Universality of Linear Recurrences Followed by Non-linear Projections: Finite-Width Guarantees and Benefits of Complex Eigenvalues

Antonio Orvieto [1] **Soham De** [2] **Caglar Gulcehre** [3] **Razvan Pascanu** [2] **Samuel L. Smith** [2]

Abstract

Deep neural networks based on linear RNNs interleaved with position-wise MLPs are gaining traction as competitive approaches for sequence modeling. Examples of such architectures include state-space models (SSMs) like S4, LRU, and Mamba: recently proposed models that achieve promising performance on text, genetics, and other data that require long-range reasoning. Despite experimental evidence highlighting these architectures' effectiveness and computational efficiency, their expressive power remains relatively unexplored, especially in connection to specific choices crucial in practice – e.g., carefully designed initialization distribution and potential use of complex numbers. In this paper, we show that combining MLPs with both real or complex linear diagonal recurrences leads to arbitrarily precise approximation of regular causal sequence-to-sequence maps. At the heart of our proof, we rely on a separation of concerns: the linear RNN provides a lossless encoding of the input sequence, and the MLP performs non-linear processing on this encoding. While we show that real diagonal linear recurrences are enough to achieve universality in this architecture, we prove that employing complex eigenvalues near unit disk – i.e., empirically the most successful strategy in S4 – greatly helps the RNN in storing information. We connect this finding with the vanishing gradient issue and provide experiments supporting our claims.

Note: The preliminary version of this manuscript (ICML workshop version) only contains a subset of the results. Our follow-up (Cirone et al., 2024) covers expressive power of gated SSMs such as Mamba (Gu & Dao, 2023).

[1]ELLIS Institute Tübingen, Max Planck Institute for Intelligent Systems, Tübingen AI Center, Tübingen, Germany [2]Google Deepmind [3]EPFL. Correspondence to: Antonio Orvieto <antonio@tue.ellis.eu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Attention (Vaswani et al., 2017) has supplanted LSTMs (Hochreiter & Schmidhuber, 1997) and GRUs (Cho et al., 2014) as the dominant sequence-to-sequence mechanism for deep learning. However, a promising line of research sparked by the S4 model (Gu et al., 2021) has initiated a come-back of recurrent sequence models in simplified *linear* form. A growing family of 'state-space' models (SSMs) (Gu et al., 2021; Hasani et al., 2022; Smith et al., 2023; Gu et al., 2022; Li et al., 2022a; Orvieto et al., 2023; Gu & Dao, 2023) has emerged which interleave recurrent linear complex-valued[1] layers with other components such as position-wise multi-layer perceptrons (MLPs), gates (Dauphin et al., 2017), residual connections (He et al., 2016), and normalization layers (Ioffe & Szegedy, 2015; Ba et al., 2016).

Traditional SSMs achieve state-of-the-art results on the long-range arena (Tay et al., 2020) and show outstanding performance in various domain including vision (Nguyen et al., 2022), audio (Goel et al., 2022), biological signals (Gu et al., 2021), reinforcement learning (Lu et al., 2023) and online learning (Zucchet et al., 2023b). SSMs, when augmented with a selectivity mechanism, are also successfully applied to text (Katsch, 2023; Gu & Dao, 2023; De et al., 2024), and are sometimes used in combination with softmax attention (Fu et al., 2023; Wang et al., 2023) or linear attention (Peng et al., 2023; Sun et al., 2023; Katsch, 2023; Yang et al., 2023). They gained significant interest in the literature due to two key factors. First, their computational complexity scales linearly in sequence length, while transformers scale quadratically (Vaswani et al., 2017; Katharopoulos et al., 2020). Second, unlike LSTMs and GRUs, training can be efficiently parallelized (Martin & Cundy, 2017)[2].

[1]The performance of non-selective SSMs trained *from scratch* on classification tasks is greatly affected by the choice of number field: using complex diagonal recurrences greatly improves performance, see discussion in (Gu et al., 2022; Orvieto et al., 2023) and, in particular, the S4D-Lin initialization. For Mamba on next-token prediction tasks, real numbers instead are effective. As shown by Amos et al. (2023), this is an effect due to a different learning paradigm which necessitates further investigations.

[2]At the current research state, effective parallelization in the time dimension is possible for recurrences which are linear in the state, and where the state-to-state transition is computationally light. Linear dense RNNs are parallelizable in principle but

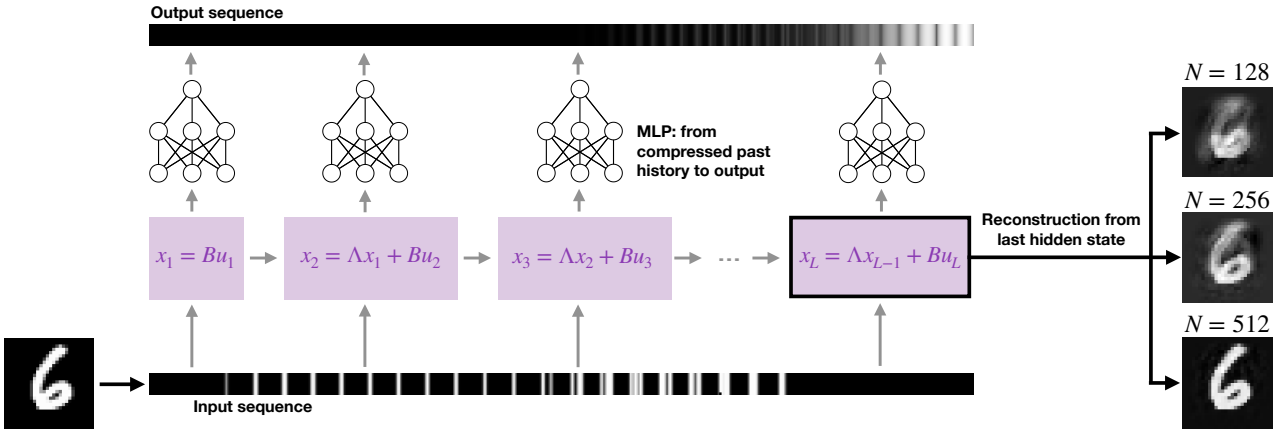

Figure 1. Illustration of a Linear RNN + position-wise MLP on flattened MNIST (LeCun, 1998) digits. In our construction, the role of the linear RNN is to compress (if possible) and store the input sequence into the hidden state: from hidden states one can recover past tokens using a linear transformation (see §3.1). As the hidden state size N increases, the reconstructions becomes more and more faithful. The MLP (same for all tokens) takes this representation as input, and is able to reproduce the action of any sufficiently regular sequence-to-sequence model (see §3.2). We provide additional insights and a thorough experimental evaluation and discussion in §4.

The success of these models is surprising, since it was previously widely thought that non-linearities within the recurrence are necessary for RNNs to model complex interactions (Schäfer & Zimmermann, 2006). In this paper, we therefore step back and study the theoretical expressivity of models based on stacks of vanilla[3] linear complex-valued RNNs interleaved with position-wise MLPs. In contrast to non-linear RNNs for which we have several expressivity results, from Turing-completeness (Chung & Siegelmann, 2021; Siegelmann & Sontag, 1992a) to universality (Schäfer & Zimmermann, 2006; Hanson & Raginsky, 2020), less is known for architectures based on linear recurrences, with nonlinearities placed outside of the sequential processing.

Results on SSMs expressivity. The Mütz-Szász Theorem (Müntz, 1914; Szász, 1916) can be used to show that linear diagonal real-valued RNNs can approximate causal convolutions of the input with an arbitrary filter in the width limit (Li et al., 2022b). However, linear filtering alone cannot provide satisfactory approximations of non-linear causal sequence-to-sequence maps. Yet, interestingly, Hanson & Raginsky (2019) showed that staking an exponential (in the sequence length) number of filters, chained with ReLUs, can approximate arbitrary time-homogeneous nonlinear causal operators. Recently, in the more realistic finite-depth regime of SSMs, Wang & Xue (2023) proved that interleaving five linear RNNs with nonlinearities leads, in the width limit, to the approximation of any continuous mappings from an input sequence to a scalar target. This result uses the Kolmogorov-Arnold Theorem (Kolmogorov, 1957). Approximating nonlinear causal sequence-to-sequence mappings (in the sense of Definition 1) is, of course, harder: Wang & Xue (2023) provided an interesting connection with Volterra-series expansions (Boyd & Chua, 1985) of time-invariant causal operators: multiplying the output of K infinitely wide linear RNNs provides a K-th order approximation of smooth non-linear sequence to sequence mappings. While this result establishes a connection to the rich literature in dynamical systems, it concerns a specific architecture design strategy (product of K RNN outputs), which is far from SSMs practice. Further, the discussion in Wang & Xue (2023) does not characterize the effect of choosing complex-valued recurrences (as crucially done in most SSMs, including S4) on expressivity. In this paper, we adopt a different strategy compared to Wang & Xue (2023), and aim instead at characterizing the information content in the RNN hidden state, further studying how this information is processed by the MLP across timestamps.

Contributions. We prove that a single linear diagonal RNN layer followed by a position-wise MLPs can approximate arbitrarily well any *sufficiently regular* (see Definition 1) nonlinear sequence to sequence map over finite length sequences. Our key insight is that, if the linear RNN can preserve a lossless compressed representation of the entire sequence seen so far, the MLP has access to all of the information in the input. Since MLPs are universal approximators (Pinkus, 1999), they can process this information to achieve any desired mapping (see Fig. 1). Along our reasoning, we give several insights on the effects of initialization and characterize the role of complex numbers on memory.

1. In §3.1, using the properties of Vandermonde matrices, we show that at a fixed timestamp k one can precisely

computationally heavy since unit operations involve dense matrix multiplications (Smith et al., 2023).

[3]For an analysis of selective SSMs such as Mamba (Gu & Dao, 2023) and Griffin (De et al., 2024), please check our follow-up (Cirone et al., 2024).

(zero error) reconstruct the value of each seen input token from the hidden state of a random diagonal linear RNN. We discuss how wide the RNN should be for this property to hold, and how the result adapts to the setting of compressible inputs. These results follow from straightforward linear algebra arguments but provide great insights into RNN memorization mechanisms. Our discussion has deep links to the HiPPO theory (Gu et al., 2020) but does not require structured matrices at the cost of stronger assumptions on the RNN width (§3.1.3).

2. In §3.2, we prove our claim: linear RNNs followed by position-independent MLPs with one hidden layer can approximate regular sequence-to-sequence maps. Starting from the results of §3.1, we use Barron's theory (Barron, 1993) to provide guarantees on the MLP width. This result involves technical steps such as the computation of the Barron constant of the interpolation of non-linear maps. Crucially, we find that the MLP width is affected by the ease of reconstruction from the linear RNN state, quantified by the condition number of the reconstruction map, which is heavily dependent on the RNN initialization.
3. In §4, we leverage our framework to understand some interesting features of basic SSMs such as S4 (Gu et al., 2021) and the LRU (Orvieto et al., 2023) – such as the need, e.g. when training from scratch on the Long-range Arena (LRA) (Tay et al., 2020), for complex-valued recurrences and initialization near the unit circle in $\mathbb{C}$. In the same section, we validate our claims and intuition on initialization on synthetic datasets and on LRA tasks (Tay et al., 2020).

Due to space limitations, proofs and additional related works are presented in the appendix.

2. Preliminaries

Consider length-L sequences of real M-dimensional inputs: $\bar{v} = (v_i)_{i=1}^L \in \mathcal{V} \subseteq \mathbb{R}^{M\times L}$. We often refer to each v_i as a "token"[4]. A *sequence-to-sequence map* is a deterministic transformation of input sequences that produces output sequences of the same length: $\bar{y} = (y_i)_{i=1}^L \in \mathcal{Y} \subseteq \mathbb{R}^{S\times L}$. We say that a sequence-to-sequence map is *causal* if for every k, y_k is blind to the tokens $(v_i)_{i=k+1}^L$.

Definition 1 (Sequence-to-sequence)**.** *A causal sequence-to-sequence map with length-L sequential M-dimensional inputs $\bar{v} = (v_i)_{i=1}^L \in \mathcal{V} \subseteq \mathbb{R}^{M\times L}$ and length-L sequential S-dimensional outputs $\bar{y} = (y_i)_{i=1}^L \in \mathcal{Y} \subseteq \mathbb{R}^{S\times L}$ is a sequence of non-linear continuous functions $T = (T_k)_{k=1}^L$, $T_k : \mathbb{R}^{M\times k} \to \mathbb{R}^S$ $\forall k \in [L]$. T acts as follows:*

$$(v_i)_{i=1}^L \overset{T}{\mapsto} (y_i)_{i=1}^L \ , \ \textit{s.t.} \ \ y_k = T_k((v_i)_{i=1}^k).$$

[4]In formal NLP (Cotterell et al., 2023), $\mathcal{V}$ is a finite vocabulary. Here, we discuss the setting where $\mathcal{V}$ is possibly dense in $\mathbb{R}^{M\times L}$.

We are going to assume without restating this that each T_k is a Barron function (Def. 3), with a well-defined integrable Fourier transform (used in Thm. 4). We consider approximations $\hat{T}$ to T using a neural network. Schematically:

$$\hat{T} = g \circ R \circ e,$$

where:

- $e : \mathbb{R}^M \to \mathbb{R}^H$ is a linear embedding layer with biases, acting tokenwise. We denote the encoded tokens sequence $\bar{u} = (u_i)_{i=1}^L \in \mathbb{R}^{H\times L}$, where $u_k = e(v_k) \in \mathbb{R}^H$, for all $k \in [L]$.
- R is a linear RNN processing the encoded input tokens $(u_i)_{i=1}^L$ producing a sequence of N-dimensional hidden states $(x_i)_{i=1}^L \in \mathbb{R}^{N\times L}$.
- $g : \mathbb{R}^N \to \mathbb{R}^S$ is a non-linear function, acting tokenwise, parametrized by an MLP. We have that $\hat{y}_k = g(x_k) \in \mathbb{R}^S$, for all $k \in [L]$.

The combination of g with R and e, which we denote as $g \circ R \circ e$, produces outputs $\hat{\bar{y}} = (\hat{y}_i)_{i=1}^L \in \mathbb{R}^{P\times L}$ which we desire to be close to $\bar{y} = T(\bar{v})$.

The Linear RNN R processes inputs recurrently. This operation is parametrized by matrices A and B.

Definition 2 (Linear RNN)**.** *$R_{A,B} : \mathbb{R}^{H\times L} \to \mathbb{R}^{N\times L}$ processes an (encoded) sequence $\bar{u} = (u_i)_{i=1}^L$ producing an output sequence of hidden states $\bar{x} = (x_i)_{i=1}^L \in \mathbb{R}^{N\times L}$ by means of the following recursion:*

$$x_k = Ax_{k-1} + Bu_k, \tag{1}$$

where $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times H}$, and $x_0 = 0 \in \mathbb{R}^N$.

