Supporting Information for: Universal set of Observables for the Koopman Operator through Causal Embedding

G Manjunath and A de Clercq

Department of Mathematics & Applied Mathematics, University of Pretoria, Pretoria 0028

Email: manjunath.gandhi@up.ac.za & decle029@umn.edu

This supporting information document is meant to serve three purposes: (i). to provide a detailed motivation and include an additional explanation of some ideas in the main article (ii). mathematical proofs of results claimed in the main article (iii). more details of the methods of the numerical results in the main article.

1 Detailed Motivation

Performing experimental measurements on biological, physical, and artificial systems to obtain a more informative dynamical model has been well established in modern-day science. The practical purpose of obtaining high fidelity models is not only for minimizing the point-wise prediction error. Although better prediction indeed helps both manage better service interruptions and resource management, models that have long-term consistency in their time-averaged characteristics are useful for understanding the response of statistical properties of the perturbations of models that have potentially distinguished applications, for instance in climate science (e.g., [**?**, **?**] and references therein). Although attempts made to build model equations from complex data such as sunspot cycles date back to the 1920s in [**?**], a major breakthrough came through the Takens embedding theorem [**?**] that made state-space reconstruction a valuable tool for any kind of analysis. Often experimental measurements from such systems are not directly the system's states, but a univariate time-series whose span is smaller than the underlying system's dynamics. Takens embedding theorem, under some generic conditions, establishes the learnability of a system that is created out of concatenating sufficiently large previous observations

of a dynamical system into a vector (called a delay coordinate map). The system determined by such vectors is equivalent (topologically conjugate) to the system from which the observed time-series was derived. While this topological result ensures an alternate representation of the underlying system, the quality of the representation remains sensitive to various parameters, and importantly stability. The main reason why stability affects performance is that there is no guarantee that the embedding in the reconstruction space does not lead to globally dissipative dynamics although it is locally conservative. In the absence of global dissipativity, small errors lead to major errors before predictions can totally fail (schematic in Fig. **??**). Theoretical conditions under which the geometry of the attractor can be preserved (e.g.,[**?**]) are not adequate for attractor reconstruction in practice. Moreover, global approximation techniques that find a single map to fit the data often work well only when the data can be fit by functions with low functional complexity, i.e., functions that have relatively fewer oscillatory graphs. An example, where neural networks often failing to learn a representation of the underlying attractor is illustrated in [**?**] and in this article in Fig. **??**.

Although not theoretically assuring accurate reconstruction as in Takens delay embedding, the more recent approaches in machine learning that employ the reservoir computing techniques [**?**, **?**] and data-driven algorithms based on Koopman's theory [**?**, **?**] have shown greater quantitative accuracy in forecasting dynamical systems than what has been achieved through delay-embedding alone. The prominent idea in reservoir computing approaches is to transform the available temporal data into another higher dimensional space through the dynamics of a driven dynamical system (e.g., [**?**, **?**]). Often, the driven dynamical system is an arbitrary network with recurrent connections, and its state is post-processed to fit the prediction task by adopting very few connections in the network rather than rigging the entire connectivity in the network from data. Although there is no theoretical framework that guarantees the existence of a post-processing function, it is approximated typically by linear regression. These methods often give good short-term predictions for some specific systems [**?**, **?**], albeit with adaptations. Unfortunately, the well-cited literature barring perhaps [**?**, **?**, **?**, **?**] in this field does not consider a broad class of chaotic dynamical systems where they fail.

The idea behind forecasting using Koopman's theory is to transform the observed data generated from an unknown dynamical system into a new space or a new coordinate system so that the induced dynamics in the new coordinate system can be approximated by a more simple set of equations, for instance by a linear system of equations. The well-known algorithms like Dynamic Mode Decomposition and Extended-Dynamic Mode Decomposition [**?**, **?**] involve a priori determination of a Koopman invariant subspace obtained by a handpicked choice of observables. If the observables and the Koopman invariant subspace they determine are not adequate enough to capture some essential dynamics, the resultant approximation to

the Koopman operator is poor, and hence the qualitative behavior like the phase portrait of the forecasted data veers down to a different one in a very few time-steps. In any case, linear systems in finite dimensions cannot be topologically equivalent to a chaotic map, and hence such methods are limited in performance while modeling complex/chaotic data.

Balancing model complexity with accuracy has recently led to the idea of sparse representations where the vector field of a differential equation that describes the dynamics is approximated by an optimal linear combination of a very few functions (determined through sparse regression) that are 'possibly nonlinear' from a chosen library. Such representations have yielded not only accurate prediction results for some classes of chaotic systems but can be used to build simpler model equations from data e.g., the popular SINDy algorithms in [**?**, **?**]. However, if the library is not big enough to span the vector field, then long-term consistent modeling is not possible for complex data. Besides requiring data obtained at high sampling rates since derivatives are approximated another crucial issue is that the method requires full knowledge of the underlying state space variables that govern the underlying system which is rare in many real-world scenarios.

Library free approaches such as combining delay-embedding and Koopman theory allow to build intermittently forced linear models [**?**, **?**] combining delay embedding and Koopman theory. However, unlike Takens embedding, the existing reservoir computing or data-driven methods do not ensure finding an equivalent (conjugate) model of the data. In the case of Koopman analysis, by capturing a portion of the spectrum like the eigenvalues [**?**, **?**] of the operator when they exist, only a topological semi-conjugacy can be obtained (see Section 2). Although there is talk about finding faithful representations in the Koopman framework leading to topological conjugacy [**?**] there is currently no framework that guarantees existence or a method to find them.

On the positive side, through generalization of Takens embedding (e.g., [**?**, **?**, **?**, **?**], and notably [**?**, Theorem 1], generic continuous-time-lagged observations of attractors of homeomorphisms on compact spaces have been shown sufficient for attractor reconstruction. On preserving the geometry of the reconstructed attractor, some recent theoretical and empirical studies suggest that it may be possible to construct delay embeddings that are not only homeomorphic or diffeomorphic to the original system but nearly isometric [**?**, **?**]. Isometric embeddings for instance obtained by the Nash embedding theorem help preserving local neighborhoods of points on the attractor, a geometric feature desired during forecasting and reconstructing attractors. More recent studies are suggestive that with large enough time-lagged observations [**?**, **?**], isometric embeddings can be obtained.

With the objective of getting close to an isometric embedding through large, perhaps

infinite, time-lagged observations we employ a driven (dynamical) system with the ability to causally embed a dynamical system so as to transform the entire left-infinite history of the temporal data onto just a pair of points (in practice, a pair of finite-dimensional vectors) in a different space. Remarkably, we find that such infinite-delay maps induced by such driven systems determine the observables for the Koopman operator of the inverse-limit system (e.g., [**?**]) of the dynamical system generating the data. This, in conclusion, gives a topological conjugacy to the dynamical systems arising from the discretization of ordinary differential equations, or more generally homeomorphisms. The conjugacy determines a finite faithful representation raised in [**?**]. Remarkably, the observables in the faithful representation are universal, in the sense that they remain unchanged and all the driven system does is to restrict the observables to a new subspace of the inverse-limit system once the underlying dynamical system is changed. This not only solves a highly pursued problem in finding a set of observables for the Koopman operator needed for theoretically precise forecasting dynamical systems but also prevents expert human intuition or machine optimization needed for choosing observables or a library of functions specific to a dynamical system.

Furthermore, we obtain *exact* equations from the data of a dynamical system, the data could be the actual states or obtained as observations in Takens delay-embedding. The equations are determined after learning a map from the data and have several auxiliary variables. Since the driven system that appears in these equations has mathematically proven stability properties [**?**, **?**], the forecasting results show greater stability to noise and parameters although it introduces auxiliary variables. Empirical evidence shows that a linearity measure like a Pearson correlation coefficient points at reduced functional complexity while learning the dynamics in the state space of the driven system. We demonstrate long-term topological consistency (attractor is learnt) and statistical consistency (density of the orbits) of the models obtained through data from standard benchmark chaotic systems and also on systems that show intermittency – Type I intermittency is a feature of transition to turbulence and convection in confined spaces [**?**], seismic data [**?**] and anomalous diffusion in biology [**?**] and are extremely difficult to model from data. In summary, we learn the action of the Koopman operator on observables of the inverse-limit system of the underlying dynamical system, and these observables are determined by a driven system and data. This in turn helps obtain exact equations from data and thus robust high-fidelity models.

