_i \mathbf{G}_i}, \tag{8}$$

where pairs of eigenvalues are denoted as $\lambda_{i_{1,2}}$ and $i = 1, 2, ..., m$.

Proof. This straightforward derivation is provided in Appendix A.1. The computation is simplified because each of the m second-order systems is decoupled. It thus suffices to show all following spectral properties and stability criteria for a given $i \in \{1, \ldots, m\}$. □

Proposition 3.1 shows that eigenvalues of D-LinOSS are tuned through choices of $\mathbf{A}$, $\mathbf{G}$, and Δt. We now detail a sufficient condition for system stability.

Proposition 3.2. *Assume $\mathbf{G}_i$, $\mathbf{A}_i$ are non-negative, and $\Delta t_i \in (0, 1]$. If the following is satisfied:*

$$(\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2 \leq 4\mathbf{A}_i, \tag{9}$$

then $\lambda_{i_{1,2}}$ come in complex conjugate pairs λ_i, λ_i^* with the following magnitude:

$$|\lambda_i| = \frac{1}{\sqrt{1 + \Delta t_i \mathbf{G}_i}} \leq 1,$$

i.e., the eigenvalues are unit-bounded. Define $\mathcal{S}_i$ to be the set of all $(\mathbf{G}_i, \mathbf{A}_i)$ that satisfy the above condition. For notational convenience, we order the eigenvalues such that $\mathrm{Im}(\lambda_i) \geq 0$, $\mathrm{Im}(\lambda_i^*) \leq 0$.

Proof. The proof is detailed in Appendix A.2. Condition (9) is simply the non-positivity of the determinant in the eigenvalue expression of Proposition 3.1, which we show is sufficient for the unit-boundedness of $|\lambda_i|$. □

We now demonstrate that the spectral image of $\mathcal{S}_i$ is the full unit disk, meaning D-LinOSS is capable of representing every stable, damped, decoupled second-order system.

Proposition 3.3. *The mapping $\Phi : \mathcal{S}_i \to \mathbb{C}_{|z| \leq 1}$ defined by $(\mathbf{G}_i, \mathbf{A}_i) \mapsto \lambda_i$ is bijective.*

Proof. Provided in Appendix A.3, we construct a well-defined inverse mapping $\Phi^{-1} : \lambda_i \mapsto (\mathbf{G}_i, \mathbf{A}_i)$, which is equivalent to showing Φ is a one-to-one mapping from $\mathcal{S}_i$ to $\mathbb{C}$. Discussed later, this inverse map has practical utility in matrix initialization, enabling the initialization of arbitrary distributions of stable eigenvalues. □

Compared to D-LinOSS, the preceding LinOSS-IM and LinOSS-IMEX are limited in the set of reachable eigenvalues. We first recall the eigenvalues of these two models (Rusch and Rus [2025]), denoted by λ^{IMEX} and λ^{IM} under each scheme. These are given by

$$\lambda_{i_{1,2}}^{\mathrm{IMEX}} = \frac{1}{1 + \Delta t_i^2 \mathbf{A}_i} \pm j\frac{\Delta t_i\sqrt{\mathbf{A}_i}}{1 + \Delta t_i^2 \mathbf{A}_i}, \quad \lambda_{i_{1,2}}^{\mathrm{IM}} = \frac{1}{2}(2 - \Delta t_i^2 \mathbf{A}_i) \pm \frac{j}{2}\sqrt{\Delta t_i^2 \mathbf{A}_i(4 - \Delta t_i^2 \mathbf{A}_i)}. \tag{10}$$

Both forms impose a rigid relationship between oscillation frequency and damping, constraining the set of reachable spectra. The following proposition formalizes this idea by showing that the set of stable eigenvalues reachable under these parameterizations occupies zero area within the unit disk.

Proposition 3.4. *For both LinOSS-IM and LinOSS-IMEX, the set of eigenvalues constructed from $\mathbf{A}_i \in \mathbb{R}_{\geq 0}$ and $\Delta t_i \in (0, 1]$ is of measure zero in $\mathbb{C}$.*

Proof. See Appendix A.4. Using the change of variables $\gamma_i = \Delta t_i\sqrt{\mathbf{A}_i}$, we can express the reachable eigenvalues for both LinOSS models as one-dimensional curves, which have zero measure in $\mathbb{C}$. □

The incorporation of explicitly learnable damping enables D-LinOSS to model a wider range of stable dynamical systems and thus increases the representational capacity of the overall deep sequence model. The empirical performance benefits are discussed in the following sections.

3.2 Motivation

It is reasonable to ask whether a larger set of reachable eigenvalues is actually useful. Notably, LinOSS is provably *universal* Rusch and Rus [2025], Lanthaler et al. [2024], meaning it can approximate any causal and continuous operator between input and output signals to arbitrary accuracy (see Appendix A.5 for a formal definition). This property trivially extends to D-LinOSS, as setting $\mathbf{G} = \mathbf{0}$ recovers the original LinOSS model. However, while universality characterizes theoretical capacity, it is not necessarily indicative of how well a model can learn in practice. To motivate the empirical benefits of a broader spectral range, we construct a synthetic regression experiment.

We simulate a dynamical system with a single discrete eigenvalue $\lambda = 0.8$, corresponding to an exponentially decaying response. No input or output transformation is applied. Random sequences of scalar inputs are passed through this system, and models are trained to predict the resulting output. We compare D-LinOSS to LinOSS-IM and LinOSS-IMEX in terms of test root mean squared error (RMSE). A small hyperparameter grid was searched independently for each model, and three random seeds were trained per setting. The reported result is the best average test RMSE across seeds. (Appendix B.2 provides more detail on experiment setup.)

Table 1: Learning exponential decay.

Model	RMSE $\times 10^{-4}$	
LinOSS-IMEX	24.5 ± 2.3	*30.6 ×*
LinOSS-IM	8.0 ± 1.7	*10.0 ×*
D-LinOSS	0.8 ± 0.1	*1.0 ×*

D-LinOSS achieves test RMSE approximately $10\times$ lower than LinOSS-IM and $30.6\times$ lower than LinOSS-IMEX. Reflecting on the expressions in 10, the target eigenvalue $\lambda = 0.8$ lies outside the reachable spectra of both baseline models. This gap in performance suggests that although the previous LinOSS models are universal, a limited spectral range can impair the models' ability to represent certain temporal relationships. By directly learning system damping $\mathbf{G}$, D-LinOSS moves beyond rigidly defined energy dissipation behavior and is capable of accurately capturing a wider range of dynamics.