Diagonal Linear RNNs. Linear RNNs have been shown to be very successful token-mixing components in deep architectures such as SSMs (Gu et al., 2021; Orvieto et al., 2023). A crucial feature of SSMs – making them appealing compared to non-linear variants such as LSTMs (Hochreiter & Schmidhuber, 1997) or GRUs (Cho et al., 2014)– is that the forward pass is cheap to compute by means of parallel scans (Blelloch, 1990). At the root of such fast computation is the diagonalizability property of linear RNNs.[5] Indeed, over the space of $N \times N$ non-diagonal real matrices, the set of non-diagonalizable (in the complex domain) matrices has measure zero (Bhatia, 2013). Hence, up to arbitrarily small perturbations, A is diagonalizable over the complex numbers, i.e. one can write $A = Q\Lambda Q^{-1}$, where $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_N) \in \mathbb{C}^{N\times N}$ gathers the eigenvalues of A,

[5]In principle, also non-diagonal linear RNNs can be parallelized. Yet, parallelizing non-diagonal RNNs involves multiplying dense matrices. To achieve a speedup over recurrent computation on modern hardware, the linear RNN has to be diagonal.

and the columns of $Q \in \mathbb{C}^{N\times N}$ are the corresponding eigenvectors.

$$x_k = Q\Lambda Q^{-1}x_{k-1} + Bu_k$$
$$\implies (Q^{-1}x_k) = \Lambda(Q^{-1}x_{k-1}) + (Q^{-1}B)u_k$$

By renaming $x_k \leftarrow Q^{-1}x_k$ and $B \leftarrow Q^{-1}B$, one arrives at the complex-valued diagonal recursion

$$x_k = \Lambda x_{k-1} + Bu_k. \tag{2}$$

The corresponding map $R_{\Lambda,B} : \mathbb{R}^{N\times L} \to \mathbb{C}^{N\times L}$ is such that $R_{A,B} = Q \circ R_{\Lambda,B}$, where Q is applied tokenwise. Since we are interested in architectures of the form $\hat{f} = g \circ R \circ e$, the linear transformation Q can be merged into g, at the price of having inputs for g with *doubled dimension* (real and imaginary parts[6]): $g : \mathbb{R}^{2N} \to \mathbb{R}^P$. Without loss in generality (see also Li et al. (2022b)), we therefore assume from now on our linear RNNs are in diagonal form. Morover, since the linear transformation Q can be merged with the output projection, we consider – as done in S4 and LRU – initialization directly on the eigenvalues, and *not passing through diagonalization* of a Glorot-initialized dense matrix A. Note that while the role of complex numbers in this setting is clearly motivated by the construction, we later show in §4 that using complex eigenvalues induces peculiar memorization properties.

MLPs and universality. MLPs with one hidden-layer (1HL-MLPs) are universal non-linear function approximators (Barron, 1993; Pinkus, 1999). In this paper, we consider g parametrized by a 1HL-MLP. g is a proxy for a general non-linear function f, which is regular in the sense that it is *not oscillating too quickly* – meaning that its Fourier transform $F(\omega)$ exhibits fast decay as $\|\omega\| \to \infty$.

Definition 3 (Barron function)**.** *Let $\mathcal{F}(\omega)$ be the Fourier transform of $f : \mathbb{R}^n \to \mathbb{R}$. f belongs to the Barron class if $C_f = \int_{\mathbb{R}^n} \|\omega\|_2 |\mathcal{F}(\omega)| d\omega < \infty$.*

We have the important result due to (Barron, 1993).

Theorem 1 (Universality of 1HL-MLPs)**.** *Consider $g(x)$ parametrized by a 1HL-MLP (with D hidden neurons): $g(x) = \sum_{k=1}^{D} \tilde{c}_k \sigma(\langle \tilde{a}_k, x\rangle + \tilde{b}_k) + \tilde{c}_0$, where σ is any sigmoidal function[7]. Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous with Barron constant C_f. If $D \geq 2r^2 C_f^2 \epsilon^{-2}$, then there exist parameters such that $\sup_{\|x\|\leq r} |f(x) - g(x)| \leq \epsilon$.*

Note that ReLU activations also work in the setting of the theorem above, up to a factor 2 in the bound, since $\mathrm{ReLU}(x) - \mathrm{ReLU}(x-1)$ is sigmoidal.

Remark 1 (Multidimensional Output)**.** *The result above is stated for the scalar output case. The S-dimensional outputs result can be easily derived by stacking neurons in the hidden layer, that becomes of dimension $D \leftarrow DS$. Therefore if $D \geq 2r^2 C_f^2 S \epsilon^{-2}$, then $\sup_{\|x\|\leq r} \|f(x) - g(x)\|_1 \leq \epsilon S$ (since errors accumulate). Let us then call $\epsilon \leftarrow \epsilon S$ the desired accuracy; the number of neurons needed to achieve that is $D \geq 2r^2 C_f^2 S^3 \epsilon^{-2}$.*

[6] In practice, some SSM variants drop the imaginary part.

[7] $\lim_{x\to-\infty} \sigma(x) = 0$ and $\lim_{x\to\infty} \sigma(x) = 1$.

3. Universality Result

In this section, we show that, as the model $\hat{T} = g \circ R \circ e$ grows in width, there exist network parameters such that $\hat{T} \approx T$. We state this result informally below.

Theorem 2 (Universality)**.** *Let the inputs set $\mathcal{V} \subset \mathbb{R}^{M\times L}$ be bounded and $\epsilon > 0$ be the desired accuracy in approximating T. Let R be a diagonal linear real or complex RNN with width $N \geq \dim(\mathcal{V})$ and let the MLP width $D \geq O(L/\epsilon^2)$. Then, $\hat{T} = g \circ R \circ e$ approximates pointwise T with error ϵ:*

$$\sup_{\bar{v}\in\mathcal{V}} \|\hat{T}(\bar{v}) - T(\bar{v})\| \leq \epsilon.$$

Here, by $\dim(\mathcal{V})$ we mean the *vector-space dimension* of $\mathcal{V}$. In the worst-case scenario where inputs are not structured, $\dim(\mathcal{V}) = LM$. In practice, we observe that one can work with smaller dimensions in the hidden state (Fig. 4). The proof comprises two steps:

- Step 1: Linear RNNs can perform lossless compression (§3.1): from the RNN output x_k one can perfectly (no error) reconstruct the input $(v_i)_{i=1}^k$, assuming N is large enough ($N \geq \dim(\mathcal{V})$).
- Step 2: The MLP head g on top of x_k can reconstruct the ground-truth map T_k: $T_k((v_i)_{i=1}^k) \simeq g(x_k)$, assuming the number of hidden MLP neurons D is large enough ($D \geq O(L/\epsilon^2)$).

While step 2 is mainly technical, step 1 – dealing specifically with the RNN – is less involved and leads to valuable insights into the architecture.

Remark 2. *A few comments on the result are needed:*

- *In Thm. 2, both the size of the RNN and the MLP agree with basic information theory reasoning: (a) if inputs are random and unstructured, the hidden state cannot perform compression: the RNN has to store $\mathcal{O}(L)$ reals, requiring $\mathcal{O}(L)$ hidden dimensions; (b) the MLP size is $\mathcal{O}(L)$ since, in the worst-case, it has to model L distinct maps $T_1, T_2, \ldots, T_L$ (Def. 1). This complexity can be reduced by assuming temporal smoothness (cf. §3.1.3).*
- *The setting of tokens living in a finite set (Cotterell et al., 2023) is drastically different and not discussed in this work. An interesting result using a construction derived from the first version of this paper can be found in Ding et al. (2023). Additional recent results can be found in Jelassi et al. (2024).*

- *A simple application of Barron's Theorem (Thm. 1) does not suffice to prove Step 2. In SSMs, the same MLP function is applied at each timestamp, ant therefore has to model the interpolation of $T_1, T_2, \ldots, T_L$. Adapting Barron's theory to this time-dependent setting is our main technical contribution.*

This section is dedicated to the proof of Thm 2, with the main intuition relevant for our discussion is presented in §3.1.1. Valuable insights on initialization and role of complex numbers can be derived from our proof strategy. These are discussed in §4.

3.1. Linear RNNs can perfectly memorize inputs

We first state the main result of this subsection.

Theorem 3 (Power of Linear RNNs, informal)**.** *Under no assumption on the encoded inputs set $\mathcal{U} \subseteq \mathbb{R}^{H\times L}$, if the hidden dimension scales with HL, then linear RNN-based processing comes with* ***no information loss****: inputs $(u_i)_{i=1}^k$ can be perfectly recovered from x_k, for any $k = 1, 2, \ldots, L$. If $\dim(\mathcal{U}) =: P < HL$ – i.e. if $\mathcal{U}$ is linearly* ***compressible*** *– the hidden dimension can be reduced to $\mathcal{O}(P)$.*

We encourage the reader to go through this subsection to get a proof idea. Insights and practical considerations will be then addressed thoroughly in §4. In §3.1.3, we compare this simple result with HiPPO theory (Gu et al., 2020): a framework that provides precise guarantees on the approximation error of *structured* linear RNNs at any width.

In this subsection, with "input" we refer to the encoded sequence $\bar{u} = e(\bar{v}) \in \mathbb{R}^{H\times L}$. Note that, as typically $H \geq M$, accurate reconstruction of $\bar{u}$ implies accurate reconstruction of $\bar{v}$. We start by unrolling Eq. (2): $x_1 = Bu_1$, $x_2 = \Lambda Bu_1 + Bu_2$, $x_3 = \Lambda^2 Bu_1 + \Lambda Bu_2 + Bu_3$, and

$$x_k = \sum_{j=0}^{k-1} \Lambda^j B u_{k-j}. \tag{3}$$

It is easy to realize that this operation stacks convolutions of the sequence $B\bar{u} \in \mathbb{C}^{N\times L}$ with N independent one-dimensional filters parametrized by $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$, and with form $(1, \lambda_i, \lambda_i^2, \ldots, \lambda_i^{L-1})$ for $i = 1, 2, \ldots N$.

3.1.1. Main idea

Let $H = M = 1$, the encoder e be the identity, and $B = (1, 1, \ldots, 1)^\top$. Then, Eq. (3) can be written as

$$x_k = \begin{pmatrix} \lambda_1^{k-1} & \lambda_1^{k-2} & \cdots & \lambda_1 & 1 \\ \lambda_2^{k-1} & \lambda_2^{k-2} & \cdots & \lambda_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_N^{k-1} & \lambda_N^{k-2} & \cdots & \lambda_N & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{pmatrix} = V_k u_{1:k}^\top. \tag{4}$$

where $u_{1:k} = v_{1:k} = (v_i)_{i=1}^k \in \mathbb{R}^{1\times k}$, and V_k is a Vandermonde matrix. As long as $N \geq k$, we can hope to recover $u_{1:k}$ by pseudoinversion of the Vandermonde:

$$v_{1:k}^\top = u_{1:k}^\top = V_k^+ x_k. \tag{5}$$

Indeed, note that if $N = k$ (number of equations = number of unknowns), the matrix V_k is invertible under the assumption that all λ_i are distinct, since (see e.g. (Bhatia, 2013)): $\det(V_k) = \prod_{1\leq i<j\leq k}(\lambda_i - \lambda_j) \neq 0$. If $N > k$ then under again the assumption that eigenvalues are distinct, the linear operation on the hidden state $V_k^+ x_k$ produces the input sequence without errors. To make this procedure work in the simplest setting, a sufficient condition is $N \geq L$. We summarize below.

Proposition 1 (Bijectivity)**.** *Let $M = 1$, $H = 1$ and let the encoder e be the identity. Consider input projection $B = (1, 1, \ldots, 1)^\top$ and recurrent matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ with $\lambda_i \neq \lambda_j$ for all $i \neq j$. Fix $k \in [L]$, let $R^k_{\Lambda,B} : \mathbb{R}^{H\times k} \to \mathbb{R}^N$ denote the map $(u_i)_{i=1}^k \mapsto x_k$. If $N \geq L$, $R^k_{\Lambda,B}$ is bijective*[8]*, with linear inverse.*

The reconstruction map (parametrized by V_k^+) is time-variant and has *time-dependent* output dimension $\mathbb{R}^k$. Since the RNN output we consider in this paper is a *time-independent MLP* head g, it is essential for the hidden state to also contain information about **time**. This is trivial to achieve with linear RNNs: consider $e : \mathbb{R}^M \to \mathbb{R}^{M+1}$ such that v_k is mapped to $u_k = (1, v_k)$ for all k. Let $B = ((1, 0, 0, \ldots, 0), (0, 1, 1, \ldots, 1))^\top$. Consider $\lambda_0 = 1$. Then, the first hidden state dimension will coincide with the index k, since $x_{k,0} = x_{k-1,0} + 1$ for all k. The construction above, allows the RNN to count the step it is on. In §3.2, we will use this fact to conclude the theoretical discussion.

Multidimensional setting. Since the RNN is linear, one can design M independent RNNs acting on separate input dimensions. Such RNNs can be combined into one single block using a properly designed B matrix.

3.1.2. RNNs can compress inputs, when possible

What we discussed in Theorem 3 is a **worst-case setting** that requires the RNN hidden state N to be of size proportional to the input sequence L. This is not surprising since we made no assumptions about the inputs $\bar{u}$: it is indeed not possible in general to store $\mathcal{O}(L)$ real numbers in a vector of size smaller than $\mathcal{O}(L)$ by means of a linear Vandermonde projection – unless such inputs are structured (cf. §3.1.3). However, the RNN hidden state dimension scales efficiently under low-dimensionality, which we refer to as *sparseness* in relation to sparse coding (Lewicki & Sejnowski, 2000).

Assumption A (Low-dimensional, a.k.a *sparse*, inputs)**.** *Let $\Psi = (\psi^i)_{i=1}^P$, with $\psi^i \in \mathbb{R}^{M\times L}$ for all $i = 1, 2, \cdots, P$ be*

[8] Note that this maps the input sequence to the last hidden state.

an ordered list of basis functions. For every input sequence $\bar{v}$, there exist coefficients $\alpha^v := (\alpha_i^v)_{i=1}^P \in \mathbb{R}^P$ such that $\bar{v} = \sum_{i=1}^P \alpha_i^v \psi^i$. We do not assume such decomposition is unique, nor to have access to the basis functions.

Let us discuss this assumption in the one-dimensional setting $M = 1$. Definitions and results carry out effortlessly to the multidimensional setting. In matrix form, Assumption A can be written as: there exists a real matrix $\Psi \in \mathbb{R}^{L\times P}$ such that, for all $\bar{v} \in \mathcal{V} \subset \mathbb{R}^{1\times L}$ there exists $\alpha^v \in \mathbb{R}^P$ such that $\bar{v}^\top = \Psi\alpha^v$. Let $\Psi_k \in \mathbb{R}^{k\times P}$ denote the first k columns of Ψ. The last equation implies $v_{1:k}^\top = \Psi_k \alpha_k^v$, where α_k is one among the estimates of the coefficients α^v holding until iteration k (e.g. one frequency component might be visible only after a specific k). Eq. (4) then implies:

$$x_k = V_k v_{1:k}^\top = V_k \Psi_k \alpha_k^v, \tag{6}$$

Under the assumption that $\Gamma_k := V_k\Psi_k$ is full rank, $\alpha_k^v = (\Gamma_k^\top\Gamma_k)^{-1}\Gamma_k^\top x_k$. We therefore get:

$$v_{1:k} = \Omega_k x_k, \quad \Omega_k := \Psi_k(\Gamma_k^\top\Gamma_k)^{-1}\Gamma_k^\top \in \mathbb{C}^{k\times N}. \tag{7}$$

General definition of Ω_k. In the non-sparse setting, it is easy to see that $\Omega_k = V_k^+$. Therefore, from this point in the paper *we will keep denoting as Ω_k the linear reconstruction map*. To make Ω_k act on real numbers, we consider instead its real/imaginary inputs representation in $\mathbb{R}^{2N\times N}$.