2 Preliminaries

A dynamical system in this work is a tuple (U, T) where U is a compact metric space and $T : U \to U$ is a function (that is not necessarily continuous). Henceforth, we consider only surjective dynamical systems, i.e., systems for which T is surjective. In fact, if A is any invariant set, i.e., if $T(A) := \cup_{u \in A} Tu = A$, then T is surjective on A, so if T is not surjective to start with we can restrict the non-transient dynamics to an invariant set. We call any sequence $\bar{u} = \{u_n\}_{n \in \mathbb{Z}}$ that obeys the update equation, $u_{n+1} = Tu_n$ where $n \in \mathbb{Z}$ as an orbit of T. Subsets of $\mathbb{R}^d$ are endowed with the standard topology induced from $\mathbb{R}^d$. The n-fold composition of a self-map f on any space is denoted by $f^{(n)}$. The ω-limit set $\omega(x; f)$ of a point x is the collection of limit points of the sequence $\{x, f(x), f^2(x), \ldots\}$; note that when f is defined on a compact space, $\omega(x; f)$ is always non-empty. Throughout, if $F : X \to Y$ is any function, and Z is a subset of X, for brevity of notation we denote the restriction of F on Z, $F_{|Z}$ by just F when the context is clear.

A core concept in dynamical systems theory is the notion of equivalent dynamical systems. Finding an equivalent dynamical system to (U, T) means finding another dynamical system (V, S) so that there exists a homeomorphism $\phi : U \to V$ with the property that $\phi \circ T = S \circ \phi$. Such a map ϕ is called a conjugacy and we say that (V, S) is conjugate to (U, T). If we relax the condition on ϕ where, instead of having a homeomorphism, we only require ϕ to be continuous, then we call ϕ a semi-conjugacy and say the systems are semi-conjugate. A system being conjugate to the original means that there is a one-to-one correspondence between the two systems, whereas a semi-conjugacy when it is many-to-one mapping provides a coarse-grained description of the original system. When (V, S) is semi-conjugate to (U, T) then it is customary to say that (V, S) is a factor of (U, T) or that (U, T) is an extension of (V, S). When the underlying spaces are clear, we just say S is a factor of T or T is an extension of S.

3 Takens Delay Embedding

The Takens Delay Embedding Theorem establishes that concatenating a sufficiently large number of previous observations of a dynamical system into a vector (schematic in Fig. **??**) can generate a map between the vectors under some conditions. We recall the theorem from [**?**] below.

Theorem 1 (**Takens Embedding Theorem (adopted from [?]).** *Let W be a compact manifold of dimension m, and $d \geq m$ so that $2d$ is an integer. It is a generic property for the pair (T, θ), where $T : W \to W$ is a smooth diffeomorphism,*

and $\theta : W \to \mathbb{R}$ a smooth function, the map $\Phi_{2d,\theta} : W \to \mathbb{R}^{2d+1}$ defined on W by $\Phi_{2d,\theta}(w) := (\theta(T^{-2d}w) \ldots, \theta(T^{-1}w), \theta(w))$ is a diffeomorphic embedding; by 'smooth' we mean at least C^2. Consequently, there exists a map $F_\theta : \Phi_{2d,\theta}(W) \to \Phi_{2d,\theta}(W)$ defined by

$$F_\theta : (\theta(T^{-2d}w), \ldots, \theta(T^{-1}w), \theta(w)) \mapsto (\theta(T^{-2d+1}w), \ldots, \theta(T^{-1}w), \theta(Tw))$$

so that (W, T) is topologically conjugate to $(\Phi_{2d,\theta}(W), F_\theta)$.

The map $\Phi_{2d,\theta}$ is called the delay-coordinate map. There are now various relaxations on the hypotheses required for such a map F_θ to exist in recent works using the central idea of using a delay-coordinate map $\Phi_{2d,\theta}$ to embed an attractor. The map F_θ can be learnt through a finite segment of an orbit of F_θ, and any such learnt map $\widetilde{F}_\theta$ would also accept arguments that are outside $\Phi_{2d,\theta}(W)$ as well. The above theorem does not guarantee that the set $\Phi_{2d,\theta}(W)$ is an attractor of the system defined by the map $\widetilde{F}_\theta$ (See Fig. **??**). When $\Phi_{2d,\theta}(W)$ is not an attractor, as schematically illustrated in Fig. **??**), a small amount of error could leave an iterate of $\widetilde{F}_\theta$ outside $\Phi_{2d,\theta}(W)$. This could then result in the future iterates under $\widetilde{F}_\theta$ moving far away from $\Phi_{2d,\theta}(W)$ resulting in erroneous prediction.

To demonstrate such fragility in forecasting, we consider the evolution of the image of the time-series obtained by an observation of the Lorenz system under the delay-coordinate map with a delay equal to 10. Specifically, if (w_n) denote the samples of the Lorenz System (as in (22)), with $(w_{n,x}), (w_{n,y})$, and $(w_{n,z})$ denoting the x, y, and z coordinates of (w_n), then we consider the observation $\theta_n = \theta(w_n) = \frac{1}{10}(\sin(0.1w_{n,x}) + \sin(0.1w_{n,y}) + \sin(0.1w_{n,z}))$. The map F_θ is thus the mapping $(\theta_{n-10}, \cdots, \theta_{n-1}) \mapsto (\theta_{n-9}, \cdots, \theta_n)$, which we approximate using a feedforward neural network. To illustrate the failure in forecasting after having learnt $\widetilde{F}_\theta$, we plot the three main principal component of the iterates $\widetilde{F}_\theta$ (in blue), and the three main principal component of the delay coordinates $\Phi_{2d,\theta}(T^{(k)}w)$ (in red) in Fig. **??**).

Figure was here

In practice, data-driven algorithms and other machine learning algorithms outperform Takens embedding often.

Figure was here

4 Koopman Operator and Forecasting

Given a dynamical system (U, T), and a vector space V of observables f whose domain is U, the operator $\mathcal{K} : V \to V$ so that $\mathcal{K}f = f \circ T$ holds is called the Koopman operator. If one knows the action of the Kooopman operator on an observable f, then since $\mathcal{K}f(u) = f(T(u))$ and likewise $\mathcal{K}f(Tu) = f(T^{(2)}u)$, one can forecast the observed values $(f(Tx), f(T^{(2)}u), f(T^{(3)}u), \ldots)$. Often for an analysis or approximation of the Koopman operator, the space of observables V is a Banach space or a Hilbert space over the field of complex numbers. In this case a complex number $\lambda \in \mathbb{C}$ and an associated $\phi \in V$ is called an eigenvalue and eigenfunction respectively if $\mathcal{K}\phi = \lambda\phi$. Thus, when ϕ is an eigenfunction associated with λ, then it follows from [**?**, **?**, **?**] that the following diagram commutes:

and hence the dynamical system $(\phi(U), F_\lambda : u \mapsto \lambda u)$ is topologically semi-conjugate to (U, T). So every eigenvalue captures a coarse-grain description of (U, T), and the representation gains physical meaning when the eigenfunctions are non-constants. Finding non-constant eigenfunctions is difficult even while the map T is known. However, they can be determined from data (e.g., [**?**, **?**]). One of the central ideas in using Koopman's theory for forecasting dynamical systems is to obtain a collection of observables $\mathbf{f} = \{f_1, \ldots, f_N\}$ so that the resultant dynamics from $\mathbf{f}(u) \mapsto \mathbf{f}(Tu)$ can be approximated by a map, say F with a lower functional complexity exploiting the fact that $\mathcal{K}$ is linear in V. The choice of the L^p space considered determines the spectrum [**?**] of the Koopman operator $\mathcal{K}$, and when it is a Hilbert space, and the span of the observables is invariant under $\mathcal{K}$, one can capture a portion of the spectrum of $\mathcal{K}$ through F. More specifically eigenvalues of F would belong to the spectrum of $\mathcal{K}$ (e.g., [**?**, **?**]). The interesting feature of such approximation is that F (in fact its matrix representation) can be derived from the observed data $\{f_i(T^{(k)}u)\}_{0\le k\le m, 1\le i\le N}$. However, there are many potential issues: (i). the span of arbitrarily chosen observables is not necessarily invariant under the Koopman operator (ii) one does not know how to expand the set of observables to capture more eigenvalues and concomitantly retaining invariance under the Koopman operator (iii). Above all, there is no guarantee that the Koopman operator has relevant eigenvalues for approximating the salient feature of the dynamics of the underlying system. For example, non-isolated eigenvalues and/or continuous spectra of the Koopman operator are a feature of many complex systems especially those which exhibit chaos (e.g., [**?**]).