3.3 Architecture details

Recall from Proposition 3.2 that $(\mathbf{G}, \mathbf{A}) \in \mathcal{S}$ is sufficient for system stability. To guarantee this condition is satisfied also during training, the system matrices $\mathbf{A}$ and $\mathbf{G}$ are parameterized as follows,

$$\mathbf{G} = \text{Relu}(\tilde{\mathbf{G}}), \quad \mathbf{A} = \text{Clamp}(\tilde{\mathbf{A}}, L(\mathbf{G}), U(\mathbf{G})),$$

where $\tilde{\mathbf{G}}, \tilde{\mathbf{A}}$ are the unconstrained, learned parameters. The function

$$\text{Clamp}(x, a, b) := x + \text{Relu}(x - a) - \text{Relu}(x - b)$$

is a bounding operation to ensure the oscillatory dynamics remain in the stable range. Here, $L(\mathbf{G}), U(\mathbf{G})$ are the lower and upper interval boundaries of $\mathcal{S}$ for a constant value of $\mathbf{G}$, which is found by solving a simple quadratic expression. More details are provided in Appendix B.1.

3.4 Initialization

Performance of SSMs are heavily impacted by the initialized distribution of eigenvalues (Orvieto et al. [2023]). Many approaches, e.g., Gu et al. [2021], Smith et al. [2023], leverage structured initialization schemes—such as the HiPPO framework (Gu et al. [2020]) to enhance long-range learning capability. However, studies indicate that simpler initialization techniques, such as uniform sampling over the complex unit disk, are capable of recovering equal or similar performance (Orvieto et al. [2023]). Building on insights from Orvieto et al. [2023], we develop a procedure to initialize the recurrent matrix $\mathbf{M}$ with eigenvalues uniformly sampled in the stable complex region.

Proposition 3.3 states that the relationship between model parameters $\mathbf{G}$, $\mathbf{A}$ and the eigenvalues within the complex unit disk is bijective, allowing us to control the spectral distribution of $\mathbf{M}$. Using

Table 2: Test accuracies averaged over five different seeds on UEA time-series classification datasets. The highest score is indicated in **bold** and the second highest is underlined. The dataset names are abbreviations of the following UEA datasets: EigenWorms (Worms), SelfRegulationSCP1 (SCP1), SelfRegulationSCP2 (SCP2), EthanolConcentration (Ethanol), Heartbeat, and MotorImagery (Motor).

	Worms	**SCP1**	**SCP2**	**Ethanol**	**Heartbeat**	**Motor**	**Avg**
Seq. length	17,984	896	1,152	1,751	405	3,000	
# of Classes	5	2	2	4	2	2	
NRDE	83.9 ± 7.3	80.9 ± 2.5	53.7 ± 6.9	25.3 ± 1.8	72.9 ± 4.8	47.0 ± 5.7	60.6
NCDE	75.0 ± 3.9	79.8 ± 5.6	53.0 ± 2.8	$\underline{29.9 \pm 6.5}$	73.9 ± 2.6	49.5 ± 2.8	60.2
Log-NCDE	85.6 ± 5.1	83.1 ± 2.8	53.7 ± 4.1	$\mathbf{34.4 \pm 6.4}$	75.2 ± 4.6	53.7 ± 5.3	64.3
LRU	87.8 ± 2.8	82.6 ± 3.4	51.2 ± 3.6	21.5 ± 2.1	$\mathbf{78.4 \pm 6.7}$	48.4 ± 5.0	61.7
S5	81.1 ± 3.7	$\mathbf{89.9 \pm 4.6}$	50.5 ± 2.6	24.1 ± 4.3	$\underline{77.7 \pm 5.5}$	47.7 ± 5.5	61.8
S6	85.0 ± 16.1	82.8 ± 2.7	49.9 ± 9.4	26.4 ± 6.4	76.5 ± 8.3	51.3 ± 4.7	62.0
Mamba	70.9 ± 15.8	80.7 ± 1.4	48.2 ± 3.9	27.9 ± 4.5	76.2 ± 3.8	47.7 ± 4.5	58.6
LinOSS-IMEX	80.0 ± 2.7	87.5 ± 4.0	$\mathbf{58.9 \pm 8.1}$	29.9 ± 1.0	75.5 ± 4.3	57.9 ± 5.3	65.0
LinOSS-IM	$\mathbf{95.0 \pm 4.4}$	87.8 ± 2.6	58.2 ± 6.9	$\underline{29.9 \pm 0.6}$	75.8 ± 3.7	$\underline{60.0 \pm 7.5}$	$\underline{67.8}$
D-LinOSS	$\underline{93.9 \pm 3.2}$	$\underline{88.9 \pm 3.0}$	$\underline{58.6 \pm 2.3}$	$\underline{29.9 \pm 0.6}$	75.8 ± 4.9	$\mathbf{61.1 \pm 2.0}$	$\mathbf{68.0}$

this result, we can initialize $\mathbf{G}$, $\mathbf{A}$ to generate any desired eigenvalue distribution via Φ^{-1}. The procedure for initializing D-LinOSS matrices in the following experiments is: 1. define a radial subinterval $[r_{\min}, r_{\max}]$ of the complex unit disk $\mathbb{C}_{|z| \leq 1}$, 2. uniformly sample conjugate eigenvalue pairs from that subregion, 3. compute $(\mathbf{G}_i, \mathbf{A}_i) = \Phi^{-1}(\lambda_i) \, \forall i = 1, \ldots, m$.

Since eigenvalues with magnitudes near 1 are shown to be useful for modeling long range dependencies (LRDs) Orvieto et al. [2023], Rusch and Rus [2025], we investigate biased sampling toward the outer complex disk. Through an initialization study detailed in Appendix B.5, we determine that sampling eigenvalues within the radial band $0.9 \leq |\lambda_i| \leq 1$ yields strong performance on learning LRDs. This initialization technique is used for all subsequent D-LinOSS results.

4 Results

We now evaluate the empirical performance of D-LinOSS on a suite of learning tasks that span disciplines across biology, medicine, chemistry, and photonics. As the linear complexity and fixed state size of SSMs emphasize their utility for learning long-range dependencies, we evaluate candidate models on datasets with temporal relationships spanning thousands of measurements. We compare model performance with nine other state-of-the-art sequence modeling approaches, including the two precursor models LinOSS-IM and LinOSS-IMEX. Experimental design and hyperparameter spreads are kept consistent across all models to ensure fair comparison.