Ψ may be unknown. One does not need to assume a specific Ψ a priori: under input sparsity, we can guarantee approximation with a learned linear map on the state, which may be task-dependent. Even if the input is strictly-speaking not sparse, not all information might be relevant for prediction. In a way, sparsity mathematically quantifies the belief that the information content relevant to a certain task has a lower dimension compared to the original signal.

3.1.3. Comparison with HiPPO

HiPPO theory (Gu et al., 2020) describes the compression properties of continuous-time linear RNNs of the form $\dot{x}_t = Ax_t + Bu_t$, where $u : \mathbb{R} \to \mathbb{R}$ is a *smooth input* and $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^N$ have a specific fixed form (i.e. are *structured*). A common choice for A and B implements the LegS operator. At time t, the LegS operator maps the input to the N polynomial coefficients $c_i(t)$ such that $g^{(t)} = \sum_{i=1}^N c_i(t)p_i^{(t)}$ is the best possible polynomial approximation of $u_{[0,t]}$ under the uniform measure in time on the interval $[0, t]$. If the inputs are Lipschitz continuous (smooth), they can indeed be approximated by an N-degree polynomial: the error is then simply the sum of the small polynomial coefficients above degree N: $|u_{[0,t]} - g^{(t)}|^2 = O(t^2\ell^2/N)$. This bound is due to the fundamental properties of polynomials and can be considered independently from the HiPPO framework: being ℓ-Lipschitz implies an error $O(t^2\ell^2/N)$ from inputs being a projection on a specific basis on N orthogonal polynomials. In particular, note that assuming the error is $O(t^2\ell^2/N)$ from being described by the first N coefficients is actually a weaker requirement compared to ℓ-Lipshitzness. This is linked with sparseness : $u_{[0,t]} = \sum_{i=1}^N c_i(t)p_i^{(t)}$ can be framed as Assumption A. While HiPPO theory, in contrast to this paper, allows for precise error estimates, our approach is more general. We show that it is not needed to fix an input measure or to hand-pick A, B: random initialization on a ring can already lead to successful reconstruction in practice (Fig. 4). Our worst-case result holds if $N \geq L$, *i.e. for the general non-smooth input setting*. Furthermore, as showcased above, Assumption A can be linked to polynomial regression, and therefore describes a general property – at the price of missing error quantifications. This can be resolved: for a fixed basis Ψ, it is possible to provide bounds using the famous lemma of Johnson et al. (1986): a similar analysis is used to quantify the error of randomized signatures in Rough Paths Theory (Cuchiero et al., 2021; Compagnoni et al., 2023).

Compared to HiPPO, an advantage of our simplified (yet less precise) approach is that we can easily move to the question of expressivity, which we explore next. Further, our non-structured approach leads to additional insights into the role of complex eigenvalues, which we explore in §4.

3.2. MLP Reconstruction

Recall that our goal is to approximate the object $T = (T_i)_{i=1}^L$, that maps $(v_i)_{i=1}^L \overset{T}{\mapsto} (y_i)_{i=1}^L$ such that $y_k = T_k((v_i)_{i=1}^k)$ for all $k = 1, 2, \ldots, L$. In the usual SSMs pipeline, this approximation takes the form $\hat{T} = g \circ R \circ e$, where compositions are tokenwise (MLP g and embedding e applied to each token), and R is a linear diagonal RNN, mixing tokens in the temporal direction.

Let x_k be the k-th RNN output. In the settings described in the last subsection, for every $k \in \{1, 2, \ldots, L\}$ there exists a linear function Ω_k' (identity on the first coordinate $x_{k,0} = k$, and Ω_k on the other coordinates) such that $\Omega_k' x_k = (k, (v_i)_{i=1}^k)$. Note that under no assumption on the set of inputs, Ω_k coincides with the Vandermonde inverse V_k^+. The last operation, $(k, v_{1:k}) \mapsto y_k$ needs to be performed by the *time-independent* map g.

Step 1: fixed timestamp. Let us focus on getting the MLP to approximate T_k, k fixed, from the RNN output. g has to satisfy $g(x_k) \overset{!}{=} T_k(\Omega_k(x_k)) =: f_k(x_k)$. The following result (based on Rmk.1) bounds the width of the MLP g for approximation of this quantity.

Proposition 2 (MLP, single timestamp)**.** *The number of hidden neurons in the MLP g sufficient to approximate $f_k := T_k \circ \Omega_k$ at level ϵ from the RNN hidden state x_k is bounded by $4r^2C_{T_k}^2\|\Omega_k\|_2^2 S^3\epsilon^{-2}$ where r bounds the hidden state magnitude, and C_{T_k} is the Barron constant of T_k.*

The result is easy to parse: to approximate the output at a specific timestamp k, the width bound is similar to that of an MLP $(v_i)_{i=1}^k \overset{g_k}{\mapsto} \hat{y}_k$, which would be $4r^2C_{T_k}^2S^3\epsilon^{-2}$ (see Thm. 1). The additional factor $\|\Omega_k\|_2^2$ appears because g operates on the hidden state and not from the input. Hence, the MLP has to pay a **penalty for reconstruction**. The result is based on the computation of the Barron constant for the map f_k, which is upper bounded by $\|\Omega_k\|_2 C_{T_k}$ (see appendix). In the next paragraph, we study how complexity is affected by the additional requirement that the map g should be able to interpolate in-between T_ks, based on the first coordinate in the hidden state $x_{k,1} = k$.

Step 2: arbitrary timestamp. To construct a time-independent global approximation, we use recent results from harmonic analysis (Tlas, 2022) to construct a proper domain relaxation and operate with a single MLP. This computation leads to a formula for the Barron complexity for function sequences, where an input dimension determines the function the MLP should implement. In addition to the definition $C_f = \int_{\mathbb{R}^n} \|\omega\|_2 |\mathcal{F}(\omega)| d\omega$, define the Fourier norm $C'_f = \int_{\mathbb{R}^n} |\mathcal{F}(\omega)| d\omega$.

Theorem 4 (Combination of Barron Maps)**.** *Let $f : [1, 2, \ldots, L] \times \mathbb{R}^n \to \mathbb{R}$ be such that each $f(k, \cdot)$ is Barron with constant C_{f_k} and Fourier norm C'_{f_k}. There exists an extension $\tilde{f} : \mathbb{R}^{n+1} \to \mathbb{R}$ of f, s.t. $\tilde{f}$ is Barron with constant $\tilde{C} \leq \mathcal{O}(\sum_{k=1}^L C_{f_k} + C'_{f_k}) = \mathcal{O}(L)$ and $\tilde{f}(x, k) = f(x, k), \forall x \in \mathbb{R}^n$ and $k \in 1, 2, \ldots, L$.*

This result, which essentially confirms the need, under no further assumptions on T, to blow up the number of hidden neurons with the sequence length (unless some smoothness is assumed, see Remark 2), directly leads to our main result (Thm. 2) under the assumption that the RNN has perfect memory, guaranteed if $N \geq \dim(\mathcal{V})$ (Thm. 3).

Proof of Thm. 2. Let $f_k := T_k \circ \Omega_k : \mathbb{R}^{2N} \to \mathbb{R}^S$. By Prop. 2, $C_{f_k} = \|\Omega_k C_{T_k}\|$. C'_{f_k} scales similarly. We have $f_k(x_k) = T_k(v_{1:k})$ thanks to Thm. 3. We then apply Thm. 4 to the sequence of f_k (one for each timestamp): there exists an interpolating function $\tilde{f}$ (time as first component followed by the $2N$ arguments of f_k) with Barron constant $\mathcal{O}(\sum_{k=1}^L C_{f_k} + C'_{f_k}) = \mathcal{O}(L)$. We let this be the function the 1-HL MLP g has to approximate, and apply Thm. 1.

4. Validation and Discussion

In this section, we revisit and validate our claims, and further discuss the practical insights associated with input reconstruction from the linear RNN hidden state.

4.1. Conditioning and role of complex numbers

In §3.1.1, we discussed the main idea behind reconstruction from the linear RNN hidden state. Let us restrict attention to the last timestamp: under no sparseness assumption (discussed at the end of this subsection) $x_L = V_k \bar{v}^\top$, with $V_L \in \mathbb{C}^{N \times L}$, hence $\bar{v}^\top = V_L^+ x_L$, where $V_L^+ = (V_L^\top V_L)^{-1} V_L^\top =: \Omega_L$. Indeed, if the eigenvalues $(\lambda_i)_{i=1}^N$ are distinct and $N \geq L$, $V_L^\top V_L \in \mathbb{C}^{L \times L}$ is invertible since the Vandermonde determinant formula ensures V_L has full column rank[9]. Such computation does not take into account a *numerical issue*: if V_L is **ill-conditioned**, $\|V_L^+\|_2 \to \infty$, preventing successful reconstruction. Instead, when initializing or learning the RNN, not only can we assume the eigenvalues are distinct, but we can also control them. Let us pick $\lambda_i s$ uniformly on the complex ring in between $[r_{\min}, r_{\max}] \subseteq [0, 1]$: For $i \in [N]$, we pick λ_i i.i.d with

$$\lambda_i \sim \mathbb{T}[r_{\min}, r_{\max}] := \{\lambda \in \mathbb{C} \mid r_{\min} \leq |\lambda| \leq r_{\max}\}.$$

Initialization of Λ **close to the unit circle** is often used in the context of modern state-space models to unlock long-range reasoning (Orvieto et al., 2023). Our theory gives grounding to this strategy: we see in Fig. 2 that if $r_{\min} \to 1$, then the Vandermonde becomes well conditioned and reconstruction becomes successful (Fig. 3 + appendix).

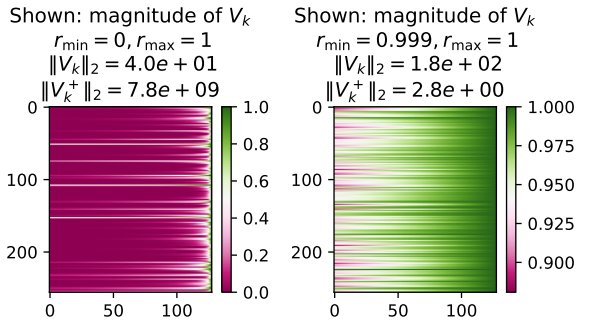

Figure 2. Effect of eigenvalue magnitude on conditioning. $L = 128$, $N = 2L = 256$, $\lambda_i \sim \mathbb{T}[r_{\min}, r_{\max}]$.

One can also link this effect to vanishing gradients: if $|\lambda_i| \to 0$ then the hidden state "forgets" inputs at a rate λ^k. This is the reason why in Fig. 3 we observe successful reconstruction only for the recent past if $r_{\min} = 0$.

Crucial role of complex numbers. If Λ is real the condition number of V_k grows exponentially with N (Gautschi & Inglese, 1987). In the complex setting it is possible to make the condition number of V_k exactly 1 by e.g. choosing the Nth-roots of unity as eigenvalues of Λ (Gautschi, 1975; Córdova et al., 1990). This fact provides a *precise justification for the use of complex numbers in the recurrent computation*: diagonal real recurrent RNNs can be also implemented fast with parallel scans (Smith et al., 2023), but suffer from an inherent information loss.

[9] By contradiction, assume $V_L^\top V_L$ has the zero eigenvalue. This implies there exists x such that $V_L^\top V_L x = 0$, hence $x^\top V_L^\top V_L x = \|V_L x\| = 0$, which is true if and only if $V_L x = 0$, meaning V_L is not full column rank.

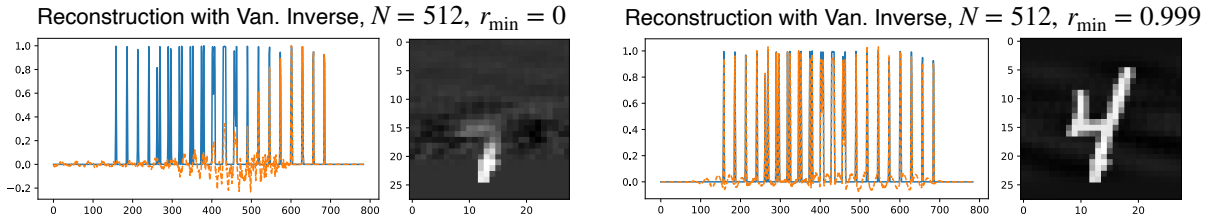

Figure 3. Reconstruction of MNIST digits from the final linear RNN hidden state using the Vandermonde inverse. For $r_{\min} = 0$, the Vandermonde is ill-conditioned (Fig. 2) and hence only the recent past can be reconstructed. For $r_{min} = 0.99$ we can reconstruct the whole image. See Fig. 4 for results on learned reconstructions.

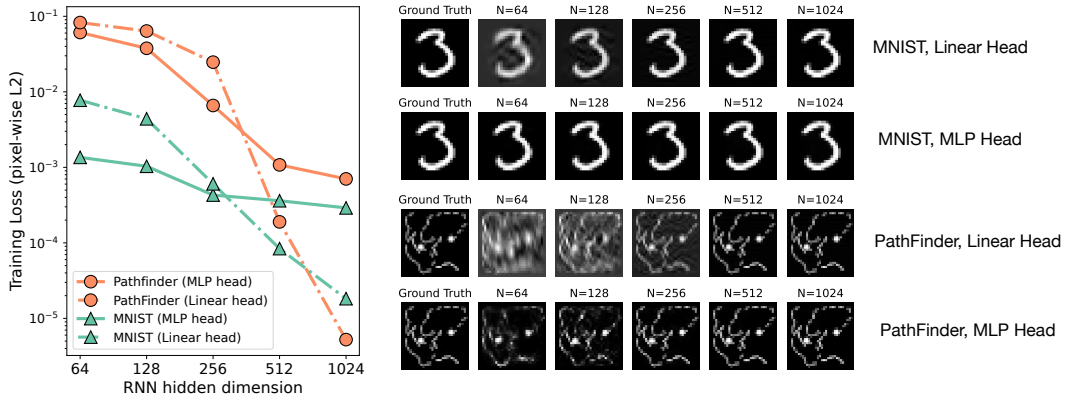

Figure 4. Reconstruction of MNIST digits and PathFinder data (Tay et al., 2020) Using a trained RNN + MLP or linear reconstruction decoder *from the last hidden state* x_L. Plotted is average L2 pixel-wise norm (mean of 3 runs). All parameters are trained, hyperparameters are tuned. $r_{\min} = 0.9$, $r_{\max} = 0.999$ are found to be best for initialization of the RNN (in line with (Gu et al., 2021)). For large hidden dimension, linear reconstruction is successful. For smaller hidden dimension, non-linear reconstruction becomes necessary.