Sparse identification of nonlinear dynamical systems (SINDy) [**?**, **?**] overcomes the difficulty with linear approximations considerably – rather than aiming at capturing eigenvalues, an optimal linear combination of elements in a pre-determined library is determined from the observed data to approximate the vector field of the unknown dynamical systems. The main disadvantage of this method is that the method requires full knowledge of the underlying state space variables that govern the underlying system which is rare in many real-world scenarios. Further, when the library is not big enough to span the vector field, then long-term consistent modeling is not possible for complex data. For instance, if the library comprises polynomials and rational functions good approximations are obtainable only if the underlying system has polynomial and rational nonlinearities, and can fail when the underlying system has a nonrational and nonpolynomial term [**?**]. Also since time-derivatives are found, one needs data at a very high sampling rate with lower noise.

Library free approaches such as combining delay-embedding and Koopman theory allows to build intermittently forced linear models [**?**, **?**]. However, unlike Takens embedding, the existing reservoir computing or data-driven methods do not guarantee an equivalent (conjugate) model of the data. Recently, Igor Mezić proposed the idea [**?**] of learning the action of the Koopman operator on a finite set of observables $\mathbf{f} = (f_1, f_2, \ldots, f_N)$ so that we can find a map G so that $\mathbf{f}$ is a topological conjugacy

between T and G, i.e., we determine $(\mathbf{f}, G)$ so that the following diagram commutes:

(2)

When such commutativity holds, the author in [**?**] calls $(\mathbf{f}, G)$ a finite faithful representation of the dynamical system. Finding a finite faithful representation would avoid approximating the Koopman operator but would instead enable to learn the action of the Koopman operator on the set of observables determined by $\mathbf{f}$. In essence, we can abandon the whole idea of approximating the spectrum of the Koopman operator while forecasting. In this work we approach to solve the problem of finding a finite faithful representation of the inverse-limit system of a dynamical system [**?**] which is an extension of the original system (U, T); extensions always contain the spectrum of their factors [**?**]. Also, we observe that employing observables from inverse-limit systems would help us forecast a system that is greatly sensitive to the distant past much more than the immediate past.

5 Driven Systems: Some Basics

A driven system comprises an input metric space (U, d_U), a compact metric (state) space (X, d) and a **continuous function** $g : U \times X \to X$. For brevity, we refer to g as a driven system with all underlying entities quietly understood.

A bi-infinite sequence $\bar{u} = \{u_n\}_{n\in\mathbb{Z}} \subset U$ which we call an input, induces a sequence of self-maps $\{g(u_n, \cdot)\}_{n\in\mathbb{Z}}$ defined on X, and the dynamics on X generated through this sequence os self-maps is given by the update equation $x_{n+1} = g(u_n, x_n)$.

Given a driven system g and an input $\bar{u}$, we call a sequence $\{x_n\}$ a solution if it satisfies $x_{n+1} = g(u_n, x_n)$ for all $n \in \mathbb{Z}$. We denote a solution obtained by $\bar{u}$ as $\{x_n(\bar{u})\}$.

We next identify a subspace X_U of X that contains all possible solutions. To realize such a subspace of a driven system g, we define the *reachable set* of the driven system g to be the union of all the elements of all the solutions, i.e.,

$$X_U := \Big\{x \in X : x = x_k \text{ where } \{x_n\} \text{ is a solution for some } \bar{u}\Big\}.$$

For example, when $U = [0, 1]$ and $X = [0, 1]$, for the driven system $g(u, x) := \frac{ux}{2}$ regardless of any input sequence in U, $x_n \equiv 0$ for $n \in \mathbb{Z}$ is the only solution of g, and hence the reachable set $X_U = \{0\}$. Also for example, when $U = [0, 1]$ and $X = [0, 1]$,

$g(u, x) = x$ regardless of any input sequence in U, all constant sequences contained in X are solutions of g, and hence the reachable set $X_U = X$.

In both these examples, we have hit somewhat extreme cases that are not useful, and we would need the reachable set to be relatable to the temporal input; we will make precise of what we mean by "relatable" later in Definition 1. To restrict to our attention to more useful cases, we say g is SI-invertible if $g(\cdot, x) : U \to X$ is invertible for all x, i.e., if the current state x_n and the future state x_{n+1} are given, then the current input u_n can be uniquely determined, since the inverse of $g(\cdot, x_n)$ exists. For instance, consider a recurrent neural network (RNN) with N artifical neurons and $U \subset \mathbb{R}^N$ and $X = [-1, 1]^N$ (the cartesian product of N copies of $[-1, 1]$) given by

$$g(u, x) = (1 - a)x + a\overline{\tanh}(Au + \alpha Bx), \tag{3}$$

where A is a $N \times N$ matrix with input connections called the input matrix. The matrix B is also of dimension $N \times N$ representing the strength of the interconnections called a reservoir matrix, and a and α are real-valued parameter that are normally called leak rate and scaling of the reservoir B respectively and $\overline{\tanh}(*)$ is (the nonlinear activation) tanh performed component-wise on $*$. The RNN accepts inputs as vectors with N elements, and if an input v_n is of dimension $K < N$, it can be embedded into $\mathbb{R}^N$, for instance one can pad $N - K$ zeroes to obtain an input of dimension N i.e., $v_n \mapsto (v_n^1, v_n^2, \ldots, v_n^K, 0, 0, \ldots 0) = u_n$. Since given x_{n+1} and x_n, we can recover u_n by

$$u_n := A^{-1}\left(\overline{\tanh}^{-1}\frac{1}{a}\Big(x_{n+1} - (1 - a)x_n\Big)\right) - \alpha Bx_n \tag{4}$$

g in (3) is SI-invertible.

We now describe the set of all solutions of g for a given input $\bar{u}$ in the nonautonomous dynamical systems setting (e.g., [**?**, **?**]). Suppose a driven system g has been fed input values $u_m, u_{m+1}, \ldots, u_{n-1}$ starting at time m. Then the map g transports a state-value $x \in X$ at time m to give a state-value $g_{u_{n-1}} \circ \cdots \circ g_{u_m}(x)$ at time n.

Formally, for every choice of $\bar{u} = (\ldots, u_{-1}, u_0, u_1, \ldots)$, we define for all pair of integers $m \leq n$, the function that 'transports' a system state at x at time m through the inputs $u_m, u_{m+1}, \ldots u_{n-1}$ to the state at time n given by a composition-operator called a process by several authors (e.g.,[**?**]) by the map $\phi_{\bar{u}} : \mathbb{Z}^2_{\geq} \times X \to X$, where $\mathbb{Z}^2_{\geq} := \{(n, m) : n \geq m, n, m \in \mathbb{Z}\}$ and

$$\phi_{\bar{u}}(n, m, x) := \begin{cases} x & \text{if } n = m, \\ g_{u_{n-1}} \circ \cdots \circ g_{u_{m+1}} \circ g_{u_m}(x) & \text{if } m < n. \end{cases} \tag{5}$$

Since $g(u, x) : X \to X$, it easily follows that $\phi_{\bar{u}}(n, m - 1, X) \subset \phi_{\bar{u}}(n, m, X)$ (see [**?**] [**?**]), and since ϕ is continuous in the variable x, these sets are all closed. Further,

since X is compact the nested intersection

$$X_n(\bar{u}) := \bigcap_{m<n} \phi_{\bar{u}}(n, m, X) \tag{6}$$

is nonempty when X is nonempty.

The set $X_n(\bar{u})$ denotes the set of all reachable states at time n if the input is $\bar{u}$. It is a well known result in nonautonomous dynamical systems literature (e.g., [**?**, **?**, **?**]) that $x \in X_n(\bar{u})$ if and only if there is a solution $\{x_k\}$ of g with input $\bar{u}$ so that $x_n = x$. Thus in the special case where we have a topological contraction in (6) where $X_n(\bar{u})$ is a singleton subset of X for each n and $\bar{u}$, then we have exactly one entire-solution for each n.

6 Universal Semi-Conjugacy and the Causal Embedding Theorem

Since the state x_{n+1} of a driven system g at time $n+1$ depends on both the input value u_n and the state value x_n at time n, we consider a question on whether the "complexity" in a solution is exclusively contributed by the input or if the sequence of maps $\{g_{u_n}(\cdot)\}_{n\in\mathbb{Z}}$ also contributes to the complexity in the solution. In the case of autonomous systems on X, i.e., for a self-map $f : X \to X$, the general feature of complex dynamics is that the sequence $\{f^{(n)}\}$ of self-compositions is not equicontinuous. So the question in the case of non-autonomous system $\{g_{u_n}(\cdot)\}_{n\in\mathbb{Z}}$ is not just about equi-continuity, but something more general since the input is involved.