4.1 UEA time-series classification

We consider a benchmark based on sequential data from the University of East Anglia (UEA) Multivariate Time Series Classification Archive (UEA-MTSCA) Bagnall et al. [2018]. This benchmark, introduced in Walker et al. [2024], consists of a subset of six datasets chosen for their number of trials and sequence lengths in order to evaluate the ability of sequence models to capture long-range interactions. The UEA datasets are all classification tasks, ranging from classifying organisms from motion readings (*EigenWorms*) to classifying fluid alcohol percentage based on measurements of transmissive light spectra (*EthanolConcentration*). We follow the experimental design proposed in Walker et al. [2024], conducting a search over a grid of 162 predetermined hyperparameter configurations for each dataset. Further, each model instance is trained on five seeds, and the average test accuracy for the top performing model instances are reported. The high scoring hyperparameter configurations of D-LinOSS model instances are tabulated in Appendix B.3. All models use the same 70-15-15 train-validation-test data splits, controlled by the seed for a given trial. Model scores for LinOSS-IM and LinOSS-IMEX are sourced from Rusch and Rus [2025] and all other model scores are sourced from Walker et al. [2024]. D-LinOSS models were trained using NVIDIA V100 GPUs.

Out of all models tested, D-LinOSS achieves the highest average test accuracy across the six UEA datasets–raising the previous high score from 67.8% to 68.0%. Notably, D-LinOSS improves state-of-

Table 3: Test accuracies averaged over five different seeds on the PPG-DaLiA time-series regression dataset. The best score is indicated in **bold** and the second best is underlined.

Model	MSE $\times 10^{-2}$
NRDE Morrill et al. [2021]	9.90 ± 0.97
NCDE [Kidger et al., 2020]	13.54 ± 0.69
Log-NCDE Walker et al. [2024]	9.56 ± 0.59
LRU Orvieto et al. [2023]	12.17 ± 0.49
S5 Smith et al. [2023]	12.63 ± 1.25
S6 Gu and Dao [2023]	12.88 ± 2.05
Mamba Gu and Dao [2023]	10.65 ± 2.20
LinOSS-IMEX Rusch and Rus [2025]	7.5 ± 0.46
LinOSS-IM Rusch and Rus [2025]	$\underline{6.4 \pm 0.23}$
D-LinOSS	$\mathbf{6.16 \pm 0.73}$

the-art accuracy on MotorImagery by 1.1% and scores in the top two for five out of the six datasets. D-LinOSS also outperforms the combination of both preceding models: the average score-wise maximum between LinOSS-IM and LinOSS-IMEX is 67.9%, still shy of D-LinOSS. D-LinOSS improves on or matches the second best model, LinOSS-IM, in all but one dataset, EigenWorms, which is the smallest dataset out of the six.

4.2 PPG-DaLiA time-series regression

We next evaluate model performance on a time series regression task using the *PPG dataset for motion compensation and heart rate estimation in Daily Life Activities* (PPG-DaLiA) (Reiss et al. [2019]). Here, models are tasked with learning human heart rate patterns as a function of various sensor measurements, such as ECG readings, wearable accelerometers, and respiration sensing. The dataset consists of 15 different subjects performing a variety of daily tasks, and bio-sensory data is collected in sequences up to 50,000 points in length. We follow the same experimental design as before, searching model hyperparameters over a grid of 162 configurations and training each model instance on five seeds. All models use the same 70-15-15 data split. D-LinOSS achieves the best results, reducing the lowest MSE from 6.4 to 6.16 ($\times 10^{-2}$) compared to LinOSS-IM.

4.3 Weather time-series forecasting

To assess the generality of D-LinOSS as a sequence-to-sequence model, as in Gu et al. [2021], we evaluate its performance on a long-horizon time-series forecasting task without any architectural modifications. In this setting, forecasting is framed as a masked sequence transformation problem, allowing the model to predict future values based solely on partially masked input sequences.

We focus on the difficult task of weather prediction introduced in Zhou et al. [2021], which involves predicting one month of hourly measured multivariate local climatological data based on the previous month's measurements. The dataset spans 1,600 U.S. locations between 2010 to 2013 and is further detailed in NCEI [2025].

In this benchmark, D-LinOSS is compared against Transformer-based architectures, LSTM variants, the structured state-space model S4, and previous versions of LinOSS. D-LinOSS achieves the best performance, reducing the lowest mean absolute error (MAE) from 0.508 (LinOSS-IMEX) to 0.486.

5 Related Work

State Space Models (SSMs) were originally introduced as a powerful deep learning framework for sequential data in Gu et al. [2021]. Early models Gu et al. [2022], Nguyen et al. [2022], Goel et al. [2022] primarily leveraged Fast Fourier Transform (FFT) and HiPPO Gu et al. [2020] parameterizations to efficiently solve linear recurrences. Over time, SSMs have undergone continuous refinement. More recently, most SSM architectures utilize diagonal state matrices combined with

Table 4: Mean absolute error on the weather dataset predicting the future 720 time steps based on the past 720 time steps. The best score is indicated in **bold** and the second best is underlined.

Model	Mean Absolute Error
Informer [Zhou et al., 2021]	0.731
Informer[†] [Zhou et al., 2021]	0.741
LogTrans [Li et al., 2019]	0.773
Reformer [Kitaev et al., 2020]	1.575
LSTMa [Bahdanau et al., 2016]	1.109
LSTnet [Lai et al., 2018]	0.757
S4 [Gu et al., 2021]	0.578
LinOSS-IMEX Rusch and Rus [2025]	0.508
LinOSS-IM Rusch and Rus [2025]	0.528
D-LinOSS	**0.486**

fast associative parallel scans, which has been used in the context of RNNs before Martin and Cundy [2017], Kaul [2020]. This approach was first introduced to SSMs in Smith et al. [2023], which still relied on HiPPO matrices for initializing SSM weights, but has since been simplified to random weight initialization, as demonstrated in Orvieto et al. [2023]. In addition, while our proposed D-LinOSS model and all aforementioned models are based on linear time-invariant (LTI) systems, there is increasing interest in SSMs based on time-varying systems Gu and Dao [2023], Hasani et al. [2022], Merrill et al. [2024].