Comparison with FFT. The case of equally-spaced eigenvalues on the disk coincides with the computation of the Fourier Transform. Perfect reconstruction in this setting is guaranteed by the inverse Fourier Transform Theorem (Folland, 2009). While this would motivate the use of unitary RNNs (Arjovsky et al., 2016; Helfrich et al., 2018), as we will see in §4.3 best performing learned architectures do not have this property: hidden-state features with vanishing memory are often the best option.

Sparseness improves conditioning. In the last section, we discussed how input sparseness (on a basis of P functions) results in a reduced hidden state dimension requirement. Specifically, if $\dim(\mathcal{V}) = P$, i.e. there exists $\Psi \in \mathbb{R}^{L\times P}$ such that for any $\bar{v} \in \mathcal{V}$ there exists $\alpha^v \in \mathbb{R}^P$ such that $\bar{v}^\top = \Psi\alpha^v$, then an RNN with width $N = P$ achieves perfect recall for inputs in $\mathcal{V}$. In Fig. 12 (appendix) we show that sparseness also improves conditioning of the reconstruction map Ω_L, appearing in Prop. 2.

4.2. Learned architecture for reconstruction

In the previous subsection we studied the reconstruction map computed from the parameters of a **random** linear RNN. While our theory provides satisfactory approximation guarantees in this setting, performance is much stronger if we allow the linear RNN to be ***trained*** *jointly with the reconstruction map*. We use here a simple LRU block (at inference time, a linear RNN (Orvieto et al., 2023)) followed by a linear reconstruction head Ω. This framework, while perfectly in line with our settings in this paper, allows the RNN eigenvalues and the decoder to synchronously adapt to the specific data structure. Out of curiosity, we also test here the performance of non-linear reconstruction from the RNN hidden state with a 1-HL MLP with 2048 hidden units[10]. We use data from the long-range arena (Tay et al., 2020), specifically sequential MNIST (784 tokens) and PathFinder (1024 tokens) and train our model[11] under the usual guidelines from random initialization (see discussion in (Orvieto et al., 2023)).

1. For both datasets, the reconstruction error (input of reconstruction map is the last hidden state) vanishes as the RNN hidden dimension increases. At $N = 256$, RNN width used in practice on the long-range arena (Gu et al., 2021; Orvieto et al., 2023), reconstruction is nearly perfect. For $N = 1024$, the error is negligible (as predicted by Thm. 3) since $N \geq L$.

2. Reconstruction is surprisingly accurate at small values

[10] This is not the function g that allows approximation of T_k. Here the MLP serves as non-linear decoder.

[11] We train on a single Nvidia RTX A5000 for 100 epochs.

of N if the decoder is non-linear. Instead, for $N = 1024$, the error is smallest with a linear decoder, a finding we believe is rooted in optimization of wide MLPs.

4.3. Approximation of non-linear ODEs.

In §4.1&4.2 we studied the information content of linear RNNs hidden states and verified empirically that near-perfect input reconstruction is possible, even if $N < L$. We now conclude the paper with results verifying our main claim (Thm. 2): a single MLP applied to the RNN hidden states, independent of the timestamp, is able to reconstruct the output of regular sequence-to-sequence maps (Def. 1). An example of a sequence-to-sequence map is the flow of a controlled differential equation, of form $\dot{z}_t = f(z_t, v_t), y_t = h(z_t)$, where $(v_t)_t$ is the input, f is a non-linear multidimensional function, and h projects the multidimensional state z_t into a one-dimensional output. Such systems can model complex interactions and stand at the root of physics. We limit our discussion here to the problem of learning a *protein transduction (PT) system* (Vyshemirsky & Girolami, 2008).*The reader can find results for other non-linear ODEs in the appendix.*
In PT, z_t is 5-dimensional, and f includes multiplicative interactions between components of z_t as well as feedback (see equations in the appendix). We include an input to the right-hand-side of the first ODE while integrating with Runge-Kutta 45, and take h as the projection of the first component of z_t. We learn to mimic the PT system with a linear RNN with $N = 128$, followed by a 1-HL MLP with $D = 256$. We sample $10k$ smooth inputs and train on the simulated outputs for $L = 2048$ integration steps. We test on $1k$ additional trajectories. The results in Fig. 5 clearly show that the architecture studied in this paper is able to learn the PT sequence-to-sequence map: plotted are two out-of-sample trajectories (not used during training), showcasing that our model prediction matches the PT dynamics up to a small error (as confirmed by the low test loss). We also include a depiction of the learned eigenvalues, to confirm that the RNN has indeed vanishing memory – but this does not necessarily imply information loss.

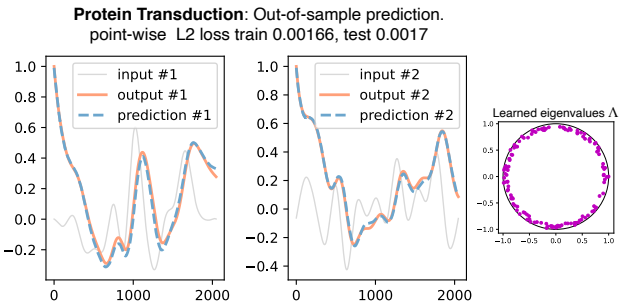

Figure 5. Trained linear RNN + MLP architecture (1 layer) learns a non-linear sequence-to-sequence map. Additional experiments can be found in the appendix.

5. Conclusion

In this paper we studied the expressive power of architectures combining linear RNNs with position-wise MLPs. These models recently exhibited remarkable performance in the context of long-range reasoning. In our theoretical framework, these two blocks work in symbiosis while having distinct roles: the linear RNN provides a compressed (where possible) representation of the input sequence, while the MLP performs non-linear processing. We believe our *separation of concerns* principle takes us one step further in understanding how to design and conceptualize deep state-space models: non-linear sequential processing can be performed by a time-independent component (the MLP). Further, the use of complex numbers in the recurrence unlocks lossless compression via a well-conditioned reconstruction map. In addition, our analysis provides guidelines for asymptotic scaling of architecture components, and can lead to novel initialization strategies based on local objectives (e.g. successful reconstruction).

Future work. Our work leaves open interesting avenues for future research, e.g. the finite-alphabet setting, the finite precision setting, and the framework of Turing completeness. Regarding the expressivity of recent selective RNNs such as Mamba and Griffin, an extension of our theory can be found in Cirone et al. (2024). Last, we note that, as remarked in §3.1.3, our framework does not model errors in reconstruction present at moderate hidden state dimensions. We believe a unified formal analysis, comprising assumptions such as smoothness in the inputs and tools such as the Johnson et al. (1986) lemma, can fill this gap.

Insights from Mamba & *Never train from scratch*. The first version of this paper, in the form of a short note, dates back to May 2023. In late 2023, selective SSMs (Gu & Dao, 2023) were introduced: these SSMs *do not use of complex numbers*, have input-controlled state transitions, and are almost exclusively trained autoregressively (on text). Amos et al. (2023) brilliantly pointed out that next-token prediction leads to a conceptually distinct objective compared to standard supervised learning: while training from scratch requires careful parametrization, initialization, and memorization, pretraining on denoising objectives can be more forgiving: transformers and less sophisticated S4 variants are able to get high accuracy on the LRA benchmark (Tay et al., 2020). Our paper, and specifically our discussion on complex numbers, is instead motivated by interesting results and the ablations holding for non-selective SSMs trained from scratch (Gu et al., 2021; Orvieto et al., 2023). We believe the supervised learning setting is scientifically interesting and relevant, and that the analysis presented here provides a precise motivation for effective design strategies. While the next-token prediction paradigm is becoming a standard, the peculiarities of supervised learning can yield insights into the next generation of generative models.

Acknowledgements

Antonio Orvieto acknowledges the financial support of the Hector Foundation.

Impact Statement

The authors do not foresee any societal consequences or ethical concerns arising from their work. This is a theoretical paper.

References

Amos, I., Berant, J., and Gupta, A. Never train from scratch: Fair comparison of long-sequence models requires data-driven priors. In *International Conference on Learning Representations*, 2023.

Arjovsky, M., Shah, A., and Bengio, Y. Unitary evolution recurrent neural networks. In *International conference on machine learning*. PMLR, 2016.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Bhatia, R. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Blelloch, G. E. Prefix sums and their applications, 1990.

Boyd, S. and Chua, L. Fading memory and the problem of approximating nonlinear operators with volterra series. *IEEE Transactions on circuits and systems*, 32(11):1150–1161, 1985.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chung, S. and Siegelmann, H. Turing completeness of bounded-precision recurrent neural networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 28431–28441. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ef452c63f81d0105dd4486f775adec81-Paper.pdf.

Cirone, N. M., Orvieto, A., Walker, B., Salvi, C., and Lyons, T. Theoretical foundations of deep selective state-space models. *arXiv preprint arXiv:2402.19047*, 2024.

Compagnoni, E. M., Scampicchio, A., Biggio, L., Orvieto, A., Hofmann, T., and Teichmann, J. On the effectiveness of randomized signatures as reservoir for learning rough dynamics. In *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023.

Córdova, A., Gautschi, W., and Ruscheweyh, S. Vandermonde matrices on the circle: spectral properties and conditioning. *Numerische Mathematik*, 57(1):577–591, 1990.

Cotterell, R., Svete, A., Meister, C., Liu, T., and Du, L. Formal aspects of language modeling. *arXiv preprint arXiv:2311.04329*, 2023.

Cuchiero, C., Gonon, L., Grigoryeva, L., Ortega, J.-P., and Teichmann, J. Expressive power of randomized signature. In *The Symbiosis of Deep Learning and Differential Equations*, 2021.

Dao, T. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 2022.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modeling with gated convolutional networks. In *International conference on machine learning*. PMLR, 2017.

De, S., Smith, S. L., Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., Desjardins, G., Doucet, A., Budden, D., Teh, Y. W., Pascanu, R., Freitas, N. D., and Gulcehre, C. Griffin: Mixing gated linear recurrences with local attention for efficient language models, 2024.

Ding, Y., Orvieto, A., He, B., and Hofmann, T. Recurrent distance-encoding neural networks for graph representation learning. *arXiv preprint arXiv:2312.01538*, 2023.

Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. Ode parameter inference using adaptive gradient matching with gaussian processes. In *Artificial intelligence and statistics*, pp. 216–228. PMLR, 2013.

Folland, G. B. *Fourier analysis and its applications*, volume 4. American Mathematical Soc., 2009.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Re, C. Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*, 2023.

Funahashi, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3): 183–192, 1989.

Gautschi, W. Optimally conditioned vandermonde matrices. *Numerische Mathematik*, 24:1–12, 1975.

Gautschi, W. and Inglese, G. Lower bounds for the condition number of vandermonde matrices. *Numerische Mathematik*, 52(3):241–250, 1987.

Goel, K., Gu, A., Donahue, C., and Ré, C. It's raw! audio generation with state-space models. *International Conference on Machine Learning*, 2022.

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

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Gu, A., Goel, K., and Re, C. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2021.

Gu, A., Gupta, A., Goel, K., and Ré, C. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 2022.

Gupta, A., Gu, A., and Berant, J. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*, 2022.

Haar, A. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 71(1):38–53, 1911.

Hanson, J. and Raginsky, M. Universal approximation of input-output maps by temporal convolutional nets. *Advances in Neural Information Processing Systems*, 32, 2019.

Hanson, J. and Raginsky, M. Universal simulation of stable dynamical systems by recurrent neural nets. In *Learning for Dynamics and Control*, pp. 384–392. PMLR, 2020.

Hasani, R., Lechner, M., Wang, T.-H., Chahine, M., Amini, A., and Rus, D. Liquid structural state-space models. In *The International Conference on Learning Representations*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Helfrich, K., Willmott, D., and Ye, Q. Orthogonal recurrent neural networks with scaled cayley transform. In *International Conference on Machine Learning*. PMLR, 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 1997.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

Jelassi, S., Brandfonbrener, D., Kakade, S. M., and Malach, E. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*, 2024.

Johnson, W. B., Lindenstrauss, J., and Schechtman, G. Extensions of lipschitz maps into banach spaces. *Israel Journal of Mathematics*, 1986.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pp. 5156–5165. PMLR, 2020.

Katsch, T. Gateloop: Fully data-controlled linear recurrence for sequence modeling, 2023.

Kitaev, N., Kaiser, Ł., and Levskaya, A. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Kolmogorov, A. N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Doklady Akademii Nauk*, pp. 953–956. Russian Academy of Sciences, 1957.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Lewicki, M. S. and Sejnowski, T. J. Learning overcomplete representations. *Neural computation*, 2000.

Li, Y., Cai, T., Zhang, Y., Chen, D., and Dey, D. What makes convolutional models great on long sequence modeling? *arXiv preprint arXiv:2210.09298*, 2022a.

Li, Z., Han, J., Weinan, E., and Li, Q. Approximation and optimization theory for linear continuous-time recurrent neural networks. *J. Mach. Learn. Res.*, 2022b.

Lorenz, E. N. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2):130–141, 1963.

Lotka, A. J. *Elements of physical biology*. Williams & Wilkins, 1925.

Lu, C., Schroecker, Y., Gu, A., Parisotto, E., Foerster, J., Singh, S., and Behbahani, F. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 2023.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

Ma, X., Zhou, C., Kong, X., He, J., Gui, L., Neubig, G., May, J., and Zettlemoyer, L. Mega: moving average equipped gated attention. *arXiv preprint arXiv:2209.10655*, 2022.

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

Mashreghi, J., Nazarov, F., and Havin, V. Beurling–malliavin multiplier theorem: the seventh proof. *St. Petersburg Mathematical Journal*, 17(5):699–744, 2006.

Müntz, C. H. *Über den approximationssatz von Weierstrass*. Springer, 1914.

Nguyen, E., Goel, K., Gu, A., Downs, G. W., Shah, P., Dao, T., Baccus, S. A., and Ré, C. S4nd: Modeling images and videos as multidimensional signals using state spaces. *Advances in Neural Information Processing Systems*, 2022.

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., and De, S. Resurrecting recurrent neural networks for long sequences. *International Conference on Machine Learning*, 2023.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

Pinkus, A. Approximation theory of the MLP model in neural networks. *Acta numerica*, 8:143–195, 1999.

Schäfer, A. M. and Zimmermann, H. G. Recurrent neural networks are universal approximators. In *Artificial Neural Networks–ICANN 2006: 16th International Conference, Athens, Greece, September 10-14, 2006. Proceedings, Part I 16*, pp. 632–640. Springer, 2006.

Siegelmann, H. T. and Sontag, E. D. On the computational power of neural nets. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pp. 440–449, New York, NY, USA, 1992a. Association for Computing Machinery.