To illustrate the idea behind the question, consider an example of a numerical simulation of a solution of a RNN (as in (3)) with two different parameter sets (see Fig. **??**). A coordinate of a solution is plotted in red and blue for the two parameters and the input sequence is shown in black. As it may be observed, the coordinate of the solution shown in red seems to just follow the input in its temporal variation, while that in blue has wild behavior with an oscillatory envelope. We say that the driven system has introduced new additional complexity to the solution indicated in blue that was not there in the input. In order to mathematically describe the scenario of $\{g_{u_n}(\cdot)\}$ not adding on to the complexity to the solution we consider a definition in Definition 1. We denote $\overleftarrow{u}^n := (\ldots, u_{n-2}, u_{n-1})$ and $\overleftarrow{U}$ denote all the left-infinite sequences in U. Symbolically we let $\overleftarrow{u}^n v := (\ldots, u_{n-2}, u_{n-1}, v)$ denote the input up to time n with $v \in U$ being the input value at time n. This introduction of a new input at time n can be described by a mapping $\sigma_v : \overleftarrow{u}^n \mapsto \overleftarrow{u}^n v$. We would like to talk of continuity of functions defined on the space of left-infinite sequences

contained in U hence we adopt the notation: if Y is a metric space then we denote the product space $\overleftarrow{Y} := \prod_{i=-\infty}^{-1} Z_i$ where $Z_i \equiv Y$ and equip this space with the product topology, and consider the definition of the universal semi-conjugacy.

Figure was here

Definition 1. Given a driven system g, we call a continuous and surjective map $h : \overleftarrow{U} \to X_U$ a universal semi-conjugacy if the following diagram commutes for all $v \in U$:

(7)

We next say a driven system g has the unique solution property (USP) if for each input $\bar{u}$ there exists exactly one solution. In other words, g has the USP if there exists a well-defined solution-map Ψ so that $\Psi(\bar{u})$ denotes the unique solution obtained from the input $\bar{u}$. In our context, the USP is equivalent to saying g is a topological contraction, i.e., each $X_n(\bar{u})$ is a singleton subset of X for all $n \in \mathbb{Z}$ and all $\bar{u}$. The USP notion is independent of SI-invertibility.

The left-finite sequence $\overleftarrow{u} := (\ldots, u_{-2}, u_{-1})$ belonging to $\overleftarrow{U}$ can also be used to equivalently define USP. Consider $\Pi_{n=-\infty}^{n=+\infty} U_i$, where $U_i \equiv U$. Since any left-infinite portion of an element of $\Pi_{n=-\infty}^{n=+\infty} U_i$ is an element of $\overleftarrow{U}$, we can express the definition of the USP more succinctly by denoting $\mathcal{E}(\overleftarrow{u}) := X_0(\bar{u})$, and then say that g has the USP if $\mathcal{E}(\overleftarrow{u})$ is a singleton subset for all $\overleftarrow{u} \in \overleftarrow{U}$. We call $\mathcal{E}$ the encoding function.

We denote the collection of all nonempty closed subsets of X by H_X. On H_X we employ the Hausdorff metric defined by $d_H(A, B) := \max(dist(A, B), dist(B, A)) = \inf\{\epsilon : A \subset B_\epsilon(B) \ \& \ B \subset B_\epsilon(A)\}$, where $B_\epsilon(A) := \{x \in X : d(x, A) < \epsilon\}$ is the open ϵ-neighborhood of A. We could treat the encoding function $\mathcal{E}(\cdot)$ and describe its continuity by treating it either as a multivalued function of $\overleftarrow{u}$ or as a regular function taking values in the space of nonempty compact subsets H_X. In particular when X is a compact metric space the continuity notions become equivalent (e.g., [**?**, Theorem 1, p. 126]). Henceforth, we consider the continuity of $\mathcal{E}(\cdot)$ a H_X-valued function f, with of course H_X being equipped with the Hausdorff metric. When $\mathcal{E}(\cdot)$ is a singleton subset of X then it is always continuous as a set-valued function (e.g., [**?**, **?**]). We define the subspace of H_X that contains the singleton subsets of X by S_X.

We borrow the following facts from [**?**, Section 3]: **(F1).** If $\mathcal{E}(\overleftarrow{v})$ is a singleton subset of X then $\mathcal{E}(\cdot)$ is continuous at $\overleftarrow{v}$. To state our theorem, we define the subspace of H_X that contains the singleton subsets of X by S_X, and define the mapping $i : (X, d) \to (\mathsf{S}_X, d_H)$ by $i(a) = \{a\}$. Clearly i is invertible. Note that i is an

isometry since

$$\begin{aligned} d_H(i(a), i(b)) &= \max\left(\sup_{a\in\{a\}} d(a,b), \sup_{b\in\{b\}} d(b,a)\right) \\ &= \max(d(a,b), d(b,a)) = d(a,b). \end{aligned}$$

We recall results from [**?**, **?**] and state them in a way to suit the context here.

Theorem 2. (Universal Semi-Conjugacy Theorem.) *Let g be a driven system. Then g induces a universal semi-conjugacy $h : \overleftarrow{U} \to X_U$ if g has the unique solution property (USP), i.e., for every input $\{u_n\}$ there exists exactly one entire-solution. Moreover, $h(\ldots, u_{k-2}, u_{k-1}) = x_k$, where $\{x_n\}$ is the solution of g.*

Proof. Let $h(\ldots, u_{k-2}, u_{k-1}) := x_k$, where $\{x_n\}$ is the solution for any input whose left-infinite sequence is $(\ldots, u_{k-2}, u_{k-1})$. Hence $h(\overleftarrow{u}^k) = i^{-1}(\mathcal{E}(\overleftarrow{u}^k))$. By definition of $\mathcal{E}$, we find (the required commutativity in the diagram in (7)) through the deduction:

$$\begin{aligned} g_v \circ i^{-1} \circ \mathcal{E}(\overleftarrow{u}) &= i^{-1} \circ \mathcal{E}(\overleftarrow{u}v), \\ &= i^{-1} \circ \mathcal{E}(\sigma_v(\overleftarrow{u})). \end{aligned}$$

It remains to be shown that h is surjective and continuous. By definition of X_U, it follows that $h(\overleftarrow{U}) = X_U$ and hence h is surjective. Also since $h = i^{-1} \circ \mathcal{E}$, and $\mathcal{E}$ is continuous and i^{-1} is continuous as it is an isometry, the function h is continuous. ■

Interestingly, the converse of the above result is also true [**?**, Lemma 5]. The map $g(v, \cdot)$ in (7) actually depends on v. In general, it is not possible to find a map on X_U that is independent of v so that the diagram in (7) commutes. However, when the inputs originate from a dynamical system, we can restrict h to a subspace of $\overleftarrow{U}$ and we then can establish a v-independent map when the *single-delay lag dynamics* is considered on a subset of $X_U \times X_U$. To describe the single-delay lag dynamics formally, we consider a dynamical system $T : U \to U$ and we define a relation on the reachable set X_U, i.e., a subset defined on $X_U \times X_U$ by

$$Y_T := \{(x_{n-1}, x_n) : \{x_k\}_{k\in\mathbb{Z}} \text{ is a solution for some orbit of } T \text{ and } n \in \mathbb{Z}\}.$$

The following result shows that T induces a self-map G_T on Y_T and its iterates describes the single-delay lag dynamics.

Theorem 3. *Let g be a driven system that is SI-invertible, and (U,T) be a dynamical system. Consider the subspace Y_T of $X_U \times X_U$ to be the tuple arising from two successive points of any solution of g, i.e.,*

$$Y_T := \{(x_{n-1}, x_n) : \{x_k\}_{k\in\mathbb{Z}} \text{ is a solution for some orbit of } T \text{ and } n \in \mathbb{Z}\}.$$

Then we have a well-defined map $G_T : Y_T \to Y_T$ defined by $G_T : (x_{n-1}, x_n) \mapsto (x_n, x_{n+1})$. Consequently, the mapping $(x_{n-1}, x_n) \mapsto u_n$ is well-defined when x_{n-1} and x_n are successive points on a solution obtained for an input $\{u_n\}$ that is an orbit of T.