The most closely related model to our proposed D-LinOSS is the original LinOSS, introduced in Rusch and Rus [2025]. While LinOSS was the first SSM built on oscillatory dynamics, several other deep learning models also incorporate oscillatory behavior. These include recurrent architectures like coupled oscillatory RNNs (coRNNs) Rusch and Mishra [2021a] and UnICORNNs Rusch and Mishra [2021b], as well as graph-based models such as Graph Coupled Oscillator Networks (GraphCON) Rusch et al. [2022].

6 Discussion and conclusion

We introduced D-LinOSS, an extension of the LinOSS model that incorporates learnable damping across all temporal scales. Through spectral analysis, we showed that existing LinOSS variants are rigidly defined by their discretization scheme and can only express a narrow set of dynamical behaviors. In contrast, D-LinOSS captures the full range of stable, damped oscillatory dynamics.

This expanded expressivity yields a 10–30$\times$ improvement on a synthetic regression task, and leads to consistent performance gains across eight real-world benchmarks. D-LinOSS outperforms all baselines considered in this work, including Transformer-based models, LSTM variants, other modern SSMs, and previous versions of LinOSS. Additionally, D-LinOSS reduces the LinOSS hyperparameter search space by 50% without adding any computational overhead. These results establish D-LinOSS as an efficient and powerful extension to the family of deep state space models.

While D-LinOSS demonstrates strong empirical results as a general sequence model, it is based on layers of LTI dynamical systems, which are fundamentally limited in their ability to capture certain contextual dependencies, such as the selective copying task (Gu and Dao [2023], Jing et al. [2019]). Building on the growing interest in time-varying SSMs sparked by Gu and Dao [2023], we aim to explore future work on selective variants of LinOSS that integrate the efficiency and expressiveness of LinOSS-type models with the powerful selectivity mechanism enabled by time-varying dynamics.

As D-LinOSS is inherently well-suited to represent temporal relationships with oscillatory structure, we aim to explore applications to domains where such patterns are fundamental. In particular, climate science, seismic monitoring, and astrophysics data all exhibit complex patterns governed by oscillatory behavior. Moving forward, we believe that D-LinOSS will play an increasingly central role in advancing machine-learning based approaches in domains grounded in the physical sciences.

Acknowledgments

References

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

Guy E Blelloch. Prefix sums and their applications. 1990.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It's raw! audio generation with state-space models. In *International Conference on Machine Learning*, pages 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*, 2021.

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, 2022.

Marco Gualtieri. Lecture notes on differential geometry, 2016. URL https://www.math.utoronto.ca/mgualt/courses/MAT1300F-2016/docs/1300-2016-notes-6.pdf.

Ramin Hasani, Mathias Lechner, Tsun-Hsuan Wang, Makram Chahine, Alexander Amini, and Daniela Rus. Liquid structural state-space models. *arXiv preprint arXiv:2209.12951*, 2022.

Zheyuan Hu, Nazanin Ahmadi Daryakenari, Qianli Shen, Kenji Kawaguchi, and George Em Karniadakis. State-space models are accurate and efficient neural operators for dynamical systems. *arXiv preprint arXiv:2409.03231*, 2024.

Li Jing, Caglar Gulcehre, John Peurifoy, Yichen Shen, Max Tegmark, Marin Soljacic, and Yoshua Bengio. Gated orthogonal recurrent units: On learning to forget. *Neural Computation*, 31(4): 765–783, 2019.

Shiva Kaul. Linear dynamical systems as a core computational primitive. *Advances in Neural Information Processing Systems*, 33:16808–16820, 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.

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

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

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

Samuel Lanthaler, T Konstantin Rusch, and Siddhartha Mishra. Neural oscillators are universal. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Chris Lu, Yannick Schroecker, Albert Gu, Emilio Parisotto, Jakob Foerster, Satinder Singh, and Feryal Behbahani. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Eric Martin and Chris Cundy. Parallelizing linear recurrent neural nets over sequence length. *arXiv preprint arXiv:1709.04057*, 2017.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. *arXiv preprint arXiv:2404.08819*, 2024.

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

NCEI. Local climatological data (lcd), 2025. URL `https://www.ncei.noaa.gov/data/local-climatological-data/`. Accessed: 2025-05-15.

Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35:2846–2861, 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*, pages 26670–26698. PMLR, 2023.

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

Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Lauren Milechin, Julia Mullen, Andrew Prout, Antonio Rosa, Charles Yee, and Peter Michaleas. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2018.

T. Konstantin Rusch and Siddhartha Mishra. Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations*, 2021a.

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

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

T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. Graph-coupled oscillator networks. In *International Conference on Machine Learning*, pages 18888–18909. PMLR, 2022.

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Ai8Hw3AXqks`.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

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

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

A   Theoretical properties

A.1   Derivation of the eigenvalues of D-LinOSS

Proposition A.1. *The eigenvalues of the D-LinOSS recurrent matrix* $\mathbf{M} \in \mathbb{R}^{2m \times 2m}$ *are*

$$\lambda_{i_{1,2}} = \frac{1 + \frac{\Delta t_i}{2}\mathbf{G}_i - \frac{\Delta t_i^2}{2}\mathbf{A}_i}{1 + \Delta t_i \mathbf{G}_i} \pm \frac{\frac{\Delta t_i}{2}\sqrt{(\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2 - 4\mathbf{A}_i}}{1 + \Delta t_i \mathbf{G}_i},$$

where pairs of eigenvalues are denoted as $\lambda_{i_{1,2}}$ *and* $i = 1, 2, ..., m$.

Proof. $\mathbf{M}$ (5) is a matrix with diagonal sub-blocks $\mathbf{M}_{11}$, $\mathbf{M}_{12}$, $\mathbf{M}_{21}$, and $\mathbf{M}_{22}$, i.e. it follows the structure:

$$\mathbf{M} = \begin{vmatrix} \begin{bmatrix} (\mathbf{M}_{11})_1 & & \\ & \ddots & \\ & & (\mathbf{M}_{11})_m \end{bmatrix} & \begin{bmatrix} (\mathbf{M}_{12})_1 & & \\ & \ddots & \\ & & (\mathbf{M}_{12})_m \end{bmatrix} \\ \begin{bmatrix} (\mathbf{M}_{21})_1 & & \\ & \ddots & \\ & & (\mathbf{M}_{21})_m \end{bmatrix} & \begin{bmatrix} (\mathbf{M}_{22})_1 & & \\ & \ddots & \\ & & (\mathbf{M}_{22})_m \end{bmatrix} \end{vmatrix}$$