Siegelmann, H. T. and Sontag, E. D. On the computational power of neural nets. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 440–449, 1992b.

Smith, J. T., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

Smith, J. T., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling. *International Conference on Learning Representations*, 2023.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Szász, O. Über die approximation stetiger funktionen durch lineare aggregate von potenzen. *Mathematische Annalen*, 77(4):482–496, 1916.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2020.

Tlas, T. Bump functions with monotone fourier transforms satisfying decay bounds. *Journal of Approximation Theory*, 278:105742, 2022.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Volterra, V. Variations and fluctuations of the number of individuals in animal species living together. *ICES Journal of Marine Science*, 3(1):3–51, 1928.

Vyshemirsky, V. and Girolami, M. A. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6): 833–839, 2008.

Wang, J., Yan, J. N., Gu, A., and Rush, A. M. Pretraining without attention, 2023.

Wang, S. and Xue, B. State-space models with layer-wise nonlinearity are universal approximators with exponential decaying memory. *arXiv preprint arXiv:2309.13414*, 2023.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

Zucchet, N., Kobayashi, S., Akram, Y., von Oswald, J., Larcher, M., Steger, A., and Sacramento, J. Gated recurrent neural networks discover attention. *arXiv preprint arXiv:2309.01775*, 2023a.

Zucchet, N., Meier, R., Schug, S., Mujika, A., and Sacramento, J. Online learning of long-range dependencies, 2023b.

Appendix

A. Related Works

Issues with attention for long-range reasoning. Efficient processing of long sequences is an important open question in deep learning. Attention-based transformers (Vaswani et al., 2017) provide a scalable approach but suffer from *quadratically increasing complexity in inference/memory* as the sequence length grows. While many approaches exist to alleviate this issue, e.g. efficient memory management (Dao et al., 2022; Dao, 2023) and architectural modifications (Wang et al., 2020; Kitaev et al., 2020; Child et al., 2019; Beltagy et al., 2020), the sequence length in large language models is usually kept to $2k/4k$ tokens for this reason (e.g. Llama2 (Touvron et al., 2023)). In addition, in some long-range reasoning tasks (Tay et al., 2020) attention does not seem to provide the correct *inductive bias*, leading to poor performance in addition to high computational costs.

Success of modern recurrent layers. Due to the issues outlined above, the community has witnessed in the last year the rise of new, drastically innovative, *recurrent* alternatives to the attention mechanism, named state-space models (SSMs). The first SSM, S4, was introduced by (Gu et al., 2021), and since then, a plethora of variants have been proposed: LiquidS4 (Hasani et al., 2022), DSS (Gupta et al., 2022), S4D (Gu et al., 2022), S5 (Smith et al., 2022), RWKV (Peng et al., 2023) and RetNet (Sun et al., 2023) among others. These models achieve remarkable performance, surpassing all modern attention-based transformer variants by an average 20% accuracy on challenging sequence classification tasks (Tay et al., 2020). SSMs have reached outstanding results in various domains beyond toy datasets (Nguyen et al., 2022; Goel et al., 2022; Gu et al., 2021; Lu et al., 2023; Zucchet et al., 2023b). SSMs also were successfully applied to language modeling, and are sometimes used in combination with attention (Fu et al., 2023; Wang et al., 2023; Ma et al., 2022). At inference time, all SSMs coincide with a stack of linear Recurrent Neural Networks, interleaved with MLPs and normalization. Linearity of the RNNs allows fast parallel processing with FFTs (Gu et al., 2022) or parallel scans (Smith et al., 2023).

The linear recurrent unit (LRU). Among modern architectures for long-range reasoning based on recurrent modules, the simplest is perhaps Linear Recurrent Unit (LRU) (Orvieto et al., 2023): while SSMs rely on the discretization of a structured continuous-time latent dynamical system, the LRU is directly designed for discrete-time systems (token sequences), and combines easy hyperparameter tuning with solid performance and scalability. The only differences between the LRU and the standard RNN update $x_k = Ax_{k-1} + Bu_k$ (u is the input at a specific layer and x is the hidden-state, then fed into a position-wise MLP) are (1) the system operates in the complex domain (required for expressivity, see discussion in (Orvieto et al., 2023)) (2) to enhance stability and better control how quickly gradients vanish, A (diagonal) is learned using polar parametrization and log-transformed magnitude and phase. Finally, (3) the recurrence is normalized through an extra optimizable parameter that scales the input to stabilize signal propagation. The parametrization of linear RNNs of (Orvieto et al., 2023) was found to be effective also in surpassing deep LSTMs and GRUs (Zucchet et al., 2023a). We use the LRU codebase[12] as a starting point for our experiments, when the linear RNN is learned.

Approximation theory for MLP and non-linear RNNs. The approximation properties of deep neural networks with ReLU activations are well studied. While recent advances concern the effect of depth (Lu et al., 2017), the study by Pinkus (1999), as well as previous works (Funahashi, 1989; Hornik et al., 1989; Hornik, 1991; Barron, 1993), already established the power of neural networks with a single hidden layer, which can approximate arbitrary continuous non-linear maps on compacts as the size of the hidden layer grows to infinity. The cleanest result is perhaps the one of Barron (1993), that we heavily use in this paper.

In the context of non-linear RNN approximation of dynamical systems (e.g. in neuroscience), the state-to-state computation can be seen as part of an MLP (see e.g. Hanson & Raginsky (2020)): we have $x_k = \sigma(Ax_{k-1} + Bu_k)$, where σ is is a non-linearity. As a result, wide non-linear RNNs can in principle approximate non-linear dyamical systems, as we discuss in detail in App. B.

Meanwhile, linear RNNs, where $x_k = Ax_{k-1} + Bu_k$, have often been considered of minor interest, as they equivalent in approximation power to convolutions (Li et al., 2022b) (see App. B). In this paper we take a different approach: we show that when sufficiently wide, the linear RNNs *do not* form a bottleneck, and the architecture maintains universality through the application of the pointwise MLP on the hidden state, as done in recent SSMs achieving state-of-the-art results (Gu et al., 2021; Smith et al., 2022; Orvieto et al., 2023). As motivated thoroughly in the paper, this architecture unlocks parallel computation, in contrast to what is possible with directly placing non-linearities in the recurrence.

[12] https://github.com/NicolasZucchet/minimal-LRU/tree/main/lru

B. Approximation theory for (non-linear) RNNs

We recall a result on universality of MLPs already stated in the main paper.

Definition 3 (Barron function)**.** *Let $\mathcal{F}(\omega)$ be the Fourier transform of $f : \mathbb{R}^n \to \mathbb{R}$. f belongs to the Barron class if $C_f = \int_{\mathbb{R}^n} \|\omega\|_2 |\mathcal{F}(\omega)| d\omega < \infty$.*

Theorem 1 (Universality of 1HL-MLPs)**.** *Consider $g(x)$ parametrized by a 1HL-MLP (with D hidden neurons): $g(x) = \sum_{k=1}^{D} \tilde{c}_k \sigma(\langle \tilde{a}_k, x\rangle + \tilde{b}_k) + \tilde{c}_0$, where σ is any sigmoidal function[13]. Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous with Barron constant C_f. If $D \geq 2r^2 C_f^2 \epsilon^{-2}$, then there exist parameters such that $\sup_{\|x\|\leq r} |f(x) - g(x)| \leq \epsilon$.*

B.1. Guarantees for RNNs with recurrent non-linearities

Research on universality of non-linear RNNs dates back to (Siegelmann & Sontag, 1992b). We present here a more recent result by Hanson & Raginsky (2020).

Theorem 5 (Universality of non-linear RNNs)**.** *Consider the continuous-time non-linear dynamical system with form*

$$\dot{\bar{x}}(t) = \bar{f}(\bar{x}(t), u(t)), \quad \bar{y}(t) = h(\bar{x}(t)), \tag{8}$$

with $\bar{x}(t) \in \mathbb{R}^{\bar{N}}$, $u(t) \in \mathbb{R}^M$. Under some technical assumptions (bounded input, non-chaotic f), for any $\epsilon > 0$ there exists a non-linear RNN

$$\dot{x}(t) = -\frac{1}{\tau} x(t) + \sigma(Ax(t) + Bu(t)), \quad y(t) = Cx(t), \tag{9}$$

for some non-polynomial σ, $\tau > 0$, $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times M}$, $C \in \mathbb{R}^{M\times N}$ that approximates the solution to Eq. 8 up to error ϵ uniformly in time, on compact sets of inputs.

The result above typically involves taking N (RNN width) to infinity.

Proof. We briefly outline the idea behind the proof, and invite the reader to refer to (Hanson & Raginsky, 2020) for details. Approximating the solution to Eq. 8 is equivalent to approximating the infinitesimal solution generator, which is a non-linear function of (x, u). By Barron's Theorem (Thm. 1), this generator can be approximated by a one-layer MLP, that is exactly the right-hand side of Eq. 9. □

B.2. Guarantees for linear RNNs

Simply taking out the non-linearity from the recurrence in Eq. 9 restricts the function class to convolutions. To start, recall that the linear continuous-time RNN on one-dimensional inputs

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t),$$

with $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times M}$, $C \in \mathbb{R}^{1\times N}$ has solutions given by a convolution.

$$x(t) = \int_0^t C^\top e^{As} Bu(t-s) ds =: \int_0^t \rho(s)^\top u(t-s) ds =: \sum_i (\rho^i \star u^i)_t.$$

Let us call $\hat{\mathcal{H}_N}$ the class of functionals parametrizable with linear RNNs with hidden state of dimension N, and $\hat{\mathcal{H}} = \cup_{N\in\mathbb{N}_+} \mathcal{H}_N$.

$$\mathcal{H}_N := \left\{ \{H_t : t \in \mathbb{R}\}, H_t(u) = \int_0^t C^\top e^{As} Bu(t-s) ds, C \in \mathbb{R}^{1\times N}, A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times M} \right\}.$$

It turns out that convolutions are dense in the class of linear functionals. Let $\mathcal{U} = C_0(\mathbb{R}, \mathbb{R}^d)$ with norm $\|u\| = \sup_{t\in\mathbb{R}} \|u(t)\|_\infty$.

Theorem 6 (Linear functionals in $C_0(\mathbb{R}, \mathbb{R}^d)$ are convolutions (Li et al., 2022b))**.** *Let $\{H_t : t \in \mathbb{R}\}$ be a family of continuous, linear, causal, regular, and time-homogeneous functionals on $\mathcal{U}$, i.e. such that*

[13] $\lim_{x\to-\infty} \sigma(x) = 0$ and $\lim_{x\to\infty} \sigma(x) = 1$.

1. *(Continuous)* $\forall t \in \mathbb{R}$, $\sup_{\|u\|<1} H_t(u) < \infty$.
2. *(Linear)* $\forall t \in \mathbb{R}$, $u, v \in \mathcal{U}$ *and* $\nu, \lambda \in \mathbb{R}$, *we have* $H_t(\lambda u + \nu v) = \lambda H_t(u) + \nu H_t(v)$.
3. *(Causal) For all* $u, v \in \mathcal{U}$ *such that* $u(s) = v(s)$ *for all* $s \leq t$, *he have* $H_t(v) = H_t(u)$.
4. *(Regular) Let* (u^n) *be a sequence in* $\mathcal{U}$ *s.t.* $u^n(s) \to 0$ *for almost every* $s \in \mathbb{R}$, *then, for all* $t \in \mathbb{R}$, *we have* $\lim_{n\to\infty} H_t(u^n) = 0$.
5. *(Time Homogeneous) For all* $u \in \mathcal{U}$ *let* $u_t^\tau = u(t-\tau)$, *then* $H_t(u^\tau) = H_{t+\tau}(u)$.

Then, for any $\{H_t : t \in \mathbb{R}\}$ *there exist a function (a kernel)* $\bar{\rho} : \mathbb{R}_+ \to \mathbb{R}^M$ *such that for all* $t \in \mathbb{R}$.

$$H_t(u) = \int_0^\infty \rho(s)^\top u(t-s)ds = \sum_i (\bar{\rho}^i \star u^i)_t.$$

Theorem 7 (Linear RNNs can parametrize any convolution (Li et al., 2022b))**.** *Let* $\{H_t : t \in \mathbb{R}\}$ *be a family of continuous, linear, causal, regular, and time-homogeneous functionals on* $\mathcal{U}$. *Then, for any* $\epsilon > 0$ *there exists* $\{\hat{H}_t : t \in \mathbb{R}\} \in \hat{\mathcal{H}}$ *such that*

$$\sup_{t\in\mathbb{R}} \sup_{\|u\|<1} \|H_t(u) - \hat{H}_t(u)\| \leq \epsilon.$$

The result above typically involves taking N (RNN width) to infinity.

This paper. There is a sizable gap between the power of nonlinear and linear RNNs. We show in this paper that placing a nonlinearity at the output of linear RNNs (unlocking parallel computation) allows approximation of arbitrary regular non-linear sequence-to-sequence mappings.

C. Details on Multidimensional Input Reconstruction

In the main text, we showed that linear diagonal RNN computations on one-dimensional input sequences can be written in matrix form using a Vandermonde matrix (Sec. 3.1). For convenience of the reader, we repeat the reasoning here: let $H = M = 1$, the encoder e be the identity, and $B = (1, 1, \dots, 1)^\top$. Then, eq. (3) can be written as

$$x_k = \begin{pmatrix} \lambda_1^{k-1} & \lambda_1^{k-2} & \cdots & \lambda_1 & 1 \\ \lambda_2^{k-1} & \lambda_2^{k-2} & \cdots & \lambda_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_N^{k-1} & \lambda_N^{k-2} & \cdots & \lambda_N & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{pmatrix} = V_k u_{1:k}^\top.$$

where $u_{1:k} = v_{1:k} = (v_i)_{i=1}^k \in \mathbb{R}^{1\times k}$, and V_k is a Vandermonde matrix. As long as $N \geq k$, we can hope to recover $u_{1:k}$ by pseudoinversion of the Vandermonde:

$$v_{1:k}^\top = u_{1:k}^\top = V_k^+ x_k,$$

Here, we give details on the design of input projections such that the RNN output from multidimensional inputs can also be seen as matrix multiplication. Let us define

$$\mathrm{vec}(v_{1:k}) := \begin{pmatrix} v_{1,1:k}^\top \\ v_{2,1:k}^\top \\ \vdots \\ v_{M,1:k}^\top \end{pmatrix} \in \mathbb{R}^{kM},$$

The matrix B we are going to use in our linear diagonal RNN is

$$B = \begin{pmatrix} 1_{N'\times 1} & \cdots & 0_{N'\times 1} & 0_{N'\times 1} \\ 0_{N'\times 1} & \cdots & 0_{N'\times 1} & 0_{N'\times 1} \\ \vdots & \ddots & \vdots & \vdots \\ 0_{N'\times 1} & \cdots & 1_{N'\times 1} & 0_{N'\times 1} \\ 0_{N'\times 1} & \cdots & 0_{N'\times 1} & 1_{N'\times 1} \end{pmatrix},$$

where we select $N = MN'$. With this choice, the linear diagonal RNN output can be written as

$$x_k = \left(\begin{array}{c|c|c|c} V_{k,1} & & & \\ \hline & V_{k,2} & & \\ \hline & & \ddots & \\ \hline & & & V_{k,M} \end{array}\right) \begin{pmatrix} v_{1,1:k}^\top \\ v_{2,1:k}^\top \\ \vdots \\ v_{M,1:k}^\top \end{pmatrix} = \tilde{V}_k \begin{pmatrix} 1 \\ \mathrm{vec}(v_{:,1:k}) \end{pmatrix},$$

where $V_k = \mathrm{diag}(V_{k,1}, V_{k,2}, \cdots, V_{k,M})$, and $V_{k,j} \in \mathbb{C}^{N'\times k}$ is the Vandermonde matrix corresponding to the block Λ_j in the diagonal recurrent matrix Λ:

$$\Lambda = \mathrm{diag}(\Lambda_1, \Lambda_2, \cdots, \Lambda_M) \in \mathbb{C}^{(N'M)\times(N'M)},$$
$$\Lambda_j = \mathrm{diag}(\lambda_{1,j}, \lambda_{2,j}, \cdots, \lambda_{N',j}) \in \mathbb{C}^{N'\times N'},$$
$$V_{k,j} = \begin{pmatrix} \lambda_{1,j}^{k-1} & \lambda_{1,j}^{k-2} & \cdots & \lambda_{1,j} & 1 \\ \lambda_{2,j}^{k-1} & \lambda_{2,j}^{k-2} & \cdots & \lambda_{2,j} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_{N',j}^{k-1} & \lambda_{N',j}^{k-2} & \cdots & \lambda_{N',j} & 1 \end{pmatrix} \in \mathbb{C}^{N'\times k}.$$

This construction effectively decouples each input dimension and reduces the discussion to the one-dimensional setting: invertibility of V_k is guaranteed by invertibility of each block, provided $N' \geq L$ and that eigenvalues are distinct. Slight changes can be made to keep also track of the timestamp (see Sec. 3.1.1) and to adapt to the sparse setting (see Sec. 3.1.2).