Proof. Since $x_n = g(u_{n-1}, x_{n-1})$ and $g(\cdot, x) : U \to X$ is invertible, we have

$$u_{n-1} = g^{-1}_{*,x_{n-1}}(x_n), \tag{8}$$

where $g^{-1}_{*,x}$ is the inverse of the map $g(\cdot, x) : U \to X$. Using this in $x_{n+1} = g(u_n, x_n) = g(Tu_{n-1}, x_n)$, we have $x_{n+1} = g(Tg^{-1}_{*,x_{n-1}}(x_n), x_n)$. Therefore, there exists a function $\theta_T : (x_{n-1}, x_n) \mapsto x_{n+1}$. As a consequence we can define a map $G_T : (x_{n-1}, x_n) \mapsto (x_n, x_{n+1})$ by

$$G_T : (x_{n-1}, x_n) \mapsto (x_n, \theta_T(x_{n-1}, x_n)). \tag{9}$$

Lastly, since $u_{n-1} = g^{-1}_{*,x_{n-1}}(x_n)$ and $u_n = Tu_{n-1}$, there exists a mapping $(x_{n-1}, x_n) \mapsto u_n$. ■

Definition 2. If (U,T) is a dynamical system then we call (Y_T, G_T) the dynamical system induced by g.

The natural question that arises is if we can obtain a relationship like a topological conjugacy or a semi-conjugacy between the single-delay lag dynamics described by (Y_T, G_T) and (U,T). The answer is affirmative when g has the USP but indirect. We can establish such a conjugacy/semi-conjugacy with what we call the inverse-limit system of (U,T) that we explain next.

Roughly, the inverse-limit system of (U,T) comprises a self-map on a subset of an infinite dimensional space (e.g., the Hilbert cube) where each point in the inverse-limit space corresponds to a backward orbit. Formally, any dynamical system (U,T) determines a nonempty subspace $\widehat{U}_T$ of $\overleftarrow{U}$ given by

$$\widehat{U}_T := \{(\ldots, u_{-2}, u_{-1}) : Tu_n = u_{n+1}\}, \tag{10}$$

and $\widehat{U}_T$ is equivalent to the inverse-limit space of (U,T) or the natural extension of (U,T) in the literature (e.g., [**?**]). Note that the inverse-limit space is well-defined since we have assumed that $T : U \to U$ is surjective. It is customary to write the

inverse-limit space as a space comprising right-infinite sequences in the literature instead of left-infinite sequences that we have considered in (10).

The map T also induces a self-map $\widehat{T}$ on $\widehat{U}_T$ defined by $\widehat{T} : (\ldots, u_{-2}, u_{-1}) \mapsto (\ldots, u_{-2}, u_{-1}, T(u_{-1}))$. We refer to the dynamical system $(\widehat{U}_T, \widehat{T})$ as the inverse-limit system of (U, T). The inverse-limit system is an extension of (U, T), since

(11)

holds where $\pi_{-1}(\ldots, u_{-2}, u_{-1}) = u_{-1}$. Further, the inverse-limit systems do not introduce any new complexity into the dynamics in the sense the topological entropies of (U, T) and $(\widehat{U}_T, \widehat{T})$ are identical [**?**]. Also, (U, T) satisfies several other topological properties [**?**] if and only if $(\widehat{U}_T, \widehat{T})$ does.

For our analysis with inputs being restricted to be orbits of T, and particularly while dealing with left-infinite spaces as in (7), it is sufficient to restrict the system on the top in (7) to the inverse-limit system $(\widehat{U}_T, \widehat{T})$. This is since the new input value at any time would be the image of the previous value under T, i.e., if u_{-1} is the current input value, then Tu_{-1} is the next input value. Hence $\sigma_v(\overleftarrow{u})$ in (7) is $\sigma_{Tu_{-1}}(\overleftarrow{u})$ and this is equal to $\widehat{T}(\overleftarrow{u})$. Note that if $\{x_k(\bar{u})\}$ is a solution, then x_n would have been influenced only by the left-infinite sequence $(\ldots, u_{n-2}, u_{n-1})$ which belongs to $\widehat{U}_T$. In this setting of inputs originating from a dynamical system (U, T), we establish a relationship between the inverse-limit system of (U, T) and its induced dynamical system (Y_T, G_T):

Theorem 4. (Causal Embedding Theorem.) *Let g be a driven system that is SI-invertible and has the USP. Let h denote the universal semi-conjugacy and $H_2(\overleftarrow{u}) := (h(r\overleftarrow{u}), h(\overleftarrow{u}))$, where r is the right-shift $r : (\cdots, u_{-2}, u_{-1}) \mapsto (\cdots, u_{-3}, u_{-2})$. Let $(\widehat{U}_T, \widehat{T})$ be the inverse-limit system of any dynamical system (U, T). The function H_2 restricted to $\widehat{U}_T$ is a topological semi-conjugacy between the inverse-limit system $(\widehat{U}_T, \widehat{T})$ and the induced dynamical system (Y_T, G_T), i.e., the following diagram commutes*

(12)

or in other words, (Y_T, G_T) is a factor of $(\widehat{U}_T, \widehat{T})$. Further, if $T : U \to U$ is a homeomorphism, then H_2 embeds $\widehat{U}_T$ in $X_U \times X_U$, and hence (Y_T, G_T) is conjugate to $(\widehat{U}_T, \widehat{T})$.

Proof. From Theorem 2, we know that h is continuous and the following diagram commutes:

(13)

Consider the inverse-limit space $\widehat{U}_T \subset \overleftarrow{U}$, and a bi-infinite orbit $\bar{u}$ of T. The left-infinite segment $\overleftarrow{u} = (\ldots, u_{-2}, u_{-1})$ then belongs to $\widehat{U}_T$. If we restrict the map σ_v

to $\widehat{U}_T$, and select $v = Tu_{-1}$, then we have $\sigma_v(\ldots, u_{-2}, u_{-1}) = \sigma_{Tu_{-1}}(\ldots, u_{-2}, u_{-1}) = \widehat{T}(\ldots, u_{-2}, u_{-1})$. Let $(x_{n-1}, x_n) \in Y_T$. Without loss of generality, let $n = 0$, i.e., $(x_{-1}, x_0) \in Y_T$. By definition of Y_T, there exists an orbit $\{u_k\}$ of T for which $\{x_n\} = \Psi(\{u_k\})$ so that $(x_{-1}, x_0) \in Y_T$ (recall Ψ is the solution map). Also by Theorem 2, and since T is surjective, $h(\overleftarrow{u}) = x_0$ and $h(r\overleftarrow{u}) = x_{-1}$ where $\overleftarrow{u} = (\ldots, u_{-2}, u_{-1})$ is the left-infinite segment of the orbit $\{u_k\}$. Hence $H_2 : \widehat{U} \to Y_T$ is surjective. Since h and r are continuous, it follows that H_2 is also continuous. Also, since $h \circ \widehat{T}(\overleftarrow{u}) = x_1$, we have $H_2 \circ \widehat{T}(\overleftarrow{u}) = (x_0, x_1) = G_T(x_{-1}, x_0)$. Using all of these in (13), the diagram in (12) commutes, and (Y_T, G_T) is a factor of $(\widehat{U}_T, \widehat{T})$.

It remains to be shown whenever T is a homeomorphism then H_2 does not map two distinct points in $\widehat{U}_T$ to the same point. Suppose there exists two orbits $\{u_k\}$ and $\{v_k\}$ so that $\overleftarrow{u} \neq \overleftarrow{v}$ and $H_2(\overleftarrow{u}) = H_2(\overleftarrow{v})$. This means that the solutions $\{x_k\} = \Psi(\{u_k\})$ and $\{y_k\} = \Psi(\{v_k\})$ are such that $(x_{-1}, x_0) = (y_{-1}, y_0)$. Since g is SI-invertible, this implies $u_{-1} = v_{-1}$. Since T is a homeomorphism, u_{-1} and v_{-1} have the same left-infinite history which implies $\overleftarrow{u} = \overleftarrow{v}$. Hence when T is a homeomorphism, H_2 embeds $\widehat{U}_T$ in $X_U \times X_U$. By definition of Y_T, $H_2(\widehat{U}_T) = Y_T$ and hence $H_2 : \widehat{U}_T \to Y_T$ is a homeomorphism. Thus (Y_T, G_T) is topologically conjugate to $(\widehat{U}_T, \widehat{T})$. ■