Since $\mathbf{M}$ represents m uncoupled oscillatory systems, we see that the operation $\mathbf{M}\mathbf{w}$, $\mathbf{w}^\intercal = [\mathbf{z}^\intercal, \mathbf{x}^\intercal]$ can be independently split across the m pairs of state variables $[\mathbf{z}_i, \mathbf{x}_i]$. Thus, it suffices to re-arrange $\mathbf{M}$ into m different 2×2 systems and analyze the eigenvalues of each system separately. Substituting the real expressions for $\mathbf{M}_{11}$, $\mathbf{M}_{12}$, $\mathbf{M}_{21}$, and $\mathbf{M}_{22}$, and using brackets to denote the 2×2 matrix formed by diagonally indexing along the sub-blocks of $\mathbf{M}$, the ith system matrix is:

$$[\mathbf{M}]_i = \begin{bmatrix} \mathbf{S}_i^{-1} & -\Delta t_i \mathbf{S}_i^{-1} \mathbf{A}_i \\ \Delta t_i \mathbf{S}_i^{-1} & 1 - \Delta t_i^2 \mathbf{S}_i^{-1} \mathbf{A}_i \end{bmatrix}$$

A straightforward exercise using the definition $\mathbf{S} = \mathbf{I} + \Delta t \mathbf{G}$ leads us to the ith pair of eigenvalues.

$$
\begin{aligned}
\det(\lambda \mathbf{I} - [\mathbf{M}]_i) &= \det \begin{bmatrix} \lambda - \mathbf{S}_i^{-1} & \Delta t_i \mathbf{S}_i^{-1} \mathbf{A}_i \\ -\Delta t_i \mathbf{S}_i^{-1} & \lambda - 1 + \Delta t_i^2 \mathbf{S}_i^{-1} \mathbf{A}_i \end{bmatrix} \\
&= (\lambda - \mathbf{S}_i^{-1})(\lambda - 1 + \Delta t_i^2 \mathbf{S}_i^{-1} \mathbf{A}_i) + (\Delta t_i \mathbf{S}_i^{-1})(\Delta t_i \mathbf{S}_i^{-1} \mathbf{A}_i) \\
&= \lambda^2 + \lambda(\mathbf{S}_i^{-1}(\Delta t_i^2 \mathbf{A}_i - 1) - 1) + \mathbf{S}_i^{-1} \\
&= \lambda^2 + \lambda \frac{-2 - \Delta t_i \mathbf{G}_i + \Delta t_i^2 \mathbf{A}_i}{1 + \Delta t_i \mathbf{G}_i} + \frac{1}{1 + \Delta t_i \mathbf{G}_i} = 0
\end{aligned}
$$

$$\implies \lambda_{i_{1,2}} = \frac{1 + \frac{\Delta t_i}{2}\mathbf{G}_i - \frac{\Delta t_i^2}{2}\mathbf{A}_i}{1 + \Delta t_i \mathbf{G}_i} \pm \frac{\frac{\Delta t_i}{2}\sqrt{(\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2 - 4\mathbf{A}_i}}{1 + \Delta t_i \mathbf{G}_i}$$

$\square$

A.2   Proof of the stability criterion

Proposition A.2. *Assume* $\mathbf{G}_i$, $\mathbf{A}_i$ *are non-negative, and* $\Delta t_i \in (0, 1]$. *If the following is satisfied:*

$$(\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2 \leq 4\mathbf{A}_i \qquad (11)$$

Then $\lambda_{i_{1,2}}$ come in complex conjugate pairs λ_i, λ_i^* with the following magnitude:

$$|\lambda_i| = \frac{1}{\sqrt{1 + \Delta t_i \mathbf{G}_i}} \leq 1,$$

i.e. the eigenvalues are unit-bounded. Define $\mathcal{S}_i$ to be the set of all $(\mathbf{G}_i, \mathbf{A}_i)$ that satisfy the above condition. For notational convenience, we order the eigenvalues such that $\mathrm{Im}(\lambda_i) \geq 0$, $\mathrm{Im}(\lambda_i^*) \leq 0$.

Proof. $(\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2 - 4\mathbf{A}_i$ is the determinant of each eigenvalue pair, so $(\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2 \leq 4\mathbf{A}_i$ means λ_i can be written in complex form with the following real and imaginary components.

$$\mathrm{Re}(\lambda_i) = \frac{1 + \frac{\Delta t_i}{2}\mathbf{G}_i - \frac{\Delta t_i^2}{2}\mathbf{A}_i}{1 + \Delta t_i \mathbf{G}_i}$$

$$\mathrm{Im}(\lambda_i) = \pm\frac{\frac{\Delta t_i}{2}\sqrt{4\mathbf{A}_i - (\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2}}{1 + \Delta t_i \mathbf{G}_i}$$

The magnitude of this complex number is:

$$|\lambda_i| = \sqrt{\mathrm{Re}(\lambda_i)^2 + \mathrm{Im}(\lambda_i)^2}$$
$$= \frac{1}{\sqrt{1 + \Delta t_i \mathbf{G}_i}}$$

$$\Delta t_i, \mathbf{G}_i \geq 0 \implies |\lambda_i| \leq 1$$

We note that this stability criterion is a sufficient but not necessary condition. There exists solutions $(\mathbf{G}_i, \mathbf{A}_i)$ rendering $|\lambda_i| \leq 1$ without satisfying the proposed inequality. However, as shown in Proposition 3.3, there already exists a bijection from $\mathcal{S}_i$ to the full complex unit disk.

$\arg(\lambda_i) = \mathrm{atan2}(\mathrm{Im}(\lambda_i), \mathrm{Re}(\lambda_i))$ can be simplified by multiplying out the denominator $1 + \Delta t_i \mathbf{G}_i$ as it is always positive.

$$\arg(\lambda_i) = \mathrm{atan2}\left(\frac{\Delta t_i}{2}\sqrt{4\mathbf{A}_i - (\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2}, 1 + \frac{\Delta t_i}{2}\mathbf{G}_i - \frac{\Delta t_i^2}{2}\mathbf{A}_i\right).$$

$\square$

A.3 Spectral image of D-LinOSS

Proposition A.3. *The mapping $\Phi : \mathcal{S}_i \to \mathbb{C}_{|z| \leq 1}$ defined by $(\mathbf{G}_i, \mathbf{A}_i) \mapsto \lambda_i$ is bijective.*

Proof. Assuming $\Delta t_i \in (0, 1]$ and $(\mathbf{G}_i, \mathbf{A}_i) \in \mathcal{S}_i$, recall the polar form of λ_i:

$$|\lambda_i| = \frac{1}{\sqrt{1 + \Delta t_i \mathbf{G}_i}} \iff \mathbf{G}_i = \frac{1 - |\lambda_i|^2}{\Delta t_i |\lambda_i|^2}$$

So $\mathbf{G}_i \in \mathbb{R}_{\geq 0}$ has its own bijection onto $|\lambda_i| \in [0, 1]$. It now suffices to show that for any constant $\mathbf{G}_i$ (and thus $|\lambda_i|$), there is a bijection from $\mathbf{A}_i \in \mathbb{R}_{\geq 0}$ to $\arg(\lambda_i) \in [0, \pi]$. The remaining angles $[-\pi, 0]$ are then reachable through $\arg(\lambda_i^*) = -\arg(\lambda_i)$.