D. Expressivity Proofs

One of our main steps involved computation of the Barron constant of the function mapping the RNN hidden state to the output.

Proposition 2 (MLP, single timestamp)**.** *The number of hidden neurons in the MLP g sufficient to approximate $f_k := T_k \circ \Omega_k$ at level ϵ from the RNN hidden state x_k is bounded by $4r^2 C_{T_k}^2 \|\Omega_k\|_2^2 S^3 \epsilon^{-2}$ where r bounds the hidden state magnitude, and C_{T_k} is the Barron constant of T_k.*

Since Ω_k is a matrix, the result can be proved by computing the Barron constant of a function where the argument is the output of a linear map.

[Change of variables] Let $A \in \mathbb{R}^{p \times n}$ and $f(x) = g(Ax)$, then

$$C_f = \|A\|_2 C_g.$$

Proof. The inverse Fourier transform formula directly leads to

$$f(x) = \int_{\mathbb{R}^p} e^{i\langle p, Ax\rangle} \mathcal{G}(\xi) d\xi. \tag{10}$$

Let us now compute the Fourier Transform of f.

$$\begin{aligned}
\mathcal{F}(\omega) &= \int_{\mathbb{R}^n} e^{-i\langle \omega, x\rangle} f(x) dx && (11)\\
&= \int_{\mathbb{R}^n} e^{-i\langle \omega, x\rangle} \left[\int_{\mathbb{R}^p} e^{i\langle \xi, Ax\rangle} \mathcal{G}(\xi) d\xi \right] dx && (12)\\
&= \int_{\mathbb{R}^n} \int_{\mathbb{R}^p} e^{-i\langle \omega, x\rangle} e^{i\langle A^\top \xi, x\rangle} \mathcal{G}(\xi) d\xi dx && (13)\\
&= \int_{\mathbb{R}^p} \left[\int_{\mathbb{R}^n} e^{i\langle A^\top \xi - \omega, x\rangle} dx \right] \mathcal{G}(\xi) d\xi. && (14)
\end{aligned}$$

Recall now the definition of the Dirac delta:

$$\delta(z) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{i\nu z} d\nu. \tag{15}$$

Therefore

$$\mathcal{F}(\omega) = \int_{\mathbb{R}^p} \delta(A^\top \xi - \omega) \mathcal{G}(\xi) d\xi. \tag{16}$$

Note that this is a singular measure in $\mathbb{R}^n$: lives in a linear p-dimensional subspace. Further, note that

$$\begin{aligned}
C_f &= \int_{\mathbb{R}^n} \|\omega\|_2 \cdot |\mathcal{F}(\omega)| d\omega && (17)\\
&= \int_{\mathbb{R}^n} \|\omega\|_2 \cdot \Big| \int_{\mathbb{R}^p} \delta(A^\top \xi - \omega) \mathcal{G}(\xi) d\xi \Big| d\omega && (18)\\
&\leq \int_{\mathbb{R}^n} \|\omega\|_2 \cdot \int_{\mathbb{R}^p} \delta(A^\top \xi - \omega) |\mathcal{G}(\xi)| d\xi d\omega && (19)\\
&\leq \int_{\mathbb{R}^p} \left[\int_{\mathbb{R}^n} \|\omega\|_2 \delta(A^\top \xi - \omega) d\omega \right] |\mathcal{G}(\xi)| d\xi && (20)\\
&= \int_{\mathbb{R}^p} \|A^\top \xi\|_2 \cdot |\mathcal{G}(\xi)| d\xi && (21)\\
&= \|A^\top\|_2 \int_{\mathbb{R}^p} \|\xi\|_2 \cdot |\mathcal{G}(\xi)| d\xi. && (22)
\end{aligned}$$

The proof is done, since $\|A^\top\|_2 = \|A\|$ (same singular values), and $C_g = \int_{\mathbb{R}^p} \|\xi\|_2 \cdot |\mathcal{G}(\xi)| d\xi$. □

We now proceed proving the complexity for interpolation of sequences of Barron functions. This directly implies our main result (Thm.2).

Theorem 4 (Combination of Barron Maps)**.** *Let $f : [1, 2, \ldots, L] \times \mathbb{R}^n \to \mathbb{R}$ be such that each $f(k, \cdot)$ is Barron with constant C_{f_k} and Fourier norm C'_{f_k}. There exists an extension $\tilde{f} : \mathbb{R}^{n+1} \to \mathbb{R}$ of f, s.t. $\tilde{f}$ is Barron with constant $\tilde{C} \leq \mathcal{O}(\sum_{k=1}^{L} C_{f_k} + C'_{f_k}) = \mathcal{O}(L)$ and $\tilde{f}(x, k) = f(x, k), \forall x \in \mathbb{R}^n$ and $k \in 1, 2, \ldots, L$.*

Proof. Let us consider the following definition for $\tilde{f}$:

$$\tilde{f}(x, t) = \sum_{k=1}^{L} f(x, k)h(t - k), \tag{23}$$

where $h : \mathbb{R} \to \mathbb{R}$ is a filter (see discussion after the proof) with support in $[-1, 1]$. Compactness in the support of h leads to the desired property $\tilde{f}(x, k) = f(x, t)$, for all $k = 1, 2, \ldots L$. Let us now compute the Fourier transform of $\tilde{f}$. Frequencies are of the form $\omega = (w, \nu)$, with $w \in \mathbb{R}^n$, $\nu \in \mathbb{R}$.

$$\begin{aligned}
\tilde{\mathcal{F}}(\omega) &= \frac{1}{(2\pi)^{n+1}} \int_{\mathbb{R}} \int_{\mathbb{R}^n} \tilde{f}(x, t) e^{-i\langle w, x\rangle} e^{-i\nu t} dx dt && (24)\\
&= \frac{1}{(2\pi)^{n+1}} \int_{\mathbb{R}} \int_{\mathbb{R}^n} \left[\sum_{k=1}^{L} f(x, k)h(t - k)\right] e^{-i\langle w, x\rangle} e^{-i\nu t} dx dt && (25)\\
&= \frac{1}{(2\pi)^{n+1}} \sum_{k=1}^{L} \left[\int_{\mathbb{R}^n} f(x, k) e^{-i\langle w, x\rangle} dx\right] \cdot \left[\int_{\mathbb{R}} h(t - k) e^{-i\nu t} dt\right] && (26)\\
&= \sum_{k=1}^{L} \tilde{\mathcal{F}}_k(w)\mathcal{H}(\nu)e^{i\nu k}, && (27)
\end{aligned}$$

where $\mathcal{H}$ is the Fourier transform of h, and the factor $e^{i\nu k}$ comes from the shift $h(\cdot - k)$. All in all, we get

$$\tilde{\mathcal{F}}(w, \nu) = \mathcal{H}(\nu) \sum_{k=1}^{L} \tilde{\mathcal{F}}_k(w)e^{i\nu k}. \tag{28}$$

Trivially,

$$|\tilde{\mathcal{F}}(w, \nu)| \leq |\mathcal{H}(\nu)| \sum_{k=1}^{L} |\tilde{\mathcal{F}}_k(w)|. \tag{29}$$

Therefore:

$$\begin{aligned}
\tilde{C} &= \int_{\mathbb{R}^{n+1}} \|\omega\|_2 |\tilde{\mathcal{F}}(\omega)| d\omega && (30)\\
&\leq \int_{\mathbb{R}^n} \int_{\mathbb{R}} \|(w, \nu)\|_2 \cdot \left[|\mathcal{H}(\nu)| \sum_{k=1}^{L} |\tilde{\mathcal{F}}_k(w)|\right] dw d\nu && (31)\\
&= \sum_{k=1}^{L} \int_{\mathbb{R}^n} \int_{\mathbb{R}} \|(w, \nu)\|_2 \cdot |\mathcal{H}(\nu)| \cdot |\tilde{\mathcal{F}}_k(w)| dw d\nu. && (32)
\end{aligned}$$

Using the triangle inequality:

$$\begin{aligned}
\tilde{C} &\leq \int_{\mathbb{R}^n} \int_{\mathbb{R}} (\|w\|_2 + |\nu|) \cdot |\mathcal{H}(\nu)| \cdot |\tilde{\mathcal{F}}_k(w)| dw d\nu && (33)\\
&= \int_{\mathbb{R}^n} \int_{\mathbb{R}} \|w\|_2 \cdot |\mathcal{H}(\nu)| \cdot |\tilde{\mathcal{F}}_k(w)| dw d\nu + \int_{\mathbb{R}^n} \int_{\mathbb{R}} |v| \cdot |\mathcal{H}(\nu)| \cdot |\tilde{\mathcal{F}}_k(w)| dw d\nu && (34)\\
&= C_{f_k} C'_h + C_h C'_{f_k}. && (35)
\end{aligned}$$

This concludes the proof since (see next paragraph) C_h and C'_h are problem-independent bounded constants. □

The proof of the theorem above concludes stating that C_h and C'_h are problem-independent bounded constants. Recall that, in our proof, $h : \mathbb{R} \to \mathbb{R}$ has compact support in $[-1, 1]$. We can design h such that its Fourier transform has fast decay:

Theorem 8 ((Tlas, 2022)). *For any $\delta \in (0,1)$ and any $c > 0$ there is a function $h(t)$ which is C^∞, real, even, nonnegative, supported in $[-1,1]$ and whose Fourier transform $\mathcal{H}(\nu)$ is monotone decreasing for $\nu \geq 0$ and satisfies the following double inequality*

$$\exp(-(1+\epsilon)c\nu^\delta) \lesssim \mathcal{H}(\nu) \lesssim \exp(-(1-\epsilon)c\nu^\delta), \tag{36}$$

for any $\epsilon > 0$.

This result, rooted in the Beurling-Malliavin multiplier theorem(Mashreghi et al., 2006), ensures that we can design h on a compact support with exponentially decaying frequencies. This implies that both integrals

$$C_h = \int_{\mathbb{R}} |\nu||\mathcal{H}(\nu)|d\nu, \qquad C'_h = \int_{\mathbb{R}} |\mathcal{H}(\nu)|d\nu \tag{37}$$

are bounded and independent on the specific form of the function f we want to approximate.

E. Additional experiments

E.1. Reconstruction using the Vandermonde inverse (support to Section 3.1)

We consider linear diagonal RNNs with the N diagonal entries of Λ sampled inside the unit disk in $\mathbb{C}$, uniformly in angle in between radii $r_{\min}$ and 1. We consider the hidden state $x_L \in \mathbb{C}^N$ computed after L RNN steps, *i.e. after the sequence is completely processed*. We want to recover the input sequence from the hidden state using the Vandermonde inverse V_L^+ (see Sec. 3.1). If $N \geq L$, since under random initialization the determinant of any set of L columns of V_L is positive, we can in theory achieve perfect reconstruction. In practice, V_L is ill conditioned — especially if $r_{\min}$ is not close to 1. This causes some problem in the pseudoinverse computation, which may result in imperfect reconstruction.

In Figure 6 we provide evidence on the MNIST dataset (flattened image = 784 tokens): we observe that, if eigenvalues are sampled with $r_{\min} = 0$, by multiplying x_L by V_L^+ we are only able to recover recent history. Instead, moving closer to the unit disk allows almost perfect reconstruction at $N = 784$, and a satisfactory result already at $N = 512$.

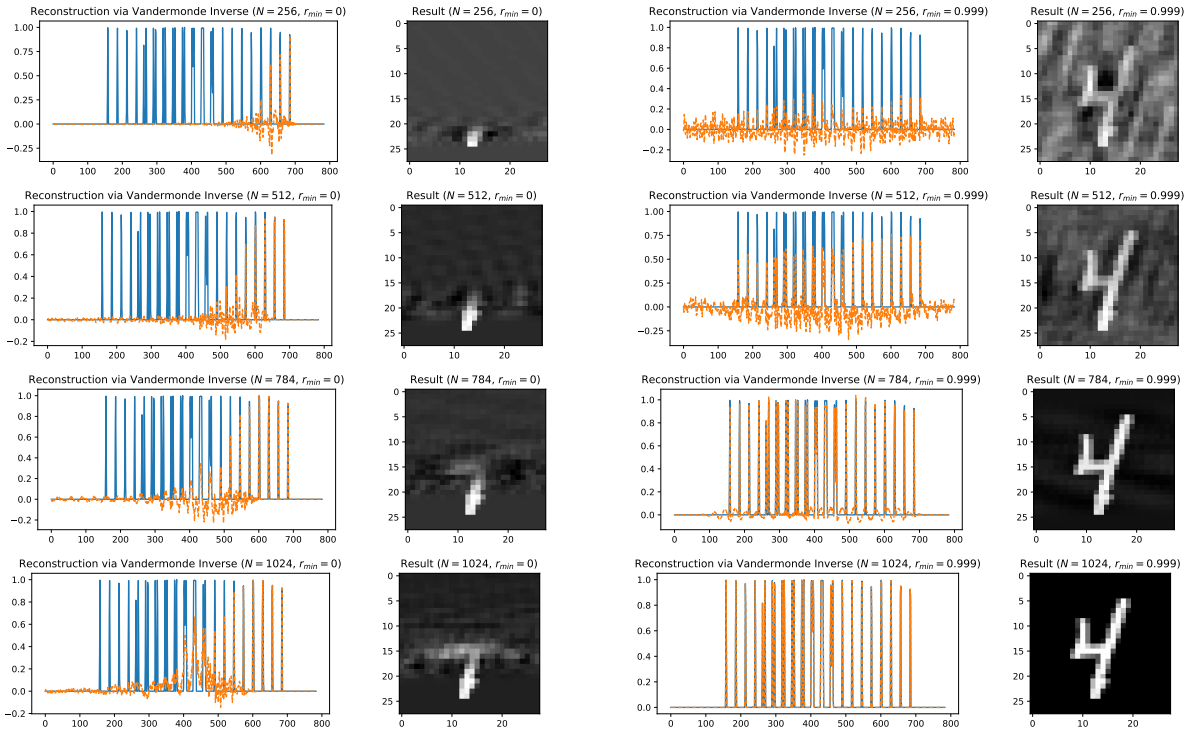

Figure 6. Reconstrucion of MNIST digits (seen as a sequence) from the last linear RNN state. **Reconstruction map is not learned** but is instead based on the Vandermonde Pseudoinverse (i.e. worst-case setting, as discussed in the main text). Results are far more accurate if the reconstruction map is learned (see Fig. 4).