Corollary 1. *Let $X \subset \mathbb{R}^N$, and $[H_2]$ denote the collection of all component functions of $H_2 : \widehat{U}_T \to Y_T$ each of which that maps X into $\mathbb{R}^2$. If $\mathcal{K}$ denotes the Koopman operator of the dynamical system $(\widehat{U}_T, \widehat{T})$ then $([H_2], G_T)$ is a finite-dimensional faithful representation and, for each $f_i \in [H_2]$, $\mathcal{K}f_i = \pi_i \circ G_T \circ H_2$, where π_i picks the i^{th} component function of $H_2 : \widehat{U}_T \to X_U \times X_U$.*

Proof. If π_i picks the i^{th} component function of H_2 then $f_i = \pi_i \circ H_2$. When the diagram in (13) commutes, we have $H_2 \circ \widehat{T} = G_T \circ H_2$. By composing π_i on both sides, we have implies $f_i \circ \widehat{T} = \pi_i \circ G_T \circ H_2$. But by definition of $\mathcal{K}$, $f_i \circ \widehat{T} = \mathcal{K}f_i$. Hence, $\mathcal{K}f_i = \pi_i \circ G_T \circ H_2$ when $\mathcal{K}$ is the Koopman operator of $\widehat{T}$. ■

The reader may note that when g is a RNN as in (3) and if $\{f_1, \ldots, f_N\}$ denotes the N component functions of $H_2(\overleftarrow{u}^n)$ then each $f_i(\overleftarrow{u}^n)$ pairs the i^{th} coordinate of x_{n-1}, i.e., the state of the i^{th} node at time $n-1$ with the state of i^{th} node at time n (i^{th} coordinate of x_n) where $x_n = h(\overleftarrow{u}^n)$, and thus f_i's define a set of $\mathbb{R}^2$-valued observables of the inverse-limit space $(\widehat{U}, \widehat{T})$. According to the definition of a faithful representation [**?**] made in Section 4, $\mathbf{f}$ the collection of component functions of H_2 and the map G_T together form a finite faithful representation of $(\widehat{U}, \widehat{T})$.

Definition 3. We say a driven system g *causally embeds* a dynamical system (U, T) if it satisfies the two properties: (i) a universal semi-conjugacy exists, i.e., the diagram in (7) commutes (and thus, (15) also commutes) (ii) $H_2(\overleftarrow{u}) := (h(r\overleftarrow{u}), h(\overleftarrow{u}))$ embeds the inverse-limit space $\widehat{U}_T$ in $X \times X$.

Since when g is SI-invertible and has the USP then $H_2 : \overleftarrow{U}_T \to Y_T$ is a homeomorphism, H_2 can embed the inverse-limit space $(\widehat{U}, \widehat{T})$ of any dynamical system in $X \times X$, and thus causally embed any dynamical system as long as its inverse-limit space is contained in $\overleftarrow{U}$. Synoptically, restricting H_2 to inverse-limit spaces of different dynamical systems contained in $\overleftarrow{U}$ establishes semi-conjugacies/conjugacies between G_T and those inverse-limit spaces. Thus one does not need to change g to learn G_T when the dynamical system that drives it changes. Bringing out this fact is a theoretical advancement that drives us to obtain a universal set of observables for learning the Koopman operator [**?**] of any $(\widehat{U}_T, \widehat{T})$ as long as $\widehat{U}_T$ is contained in $\overleftarrow{U}$.

7 Equations from Data and Forecasting

Since two successive points (x_{n-1}, x_n) of a solution of g with an input $\bar{u}$ lie in Y_T, one can learn the single-lag dynamics of the driven states through the map $G_T : (x_{n-1}, x_n) \mapsto (x_n, x_{n+1})$ from a sufficiently large finite set of data points (x_0, x_1) , $(x_1, x_2), \ldots, (x_{m-1}, x_m)$. Once having a learnt version of G_T, one can iterate to forecast two successive points on a solution corresponding to $\bar{u}$. Forthwith, one can also predict the input value u_n, since two successive states x_n and x_{n+1} determine u_n since g is SI-invertible (14). We recall that g is SI-invertible, any two successive values of the solution (x_n, x_{n+1}) determine u_n uniquely since

$$u_n = g^{-1}_{*,x_n}(x_{n+1}) \tag{14}$$

where $g^{-1}_{*,x}$ is the inverse of the map $g(\cdot, x) : U \to X$. In summary, if G_T is learnt without errors, prediction would be exact whenever H_2 causally embeds (U, T), or else the predicted value of u_n would be an approximation to it obtained from the system (Y_T, G_T) that is semi-conjugate to $(\widehat{U}_T, \widehat{T})$.

We empirically illustrate that the single-lag dynamics described by G_T is less functionally complex than learning T. We consider an RNN as in (3) with the USP, and by varying the dimensions of X (i.e., the neurons) in the RNN we measure the functional complexity of the resultant single-delay dynamics. Empirically (see Table 1), increasing the dimension of X is found to increase the linear relationship (or intuitively reduce the functional complexity of G_T) that is measured as a generalization of the Pearson correlation coefficient to random vectors (e.g., [**?**]) between (x_{n-1}, x_n) and $G_T(x_{n-1}, x_n)$. If Σ_a and Σ_b denotes the covariance matrices of the vectors (x_{n-1}, x_n) and $G_T(x_{n-1}, x_n)$ respectively, and if Σ_{ab} denotes the covariance matrix between the vectors (x_{n-1}, x_n) and $G_T(x_{n-1}, x_n)$, then the multidimensional

correlation coefficient is computed by using the traces of these matrices:

$$\rho = \frac{\mathrm{tr}(\Sigma_{ab})}{\mathrm{tr}(\sqrt{\Sigma_a \Sigma_b})}.$$

This multidimensional correlation coefficient satisfies most of the well-known properties of the one-dimensional Pearson coefficient. In particular, $\rho = \pm 1$ if and only if $Y \stackrel{d}{=} AX + b$ for some invertible matrix A and vector b. Hence ρ, and in practice, the estimator $\hat{\rho}$, can be used to measure the linear relationship between two random vectors of the same dimension and thus serves as an indicator of functional complexity.

We remark that the estimation of such linear relationship in (see Table 1) obtained by sample correlations alone is justified whenever $\{u_n\}$ is a realization of an ergodic process since then when g has the USP, any solution $\{x_n\}$ is also a realization of an ergodic process [**?**]. In the ergodic input case the generalized Pearson correlation coefficient (between (x_{n-1}, x_n) and $G_T(x_{n-1}, x_n)$) is independent of n.

Having observed that learning the single-delay dynamics entails the learning of a less functional complex map, we also note that Theorem 3 there exists a well-defined map $(x_{n-1}, x_n) \mapsto u_n$ when $(x_{n-1}, x_n) \in Y_T$. We denote this map by Γ.

Hence, one can learn G_T indirectly by first learning the map $\Gamma : (x_{n-1}, x_n) \mapsto u_n$. Then clearly $(\Gamma, \pi_2)(x_{k-1}, x_k) = (u_k, x_k)$ where $\pi_2 : (a, b) \mapsto b$ is the coordinate projection. Next, $(\pi_2, g)(u_k, x_k) = (x_k, x_{k+1})$. Therefore, we can rewrite the commutativity in (7) to include the map Γ as:

(15)

Since, $(\pi_2, g)(u_k, x_k) = (x_k, x_{k+1})$ (see the vertical line in red in (15) or that in Fig. **??**A) – we can realize the mapping $(u_k, x_k) \mapsto (x_k, x_{k+1})$ by feeding u_k back to the driven system (a schematic in Fig. **??**). Thus, we obtain iterative roots of G_T, i.e., $G_T = (\pi_2, g) \circ (\Gamma, \pi_2)$. This equivalent representation of G_T entails another map (see the vertical line in red and the diagonal line in red in that order in Fig. **??**A), $S : (u_k, x_k) \mapsto (u_{k+1}, x_{k+1})$ defined by