We proceed by replacing the atan2 function with $\tan^{-1}$. Since $\mathrm{range}(\tan^{-1}) = [-\frac{\pi}{2}, \frac{\pi}{2}]$, we first assume $\arg(\lambda_i) \in [0, \frac{\pi}{2}]$. The alternative case $\arg(\lambda_i) \in [\frac{\pi}{2}, \pi]$ can be conducted similarly noting that $\mathrm{atan2}(y, x) \in [\frac{\pi}{2}, \pi] \implies \mathrm{atan2}(y, x) = \tan^{-1}(\frac{y}{x}) + \pi$.

$$\theta_i := \arg(\lambda_i) = \tan^{-1}\left(\frac{\frac{\Delta t_i}{2}\sqrt{4\mathbf{A}_i - (\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2}}{1 + \frac{\Delta t_i}{2}\mathbf{G}_i - \frac{\Delta t_i^2}{2}\mathbf{A}_i}\right)$$

$$\tan^2(\theta_i)\left(1 + \frac{\Delta t_i}{2}\mathbf{G}_i - \frac{\Delta t_i^2}{2}\mathbf{A}_i\right)^2 = \frac{\Delta t_i^2}{4}\left(4\mathbf{A}_i - (\mathbf{G}_i - \Delta t_i\mathbf{A}_i)^2\right)$$

$$\mathbf{A}_i^2\left(\tan^2\theta_i+1\right)\left(-\frac{\Delta t_i^4}{4}\right)+\mathbf{A}_i\left(\tan^2\theta_i+1\right)\left(\Delta t_i^2+\frac{\Delta t_i^3}{2}\mathbf{G}_i\right)-\tan^2\theta_i\left(1+\frac{\Delta t_i}{2}\mathbf{G}_i\right)^2-\frac{\Delta t_i^2}{4}\mathbf{G}_i^2 = 0$$

This expression is quadratic in $\mathbf{A}_i$ with constants Δt_i and $\mathbf{G}_i$ and thus is solvable for $\mathbf{A}_i$. It remains to show that $\mathbf{A}_i \in \mathbb{R}_{\geq 0}$ to verify this inverse is well-defined.

Computing the determinant of the derived quadratic expression in $\mathbf{A}_i$ leaves us with:

$$\left(\tan^2\theta_i+1\right)^2\left(\Delta t_i^2+\frac{\Delta t_i^3}{2}\mathbf{G}_i\right)^2 - 4\left(\tan^2\theta_i+1\right)\left(\frac{\Delta t_i^4}{4}\right)\left(\tan^2\theta_i\left(1+\frac{\Delta t_i}{2}\mathbf{G}_i\right)^2+\frac{\Delta t_i^2}{4}\mathbf{G}_i^2\right)$$

Recalling the assumption that $\tan\theta_i \in [0, \frac{\pi}{2}]$, we employ the two inequalities $\tan^2\theta_i + 1 \geq 1$ and $\frac{\Delta t_i^2}{4}\mathbf{G}_i^2 \leq (1 + \frac{\Delta t_i}{2}\mathbf{G}_i)^2$ and divide by $(\tan^2\theta_i + 1)^2$. The determinant expression is then bounded from below by:

$$\geq \left(\Delta t_i^2+\frac{\Delta t_i^3}{2}\mathbf{G}_i\right)^2 - \Delta t_i^4\left(1+\frac{\Delta t_i}{2}\mathbf{G}_i\right)^2 = \Delta t_i^4\left(1+\frac{\Delta t_i}{2}\mathbf{G}_i\right)^2 - \Delta t_i^4\left(1+\frac{\Delta t_i}{2}\mathbf{G}_i\right)^2 = 0$$

This explicit construction of a well-defined inverse mapping $\theta_i \mapsto (\mathbf{G}_i, \mathbf{A}_i)$ renders Φ a bijection.

$\square$

A.4 Proof of the set measure of LinOSS eigenvalues

Proposition A.4. *For both LinOSS-IM and LinOSS-IMEX, the set of eigenvalues constructed from $\mathbf{A}_i \in \mathbb{R}_{\geq 0}$ and $\Delta t_i \in (0, 1]$ is of measure zero in $\mathbb{C}$.*

Proof. We first prove the proposition for LinOSS-IM. Recall the eigenvalues of $\mathbf{M}^{\text{IM}}$ (Rusch and Rus [2025]) are

$$\lambda_{i_{1,2}}^{\text{IM}} = \frac{1}{1 + \Delta t_i^2\mathbf{A}_i} \pm j\frac{\Delta t_i\sqrt{\mathbf{A}_i}}{1 + \Delta t_i^2\mathbf{A}_i}$$

Since $\lambda_{i_1} = \lambda_{i_2}^*$, it suffices to prove the proposition for the first eigenvalue, i.e. $\{\lambda_{i_1} \in \mathbb{C}|\mathbf{A}_i \in \mathbb{R}_{\geq 0}, \Delta t_i \in (0, 1]\}$ is a set of measure zero.