In Figure 7 we clearly show that the average reconstruction error (average over $10k$ images samples and 10 random re-samplings of the linear RNN) is decreasing both as a function of the hidden state size (see discussion in Sec. 3.1) and of $r_{\min}$ ($r_{\max} = 1$). The same pattern is observed for the condition number of $V_L^\top V_L$. On the same figure, we show how the error is distributed over timestamps: it is clear that, for $r_{\min} \ll 1$, the reconstruction only covers the last few hundreds of tokens – a property which is liked to the bad condition number observed in this setting.

Last, in Figure 8 we show what happens when picking the N diagonal entries of Λ to be the N-th complex roots of 1: as shown in (Córdova et al., 1990), in this setting the Vandermonde condition number is 1. We observe that we can indeed reconstruct perfectly the output for $N = 784$. However, for smaller values of N, the reconstruction presents undesired artifacts.

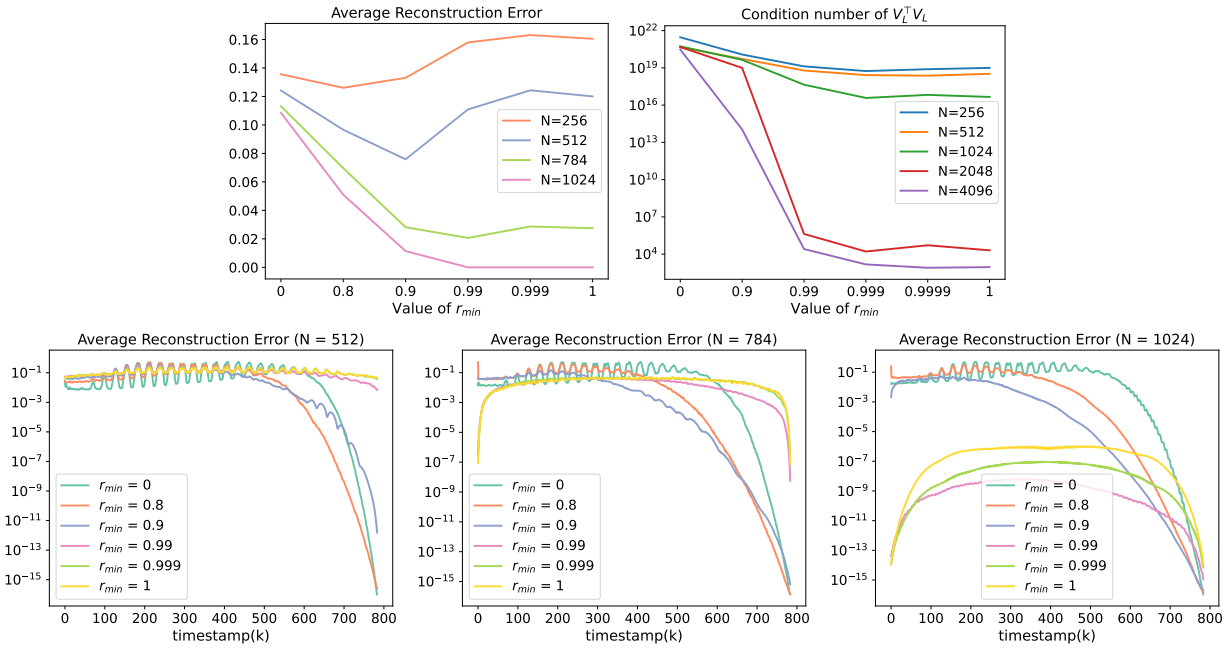

Figure 7. Error over $10k$ MNIST images and 10 re-sampling of the RNN. Comment in text. Reconstruction is based on the Vandermonde Pseudoinverse.

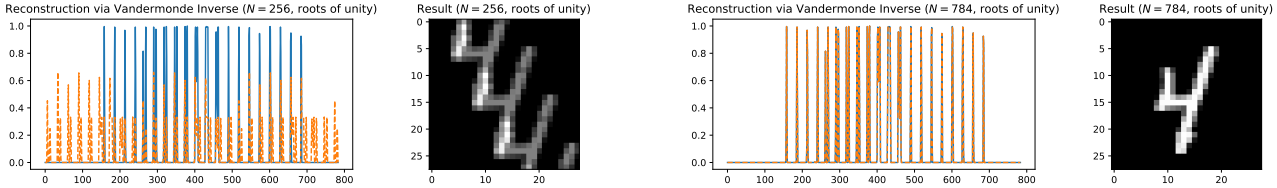

Figure 8. Same setting as Figure 6. Eigenvalues are the N-th complex roots of 1. Comment in text. **Reconstruction map is not learned** but is instead based on the Vandermonde Pseudoinverse (i.e. worst-case setting, as discussed in the main text).

E.2. Reconstruction under sparsity (support to §3.1.2)

In Figures 10 and 11 we test the discussion in Sec. 3.1.2 in a controlled setting. We consider one-dimensional stream of $L = 4096$ random inputs sparse in a basis of $P = 32$ Haar wavelets (Haar, 1911). The linear diagonal RNN has $\Lambda \in \mathbb{C}^{N\times N}$ with eigenvalues sampled uniformly at random from $\mathbb{T}(0.95, 1)$ (Fig. 10) or $\mathbb{T}(0.99, 1)$ (Fig. 11), where

$$\mathbb{T}[r_{\min}, r_{\max}] := \{\lambda \in \mathbb{C} \mid r_{\min} \leq |\lambda| \leq r_{\max}\}.$$

We use matrix $B = (1, 1, \ldots, 1)^\top$. Plotted is the rank of V_k, Ψ_k, $\Omega_k = V_k\Psi_k$ as k increases (see notation in Sec. 3.1.2). We show how the reconstruction error behaves when reconstructing $u_{1:k} = \Psi_k\alpha_k^u$ with $\alpha_k^u = \Omega_k^+ x_k$. In the figures, we plot the error for reconstruction of the tokens $(u_i)_{i=1}^k$ from x_k, for all $k \leq L$. As N gets larger the reconstruction error gets uniformly negligible. In particular, if we initialize in $\mathbb{T}(0.95, 1)$ then the minimum N we need for perfect reconstruction of around $N = 256$. If instead we initialize closer to the unit circle, then $N = 64$ is enough for perfect reconstruction. This finding is similar to the one presented in Sec. E.1.

On a more fundamental level, we study the condition number of the matrix $\Omega^T\Omega$, where $\Omega = V_L\Psi_L$. This condition number quantifies the numerical stability of the pseudoinverse Ω^+, used in reconstruction. In Figure 9, we show the logarithm of the condition number for a sequence of length 512 sparse in a basis of P eigenfunctions. As $r_{\min}$ gets close to 1 the condition number decreases as we saw in §E.1. Crucially however, the condition number also decreases as P decreases.

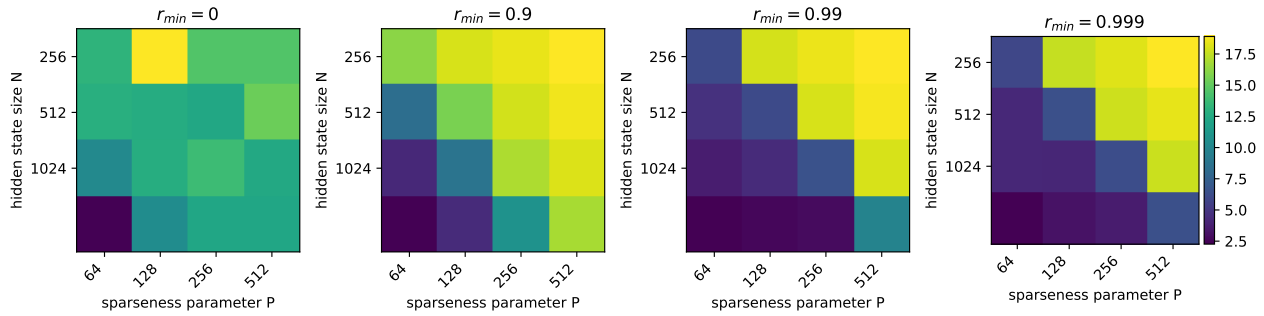

Figure 9. Logarithm of the condition number for $\Omega^T\Omega$ decreases as N (hidden state dimension) increases and as P (number of basis functions) decreases. As always, $r_{\min}$ closer to 1 leads to better conditioning and therefore more stable reconstruction.

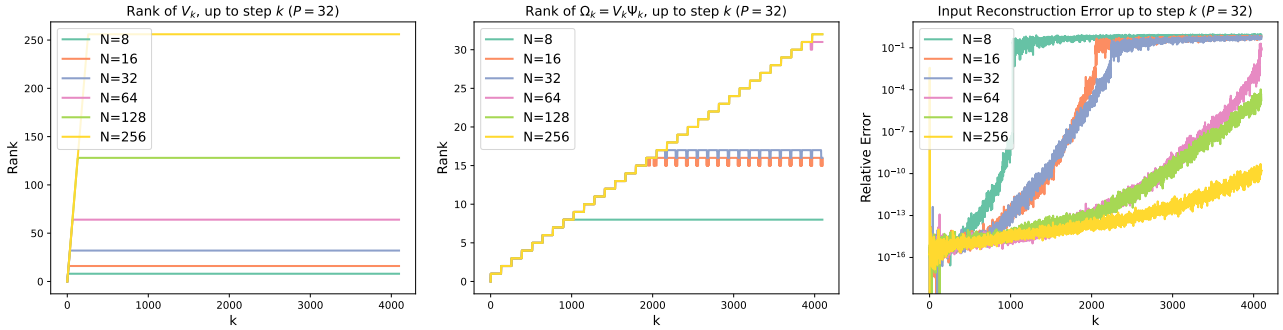

Figure 10. Reconstruction of a random sparse input. Λ initialized uniformly at random on $\mathbb{T}(0.95, 1)$. Comment in the text.

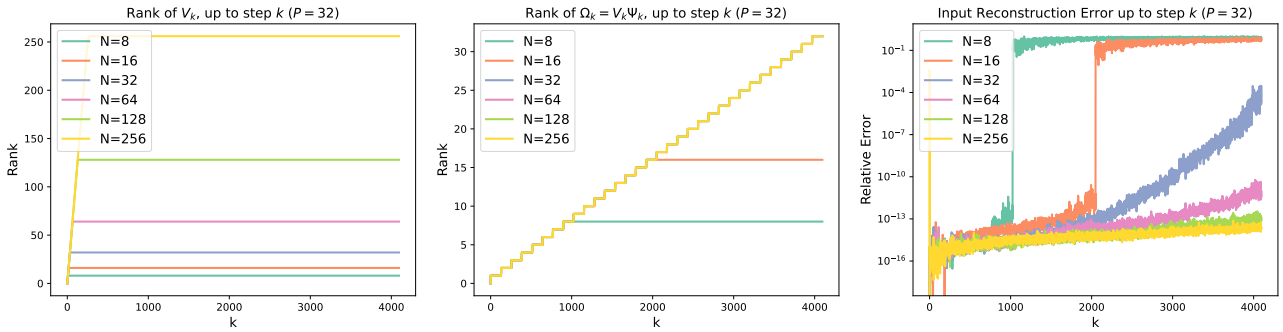

Figure 11. Reconstruction of a random sparse input. Λ initialized uniformly at random on $\mathbb{T}(0.99, 1)$. Comment in the text.

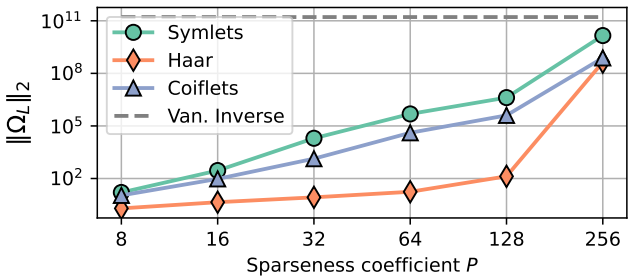

Figure 12. Conditioning of the reconstruction map for inputs sparse in 3 different wavelet basis ($N = 256$). Average over $1k$ random init., $r_{\min} = 0.999, r_{\max} = 1$. Comment in the main text.

E.3. Approximation of sequence-to-sequence maps (ODE Systems)

We consider approximating sequence-to-sequence maps $(v_i)_{i=1}^L \overset{T}{\mapsto} (y_i)_{i=1}^L$ defined by Runge-Kutta discretization of the flow of a controlled differential equation $\dot{z}_t = f(z_t, v_t), y_t = h(z_t)$, where $(v_t)_t$ is the input, f is a non-linear multidimensional function, h projects the multidimensional state z_t into a one-dimensional output. An example is the **Protein Transduction (PT)** system (Vyshemirsky & Girolami, 2008):

$$
\begin{aligned}
\dot{z}_1(t) &= -k_1 z_1(t) - k_2 z_1(t) z_3(t) + k_3 z_4(t) + v(t) \\
\dot{z}_2(t) &= k_1 z_1(t) \\
\dot{z}_3(t) &= -k_2 z_1(t) z_3(t) + k_3 z_4(t) + V \frac{z_5(t)}{K_m + z_5(t)} \\
\dot{z}_4(t) &= k_2 z_1(t) z_3(t) - (k_3 + k_4) z_4(t) \\
\dot{z}_5(t) &= k_4 z_4(t) - V \frac{z_5(t)}{K_m + z_5(t)}
\end{aligned}
$$

We identify $y_k = z_1(\Delta k)$, where $\Delta = 0.01$, and $v_k = v(\Delta k)$. We sample $(v_i)_{i=1}^L$ ($L = 2048$ in PT) from a linear combination of 16 base wavelets of low frequency (*bias: input is random but has slow variations*), and set the ground truth $(y_i)_{i=1}^L$ to be the result of Runge-Kutta integration with stepsize Δ. As hyperparameters, we use $k_1 = 0.07, k_2 = 0.6, k_3 = 0.05, k_4 = 0.3, V = 0.017, K_m = 0.3, z_1(0) = 1, z_2(0) = 0, z_3(0) = 1, z_4(0) = z_5(0) = 0$ as prescribed by Vyshemirsky & Girolami (2008). Results using approximation of one linear RNN followed by a 1HL-MLP (shared across timestamps) are shown in Fig. 5 and discussed in the main text. To train, we use $10k$ random (low frequency) trajectories and test on $1k$ trajectories with same distribution. Experiments run on a single A5000 using an LRU in JAX (`https://github.com/NicolasZucchet/minimal-LRU/tree/main/lru`).