$$u_{k+1} = \pi_1 \circ (\Gamma, \pi_2) \circ (\pi_2, g)(u_k, x_k) \quad (16)$$

$$x_{k+1} = \pi_2 \circ (\Gamma, \pi_2) \circ (\pi_2, g)(u_k, x_k). \quad (17)$$

Input	Dimension of X	u_n vs u_{n+1}	$\begin{bmatrix} x_{n-1} \\ x_n \end{bmatrix}$ vs $\begin{bmatrix} x_n \\ x_{n+1} \end{bmatrix}$
(w_n) is the Lorenz states, sampled every 0.1 timestamps.			
	10	0.9311	0.9731
$u_n = \frac{1}{100} w_n$	100		0.9934
	1000		0.9930
	10	0.8401	0.9392
$u_n = \frac{1}{10}(\sin(0.1w_{n,x}) + \sin(0.1w_{n,y}) + \sin(0.1w_{n,z}))$	100		0.9616
	1000		0.9737
(w_n) comes from the Hénon map: $w_{n+1} = \begin{bmatrix} 1 - 1.4w_{n,x}^2 + w_{n,y} \\ 0.3w_{n,y} \end{bmatrix}$			
	10	-0.2996	0.4278
$u_n = w_n - \overline{w}$	100		0.5733
	1000		0.4953
(w_n) comes from a Pomeau-Manneville map: $w_{n+1} = \begin{cases} w_n(1 + 2^{0.6} w_n^{0.6}) & \text{if } w_n \leq 0.5 \\ 2w_n - 1 & \text{if } 0.5 < w_n \end{cases}$			
	10	0.6943	0.9260
$u_n = w_n - \overline{w}$	100		0.9542
	1000		0.9669
(w_n) comes from the Logistic map: $w_{n+1} = 4w_n(1 - w_n)$			
	10	-0.0314	0.5577
$u_n = w_n - 0.5$	100		0.7150
	1000		0.7826

Table 1: Multidimensional Correlation Coefficients ρ to indicate a general reduction in the functional complexity of the map G_T that arises from a RNN. Each row corresponds to a different dynamical system used in the experiments in [**?**]. In the second column corresponds to the dimension of X (number of artificial neurons) in the RNN used. The numerical estimate of ρ for the relevant vectors is plotted in the last two columns. The reader can compare the linear relationship between u_n vs $T(u_n)$ and the linear relationship $\begin{bmatrix} x_{n-1} \\ x_n \end{bmatrix}$ vs $G_T\left(\begin{bmatrix} x_{n-1} \\ x_n \end{bmatrix}\right)$. The dimension of the RNN from (3) is also given, showcasing that increasing the dimension of the driven system typically yields a map G_T of lower functional complexity.

The expressions (16)-(17) are equations constructed from data! We digress to note that for a system with the USP, we can replace $h(\overleftarrow{u}^k)$ by x_k in (16) to obtain

$$u_{k+1} = \pi_1 \circ (\Gamma, \pi_2) \circ (\pi_2, g)(u_k, h(\overleftarrow{u}^k)),$$

which is a nonlinear difference equation that only refers to the variable U. Thus the variable x_k acts as an auxiliary variable in the system defined by S.

There are advantages of employing (16)-(17) instead of learning G_T. First, we can save resources by learning Γ instead of learning the map G_T on a subset of a very large dimensional space $X \times X$. This is since in practice u_n is an isometrically embedded image (due to the zero-padding) of the actual input v_n, and since v_n lies in a space with a much lower dimension than X to learn Γ, it is sufficient to learn the map $(x_{n-1}, x_n) \mapsto v_n$.

Secondly, if $\overleftarrow{u}$ and say its noisy version $\overleftarrow{v}$ are nearby then their tails are permitted to be significantly different in the product topology. In that case, $h(\overleftarrow{u})$ and $h(\overleftarrow{v})$ are nearby owing to the continuity of h due to the USP [**?**], and thereby provides robustness to input noise. The continuity in the product topology can be interpreted as follows: when the noise in the input that occurs in the distant past is mitigated by the driven system as time flows. Such robustness to the input noise in the driven system is called input-related stability in [**?**]. More precisely, when there is input-related stability $h(\overleftarrow{u})$ is affected to an arbitrarily small extent by that left-infinite segment of the past of the input beyond some arbitrarily large but finite time [**?**, Theorem 3.1]. Further, when a parameter λ is such that $\lambda \mapsto g_\lambda$ is continuous, the universal semi-conjugacy h_λ is also continuous with respect to the parameter λ. This is referred to as parameter-related stability [**?**, Theorem 3.2], a consequence of the USP.

Lastly, learning G_T directly would lead to the same risk as learning a map after employing delay coordinates in Takens embedding where global dissipativity is not guaranteed (see Fig. **??** and Fig. **??**). This is especially crucial when the underlying dynamical system T exhibits chaos, in which case G_T also would exhibit chaos. This is since $\widehat{T}$ inherits chaos since it is an extension of T, and when G_T is conjugate to $\widehat{T}$, G_T also exhibits chaos. Global approximation of maps that exhibit chaos is numerically stable when it is locally conservative on an invariant set and globally dissipative lest would often lead to errors. An empirical illustration would follow in Fig. **??**). First, we formally define global dissipativity and elaborate with a detailed analysis.

Definition 4. We say a dynamical system (Z, f) is globally dissipative if there exists a nonempty proper closed subset B of Z so that for all $x \in Z \setminus B$, (i). $\omega(x; f) \subset B$ and (ii). B is positively invariant that is $f(B) \subset B$. We call any such closed subset B a trapping set of f.

It is obvious when (Z, f) is globally dissipative, every invariant set or attractor of f would be contained in a trapping set B. In fact one can find that $\bigcap_{n=1}^{\infty} f^{(n)}B$ is an attractor.

An expansion of a set A contained is a fattening of A that is also closed. Formally, an expansion of a set $A \subset \mathbb{R}^n$ ($A \subset \mathbb{R}^n \times \mathbb{R}^n$) is any closed subset $A^+ \subset \mathbb{R}^n$ ($A^+ \subset \mathbb{R}^n \times \mathbb{R}^n$) that contains A. Given a function f defined on A, we denote a function f^+ to be some continuous function that preserves f on A and also leaves A^+ positively-invariant. Formally, suppose f is a continuous function defined on A, then f^+ denotes a choice of a continuous function defined on A^+ so that $f^+(a) = f(a)$ for all $a \in A$, and $f^+(A^+) \subset A^+$. Now if (A^+, f^+) is globally dissipative with a trapping set contained in A, then the forward asymptotic behavior of the orbits under f^+ and f would be identical. Our aim is to show that a system $(Y_T^+, (\pi_2, g^+) \circ (\Gamma_{exp}, \pi_2))$ that is a learnt version of $(Y_T, (\pi_2, g) \circ (\Gamma, \pi_2))$ can have global dissipativity, and hence their forward asymptotic dynamics are identical; Γ_{exp} would be defined in the discussion later. Note that we have assumed that the dynamics of the learnt version has the same dynamics of $(\pi_2, g) \circ (\Gamma, \pi_2)$ on Y_T, and the purpose of this discussion is to analyse what can happen if the iterates of G_T slip on to Y_T^+ due to transient or noise at isolated moments in the input.

First we would consider the map $G_T : Y_T \to Y_T$. Suppose successive iterates of of (x_{n-1}, x_n) under G_T is employed to forecast successive values of a solution of g contained in Y_T, then these iterates would remain in Y_T. However, in practice, due to noise at some moment, an iterate could slip beyond Y_T and into an expansion Y_T^+; for instance Y_T^+ in this case could be

$$Y_T^+ := \text{Closure}\Big(Y_T \cup \cup_{n>0}(y_{n-1}, y_n)\Big),$$

where y_i is an approximation of x_i. In practice, a function like a feedforward neural network that is made to learn G_T would also accept a value outside Y_T as an argument. So in practice, we have a map G_T^+ on an expansion Y_T^+. In general, Y_T need not be an attractor of $G_T^+ : Y_T^+ \to Y_T^+$, or more generally there is no guarantee that (Y_T^+, G_T^+) is globally dissipative. Especially for chaotic maps, a map G_T^+ could continue to show sensitive dependence on initial conditions on $Y_T^+ \setminus Y_T$ as well, which could take the iterates of G_T^+ further away from Y_T, and hence forecasting leads to large errors (see Fig. **??** for a schematic, and an empirical illustration in the figure on the left in Fig. **??**). If we learn Γ instead of G_T, then the system $(Y_T^+, (\pi_2, g^+) \circ (\Gamma_{exp}, \pi_2))$ which is what we realise in practice while learning $(\pi_2, g) \circ (\Gamma, \pi_2)$ in (15) turns out to be globally dissipative (Theorem 5) under some conditions on g; here Γ_{exp} denotes some function defined on Y_T^+ that agrees with Γ on Y_T. In summary, by feeding the predicted values of the input into the driven system g during forecasting as in Fig. **??** or more formally, as indicated in [**?**, Fig. **??**A], the iterates under $(Y_T^+, (\pi_2, g^+) \circ (\Gamma_{exp}, \pi_2))$ would not wander far away from the iterates

of $(\pi_2, g) \circ (\Gamma, \pi_2)$ in practice. (for a schematic see Fig. **??** and for an empirical illustration see the figure on the right in Fig. **??**).