Lemma A.5. *Let $f : M \to N$ be a continuously differentiable map of manifolds where $dim M < dim N$. Then $f(M)$ is of measure zero in N (Gualtieri [2016]).*

Using Lemma A.5, it suffices to show that range$(\lambda_{i_1}^{\text{IM}})$ is a 1-manifold in $\mathbb{R}^2$ (an isomorphism to $\mathbb{C}$). By defining $t := \Delta t\sqrt{\mathbf{A}_i}$, we can equivalently write:

$$\lambda_{i_1}^{\text{IM}} = \left(\frac{1}{1 + t^2}, \frac{t}{1 + t^2}\right)$$

Where the domain of $\mathbf{A}_i \in \mathbb{R}_{\geq 0}, \Delta t_i \in (0, 1]$ translates to the domain of $t \in \mathbb{R}_{\geq 0}$. We note a few key properties of the map $t \mapsto (\frac{1}{1+t^2}, \frac{t}{1+t^2})$: it is continuously differentiable, it is bijective from $\mathbb{R}_{\geq 0}$ to range$(\lambda_{i_1}^{\text{IM}})$, and its inverse is differentiable. Thus the conditions of Lemma A.5 are satisfied, and we conclude that range$(\lambda_{i_1}^{\text{IM}})$ is an embedded 1-manifold in $\mathbb{R}^2$ and a set of measure zero.

Now we prove the proposition for LinOSS-IMEX. Recall the eigenvalues for $\mathbf{M}^{\text{IMEX}}$ Rusch and Rus [2025] are

$$\lambda_{i_{1,2}}^{\text{IMEX}} = \frac{1}{2}(2 - \Delta t_i^2 \mathbf{A}_i) \pm \frac{j}{2}\sqrt{\Delta t_i^2 \mathbf{A}_i(4 - \Delta t_i^2 \mathbf{A}_i)}$$

By defining $t := \Delta t^2 \mathbf{A}_i$, and again only considering the first eigenvalue, we equivalently write:

$$\lambda_{i_1}^{\text{IMEX}} = \left(1 - \frac{t}{2}, \frac{1}{2}\sqrt{t(4 - t)}\right)$$

Note that $\Delta t_i \leq \frac{2}{\sqrt{\mathbf{A}_i}}$ is an additional stability assertion of the IMEX model, so the domain $\mathbf{A}_i \in \mathbb{R}_{\geq 0}, 0 \leq \Delta t_i \leq \frac{2}{\sqrt{\mathbf{A}_i}}$ maps to $0 \leq t \leq 4$. It can be similarly shown that $t \mapsto (1 - \frac{t}{2}, \frac{1}{2}\sqrt{t(4 - t)})$ is a continuously differentiable diffeomorphism from the closed interval $[0, 4]$ to range($\lambda_{i_1}^{\text{IMEX}}$), and so we similarly conclude that range($\lambda_{i_1}^{\text{IMEX}}$) is a 1-manifold in $\mathbb{R}^2$ and thus a set of measure zero.

$\square$

A.5 Universality of LinOSS

We provide relevant concepts of operators and state a paraphrased version of the theorem that LinOSS blocks are universal Rusch and Rus [2025].

An operator mapping between continuous vector-valued functions $\Phi : C_0([0, T]; \mathbb{R}^p) \to C_0([0, T]; \mathbb{R}^q)$ is said to be *causal* if for any $t \in [0, T]$, if $\mathbf{u}, \mathbf{B} \in C_0([0, T]; \mathbb{R}^p)$ are two input signals such that $\mathbf{u}|_{[0,t]} \equiv \mathbf{B}|_{[0,t]}$, then $\Phi(\mathbf{u})(t) = \Phi(\mathbf{B})(t)$.

An operator is said to be *continuous* if $\Phi : (C_0([0, T]; \mathbb{R}^p), \|\cdot\|_\infty) \to (C_0([0, T]; \mathbb{R}^q), \|\cdot\|_\infty)$, i.e. Φ is a map between continuous functions with respect to the L^∞-norms on the input/output signals.

The theorem of Rusch and Rus [2025] is briefly paraphrased:

Theorem A.6. *Let Φ be any causal and continuous operator. Let $K \subset C_0([0, T]; \mathbb{R}^p)$ be compact. Then for any $\epsilon > 0$, there exists a configuration of hyperparameters and weight matrices, such that the output $\mathbf{z} : [0, T] \to \mathbb{R}^q$ of a LinOSS block satisfies:*

$$\sup_{t \in [0,T]} |\Phi(\mathbf{u})(t) - \mathbf{z}(t)| \leq \epsilon, \quad \forall \mathbf{u} \in K.$$

In other words, a LinOSS block can approximate any causal and continuous operator to arbitrarily high accuracy.

B Experiments and results

B.1 Parameterization

Recall that $(\mathbf{G}_i - \Delta t_i \mathbf{A}_i)^2 \leq 4\mathbf{A}_i$ is a sufficient condition for stability. This inequality can be equivalently expressed as

$$2 + \Delta t_i \mathbf{G}_i - 2\sqrt{\Delta t_i \mathbf{G}_i + 1} \leq \Delta t_i^2 \mathbf{A}_i \leq 2 + \Delta t_i \mathbf{G}_i + 2\sqrt{\Delta t_i \mathbf{G}_i + 1}.$$

Dividing both sides by Δt^2 arrives at our boundary expressions of $L(\mathbf{G}_i), U(\mathbf{G}_i)$, where $\mathbf{A}_i$ is clamped between the boundaries to guarantee stability.

$$L(\mathbf{G}_i) = \frac{2 + \Delta t_i \mathbf{G}_i - 2\sqrt{\Delta t_i \mathbf{G}_i + 1}}{\Delta t_i^2}, \tag{12}$$

$$U(\mathbf{G}_i) = \frac{2 + \Delta t_i \mathbf{G}_i + 2\sqrt{\Delta t_i \mathbf{G}_i + 1}}{\Delta t_i^2}, \tag{13}$$

$$\mathbf{A}_i = \text{Clamp}(\tilde{\mathbf{A}}_i, L(\mathbf{G}_i), U(\mathbf{G}_i)), \tag{14}$$

$\tilde{\mathbf{G}}_i, \tilde{\mathbf{A}}_i$ are the unconstrained variables learned during gradient descent and $\mathbf{G}_i = \text{Relu}(\tilde{\mathbf{G}}_i)$

B.2 Regression experiment

A small hyperparameter grid search was conducted, using hidden dimension $\in \{8, 64\}$, SSM dimension $\in \{8, 64\}$, and number of SSM blocks $\in \{2, 6\}$. Learning rate was kept constant at 1e-3.

To generate the data, sequences were sampled from white noise and passed through the dynamical system corresponding to parameters $A = 0.8, B = 1, C = 1, D = 0$. A sequence length of 1000 steps was selected as a sufficiently high value to study the model's behavior within the regime of long sequence learning.