In this appendix, we additionally discuss performance in approximating the solution of two other controlled ODEs. Settings are same as used for PT unless stated otherwise. The first ODE (results in Fig. 15) is a **Lotka-Volterra (LV)** system (Lotka, 1925; Volterra, 1928):

$$
\begin{aligned}
\dot{z}_1(t) &= z_1(t)(a - b \cdot z_2(t)) \\
\dot{z}_2(t) &= -z_2(t)(c - d \cdot z_1(t)) + v(t)
\end{aligned}
$$

where $a = 1.0, b = 0.6, c = 1.0, d = 0.7$, and we initialize $z_1(0) = 1.0, z_2(0) = 0.5$ as used in Dondelinger et al. (2013). For integration, we use a stepsize of $\Delta = 0.01$ and $y_k = z_1(\Delta k)$. Here, sequence length is again $L = 2048$.

Finally, we consider an extremely *challenging scenario*: the **Lorentz system (LZ)** (Lorenz, 1963), which notoriously has chaotic solutions (named strange attractor, linked to the butterfly effect):

$$
\begin{aligned}
\dot{z}_1(t) &= \sigma \cdot (z_2(t) - z_1(t)) \\
\dot{z}_2(t) &= (r - z_3(t)) z_1(t) - z_2(t) + v(t) \\
\dot{z}_3(t) &= z_2(t) z_1(t) - b \cdot z_3(t)
\end{aligned}
$$

With our parameter choices $\sigma = 10, r = 26, b = 8/3$ and initialization $z_1(0) = -0.89229143, z_2(0) = 1.08417925, z_3(0) = 2.34322702$ (parameter source: Wikipedia, visited September 2023), the system has a chaotic

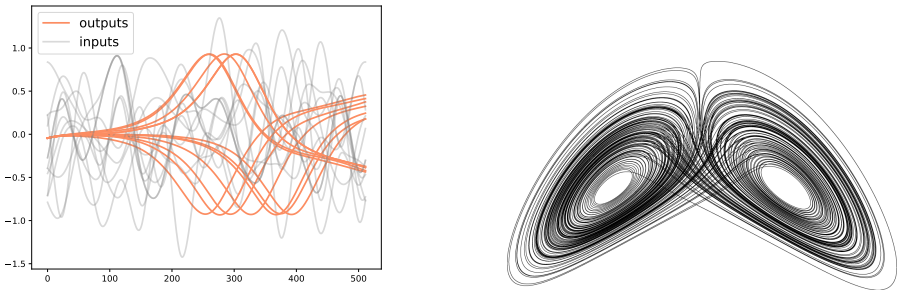

Figure 13. Behavior of the input-output LV map for random low-frequency inputs controlling the second state equation (output is the discretized first component). The output has binary nature (up or down) with shifted phase: tiny variations in the input cause drastic effects on the output (butterfly effect). Plotted is also an illustration of the 3-dimensional dynamics, source: https://www.iaacblog.com/programs/algorithmic-emergence-chaotic-attractor-equations/.

behavior as shown in Figure 13. We choose an integration timestep $\Delta = 0.002$ and, due to the non-linear chaotic and possibly unstable nature of the controlled attractor, consider $L = 512$ in this setting.

Note on training. Training one layer of linear RNN + MLP on these ODEs exhibits huge variation across seeds (see Fig. 14). Since in this paper we want to show that *there exist* a Linear RNN+MLP configuration able to model non-linear sequence to sequence maps, we consider the following setup: we run each experiment and hyperparameter sweep on seeds $1-6$, and only report the best performance on the test data (train loss always lower). Based on our experience with the LRU, we conclude that instability is due to the small dimension of our model: performance on standard tasks is more stable as depth increases (Orvieto et al., 2023). To test our theory, we limit ourselves to a linear RNN with either $N = 128$ or $N = 256$, followed by an MLP with one hidden layer. We grid-search hyperparameters for each model configuration and report test error for the best-performing models. Usual best-performing stepsizes are 0.003 and 0.01. In Lotka-Volterra experiments (Fig. 15) we train for 200 epochs, while we train on the Lorentz System (Fig. 16) for 1000 epochs. No weight decay, dropout or normalizations are applied. **Results are discussed in the relative figure captions and nicely validate our claims**. *Code for reproducing the experiments will be provided upon acceptance of this paper*.

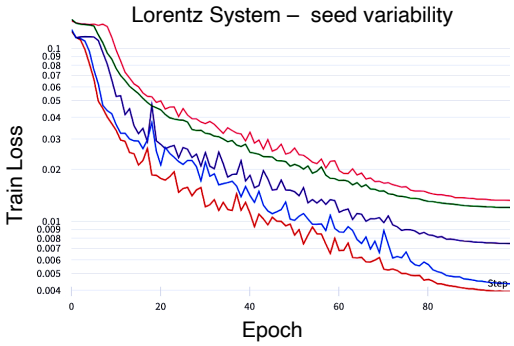

Figure 14. Variability in training loss dynamics when dealing with randomly initialized linear RNN + 1HL-MLP. Plotted are the dynamics for seeds $1-6$. All other hyperparameters are shared across runs.

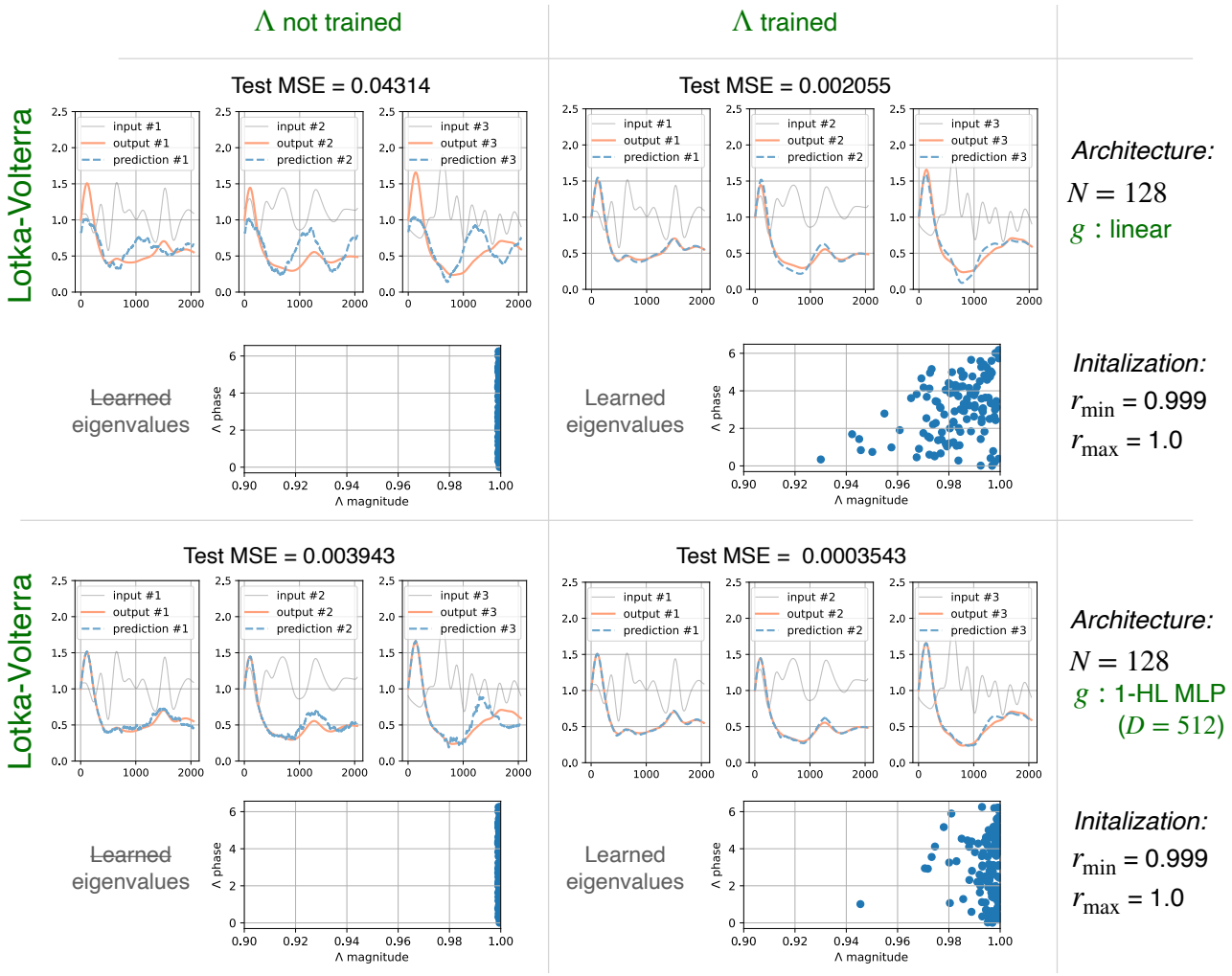

Figure 15. Performance of one layer linear RNN followed by a linear layer or a 1HL-MLP (shared accross timestamps), when training or clamping the RNN at initialization. The encoder and the readout from the RNN hidden state are always trained. **Results:** The model greatly benefits from learning the recurrent eigenvalues, but works already quite well with linear projections on the hidden state – indicating the LV system can be approximated (with some noticeable yet small error) by a linear functional, i.e. a linear system (see Thm 7). When used, the 1HL-MLP has $D = 512$ hidden neurons, to double the input size of $2N$ (real + imaginary part). Even though some elements of our theory suggest $r_{\min} \simeq 1$ for reconstruction guarantees, here we clearly see that successful learning pushes eigenvalues slightly inside the unit disk. As (Gu et al., 2021; Smith et al., 2022; Orvieto et al., 2023) we found it benefitial to initialize eigenvalues close to the unit circle.

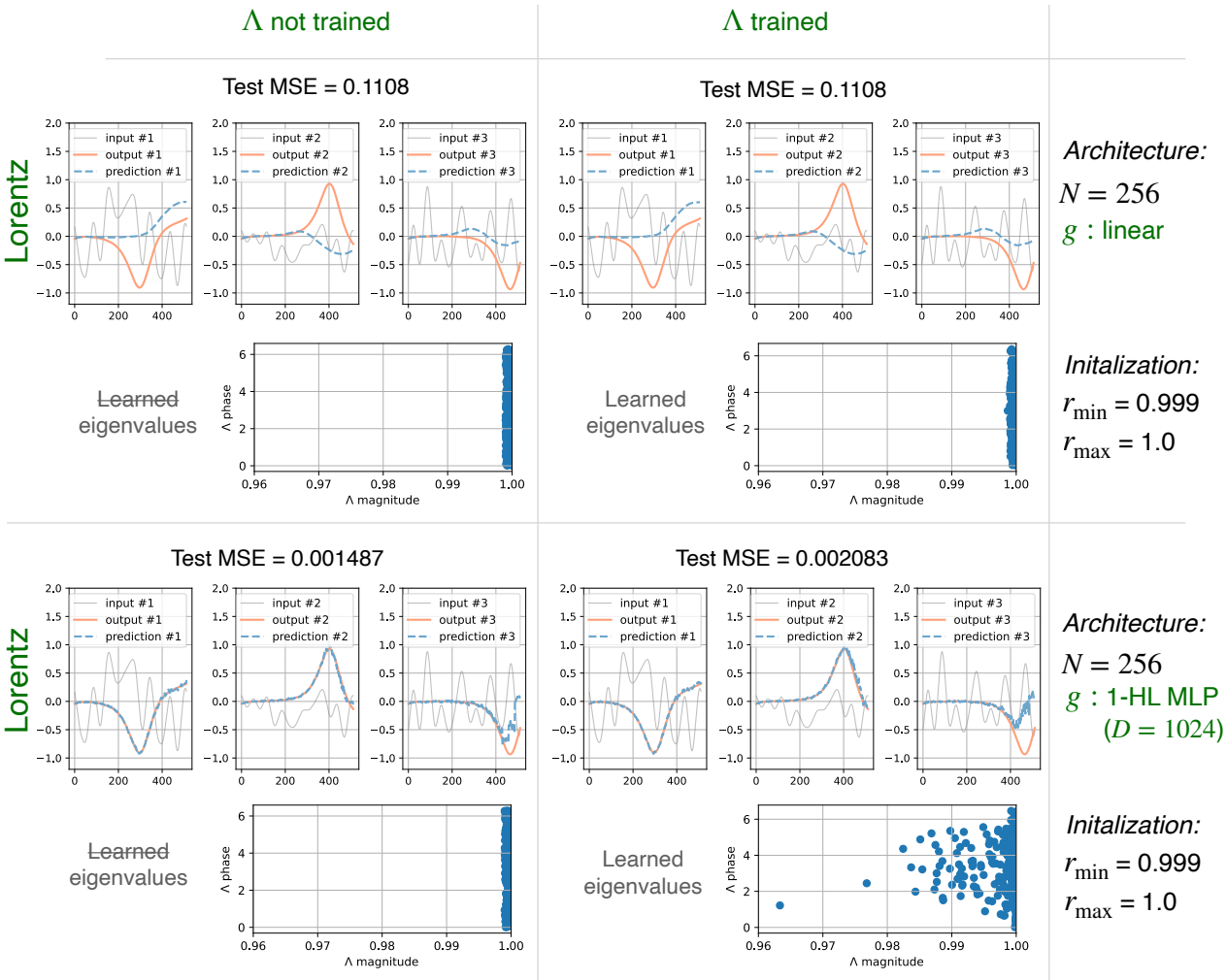

Figure 16. **Results:** compared to Fig. 15, here the task is considerably more challenging: no linear functional is able to learn useful information about the system — the 1HL-MLP is necessary for decent approximation as the system is highly non-linear. Learning input-output maps with $L = 512$ is already challenging: a larger (compared to LV) hidden state $N = 256$ and an MLP width $D = 1024$ are therefore chosen for this task. The learned parameters lead to a solution which is not perfect: at the edge of the interval the learned model struggles. This can be improved with bigger models. However, the learned architecture is able to capture most of the system's chaotic behavior. Surprisingly, compared to LV, we found that here learning just the 1HL-MLP already gives great results: we think this is due to the fact that the sequence length is shorter and the hidden dimension is bigger: the model does not need to adjust eigenvalues as enough variability is provided in the hidden state.