B.3 Hyperparameters

For the UEA-MTSCA classification and PPG regression experiments of Section 4.1 and Section 4.2, model hyperparameters were searched over a grid of 162 total configurations defined by Walker et al. [2024]. This grid consists of learning rate $\in \{$1e-3, 1e-4, 1e-5$\}$, hidden dimension $\in \{16, 64, 128\}$, SSM dimension $\in \{16, 64, 256\}$, number of SSM blocks $\in \{2, 4, 6\}$, and inclusion of time $\in \{$True, False$\}$. For the weather forecasting experiment of Section 4.3, we instead perform random search over the same grid, except we sample learning rate continuously from a loguniform distribution and allow for odd-numbered selections of the number of blocks.

Table 5: Best performing hyperparameters for D-LinOSS across each of the eight datasets.

Dataset	lr	hidden dim	state dim	num blocks	include time
Worms	1e-3	128	64	2	False
SCP1	1e-4	128	256	6	True
SCP2	1e-5	128	64	6	False
Ethanol	1e-5	16	256	4	False
Heartbeat	1e-4	16	16	2	False
Motor	1e-3	16	64	4	False
PPG	1e-3	64	64	4	True
Weather	7.95e-5	128	128	3	False

B.4 Compute requirements

Below, we tabulate the compute resources required for each model across all datasets considered in the UEA-MTSCA classification experiments of Section 4.1. The main table is sourced from Rusch and Rus [2025], which lists the total number of parameters, average GPU memory usage measured in MB, and average run time per 1000 training steps measured in seconds. All models operate on the same codebase and python libraries, adopted from both Walker et al. [2024] and Rusch and Rus [2025]. These compute requirements are evaluated using an Nvidia RTX 4090 GPU.

B.5 Initialization techniques

To understand how to best initialize the magnitude of λ, we conduct a study on model performance comparing four different sampling techniques, each with four different sampling intervals.

- Experiment 1: Uniformly sample $\mathbf{G} \in [0, G_{\max}]$ for different values of $G_{\max}$ and uniformly sample $\mathbf{A} \in [0, 1]$

- Experiment 2: Uniformly sample $\mathbf{G} \in [0, G_{\max}]$ for different values of $G_{\max}$ and uniformly sample $\phi \in [0, \pi]$

- Experiment 3: Uniformly sample $|\lambda| \in [r_{\min}, 1]$ for different values of $r_{\min}$ and uniformly sample $\mathbf{A} \in [0, 1]$

Table 6: Compute requirements for the classification experiments considered in Section 4.1.

|		NRDE	NCDE	Log-NCDE	LRU	S5	Mamba	S6	LinOSS-IMEX	LinOSS-IM	D-LinOSS
Worms	$|\theta|$	105110	166789	37977	101129	22007	27381	15045	26119	134279	134279
	mem.	2506	2484	2510	10716	6646	13486	7922	6556	3488	3488
	time	5386	24595	1956	94	31	122	68	37	14	14
SCP1	$|\theta|$	117187	166274	91557	25892	226328	184194	24898	447944	991240	992776
	mem.	716	694	724	960	1798	1110	904	4768	4772	4790
	time	1014	973	635	9	17	7	3	42	38	38
SCP2	$|\theta|$	200707	182914	36379	26020	5652	356290	26018	448072	399112	399496
	mem.	712	692	714	954	762	2460	1222	4772	2724	2736
	time	1404	1251	583	9	9	32	7	55	22	22
Ethanol	$|\theta|$	93212	133252	31452	76522	76214	1032772	5780	70088	6728	71112
	mem.	712	692	710	1988	1520	4876	938	4766	1182	4774
	time	2256	2217	2056	16	9	255	4	48	8	37
Heartbeat	$|\theta|$	15657742	1098114	168320	338820	158310	1034242	6674	29444	10936	4356
	mem.	6860	1462	2774	1466	1548	1650	606	922	928	672
	time	9539	1177	826	8	11	34	4	4	7	4
Motor	$|\theta|$	1134395	186962	81391	107544	17496	228226	52802	106024	91844	20598
	mem.	4552	4534	4566	8646	4616	3120	4056	12708	4510	4518
	time	7616	3778	730	51	16	35	34	128	11	20
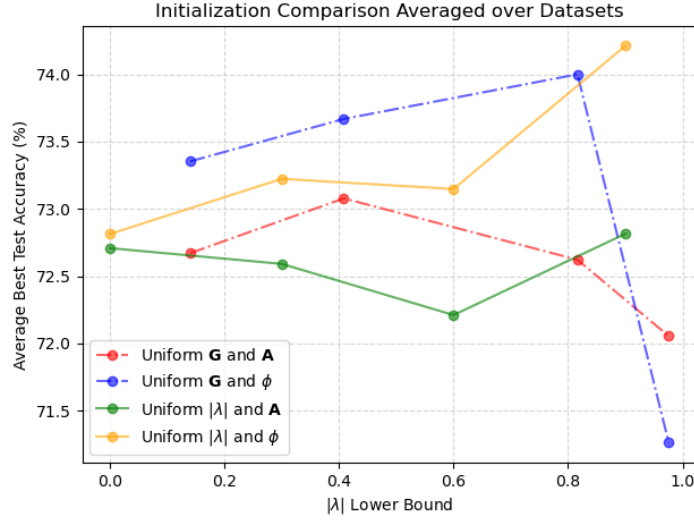
Figure 2: Initialization study varying intervals of eigenvalue magnitude and methods of sampling.

- Experiment 4: Uniformly sample $|\lambda| \in [r_{\min}, 1]$ for different values of $r_{\min}$ and uniformly sample $\phi \in [0, \pi]$

Where sampling $\mathbf{G}$ or sampling $|\lambda|$ are two different ways of controlling eigenvalue magnitude, and sampling $\mathbf{A}$ or sampling $\phi := \arg(\lambda)$ are two different ways of controlling eigenvalue phase. For each experiment, values of $G_{\max}$ and $r_{\min}$ are varied to generate different eigenvalue distributions. For each instance of an initialization technique (a single point on figure 2), D-LinOSS is trained over a small sweep of 16 hyper-parameter configurations. Each model instance is trained on 5 different seeds, and this process is repeated for the three datasets SelfRegulationSCP1, Heartbeat, and MotorImagery. Figure 2 displays the average highest test score over all three datasets.

Uniformly sampling eigenvalues in both phase and magnitude is the best performing initialization technique. For these three datasets, a lower bound of $r_{\min} = 0.9$ results in the highest average test accuracy. This technique is adopted for all empirical results in the paper.