Figure was here

Before proving Theorem 5, we deal with how Γ can be learnt reliably from successive points on a simulated solution. Note that for obtaining an actual solution, the entire left-infinite input has to be fed to the driven system. When g has the USP, the solutions of g can be recognized as non-autonomous uniform attractors [**?**], that is each solution attracts all other initial conditions towards the components of the solution, and moreover, this attractivity is uniform for all solutions – uniform attraction property [**?**], which we state next. Recall that when A and B belong to H_X, the quantity $dist(A, B) := \inf\{\epsilon : A \subset B_\epsilon(B)\}$ is the Hausdorff semi-distance between A and B.

Definition 5. Let g be a driven system. Then g is said to have the uniform attraction property (UAP) if for each $\bar{u} \in \overleftarrow{U}$ the process $\phi_{\bar{u}}$ is such that there exists a sequence of singleton subsets $\{A_k(\bar{u})\}$ of X so that $\phi_{\bar{u}}(k, k-j, A_k(\bar{u})) = A_{k+1}(\bar{u})$ and

$$\lim_{j\to\infty} \sup_k \sup_{\bar{u}} dist\Big(\phi_{\bar{u}}(k, k-j, X), A_k(\bar{u})\Big) = 0. \tag{18}$$

It is a result in [**?**, Theorem 1] that the UAP is equivalent to the USP. Now, clearly, when g has the USP, the sequence of subsets that satisfy Definition 5 are subsets containing the elements of the solution $\Psi(\bar{u})$. Hence from (18), we have

$$\lim_{j\to\infty} \sup_k \sup_{\bar{u}} d\Big(\phi_{\bar{u}}(k, k-j, y), x_k(\bar{u})\Big) = 0,$$

for all $y \in X$, and where $\{x_k(\bar{u})\} = \Psi(\bar{u})$. By replacing k by $k+j$, we obtain

$$\lim_{j\to\infty} \sup_k \sup_{\bar{u}} d\Big(\phi_{\bar{u}}(k+j, k, y), x_{k+j}(\bar{u})\Big) = 0, \tag{19}$$

for all $y \in X$, and where $\{x_k(\bar{u})\} = \Psi(\bar{u})$. This in turn implies for any $\bar{u}$, and $k \in \mathbb{Z}$ and any pair $y_1, y_2 \in X$

$$\lim_{j\to\infty} \max\left(d\Big(\phi_{\bar{u}}(k+j-1, k, y_1), x_{k+j-1}(\bar{u})\Big), d\Big(\phi_{\bar{u}}(k+j, k, y_2), x_{k+j}(\bar{u})\Big)\right) = 0, \tag{20}$$

where $\{x_k(\bar{u})\} = \Psi(\bar{u})$.

The limit in (20) ensures that if we initialize the driven system with an arbitrary initial value $y_m \in X$, then the sequence $y_{m+1}, y_{m+2}, y_{m+3}, ...$ satisfying $y_{k+1} = g(u_k, y_k)$

for $k \geq m$ approximates the actual solution $\{x_n\}$ *uniformly* (see (18)). To be precise, given $\epsilon > 0$ (independent of y_m) there is an integer n so that the distance between x_{n+i} and y_{n+i} is less than ϵ for all $i \geq 0$, where $\{x_m\} = \Psi(\bar{u})$ and y_{n+i} is generated by $y_{k+1} = g(u_k, y_k)$ for $k \geq m$. Thus by leaving out a few values of y_i (termed as washing out initial conditions in the RC literature [**?**]), for practical purposes the subsequent values of y_i are indistinct from the actual solution values. In contrast, the Takens delay embedding theorem in [**?**] does not guarantee embedding a subset of the manifold as an attractor (recall Fig. **??**).

We next elaborate on the conditions on g^+ so that $(Y_T^+, (\pi_2, g^+) \circ (\Gamma_{exp}, \pi_2))$ is globally dissipative. Now suppose g^+ is such that $g^+(U^+ \times X^+) \subset X$. An example of this is the RNN in (3). In this case all solutions of g^+ would lie within $X = [-1, 1]^N$ thanks to the tanh function. Hence Y_T^+, an expansion of Y_T would be always contained in $X \times X$. Next, let $U^+ = \Gamma_{exp}(Y_T^+)$ be an expansion of U, where Γ_{exp} is the learnt version of Γ with domain Y_T^+. Clearly, U^+ is compact since X is compact and Γ_{exp} is continuous. In this scenario, if g^+ has the USP, then it turns out that the product of the reachable set $X_{U^+} \times X_{U^+}$ is a trapping set of the map $(\pi_2, g^+) \circ (\Gamma_{exp}, \pi_2)$ as indicated in Fig. **??**. As promised a formal proof is in Theorem 5.

Figure was here

Theorem 5. *If g^+ is such that $g^+ : U^+ \times X^+ \to X$ and has the USP then $(\pi_2, g^+) \circ (\Gamma_{exp}, \pi_2)$ is globally dissipative with $X_{U^+} \times X_{U^+}$ being a trapping set.*

Proof. By definition $(\pi_2, g^+) \circ (\Gamma_{exp}, \pi_2)$ is a self-map on Y_T^+. For brevity of notation, set $f := (\pi_2, g^+) \circ (\Gamma_{exp}, \pi_2)$. To prove the theorem, we show that $\omega((y_1, y_2); f) \subset X_{U^+} \times X_{U^+}$.

Fix $(y_{-1}, y_0) \in Y_T^+$. Let $u_0 := \Gamma_{exp}(y_{-1}, y_0)$. In general denote $u_i := \Gamma_{exp}(f^{(i)}(y_{-1}, y_0))$. So under the repeated iterations of f we also find an input sequence $\{u_0, u_1, u_2, \ldots \subset U^+\}$. Let $\bar{u}$ be the bi-finite sequence contained in U^+ so that its right infinite sequence matches with $\{u_1 u_2, \ldots\} \subset U^+$. Let $\phi_{\bar{u}}^+$ be the composition operator of g^+ with input $\bar{u}$. Then by definition of $\bar{u}$, we have $f(y_{-1}, y_0) = (y_0, \phi_{\bar{u}}^+(1, 0, y_0)) = (\phi_{\bar{u}}^+(0, 0, y_0), \phi_{\bar{u}}^+(1, 0, y_0))$. In general, for all $j > 0$,

$$f^{(j)}(y_{-1}, y_0) = \Big(\phi_{\bar{u}}^+(j-1, 0, y_0), \phi_{\bar{u}}^+(j, 0, y_0)\Big).$$

Let $\{x_k\} := \Psi(\bar{u})$. By definition of the reachable set X_{U^+}, $(x_k, x_{k+1}) \in X_{U^+} \times X_{U^+}$ for all $k \in \mathbb{Z}$. The product $X_{U^+} \times X_{U^+}$ is compact since by definition X_{U^+} is compact. Hence to prove that $\omega((y_1, y_2); f) \subset X_{U^+} \times X_{U^+}$ it is sufficient if we show that $f^{(j)}(y_{-1}, y_0) \to (x_{j-1}, x_j)$ as $j \to \infty$. Recalling (20), we have for all $k \in \mathbb{Z}$, and

(z_1, z_2),

$$\lim_{j\to\infty} \max\left(d\Big(\phi_{\bar{u}}^{+}(k+j-1,k,z_1), x_{k+j-1}(\bar{u})\Big), d\Big(\phi_{\bar{u}}^{+}(k+j,k,z_2), x_{k+j}(\bar{u})\Big)\right) = 0.$$

In particular by setting $k=0$, $z_1 = z_2 = y_0$, we have

$$\lim_{j\to\infty} \max\left(d\Big(\phi_{\bar{u}}^{+}(j-1,0,y_0), x_{j-1}(\bar{u})\Big), d\Big(\phi_{\bar{u}}^{+}(j,0,y_0), x_{j}(\bar{u})\Big)\right) = 0$$

which implies $f^{(j)}(y_{-1}, y_0) \to (x_{j-1}, x_j)$ as $j \to \infty$. ■

Figure **??** illustrates the importance of Theorem 5 where we plot the three principal components of different evolution of the states after learning G_T. The principal components of $(G_T^{(n)}(y_0, y_1))$ for different randomly initialized values of (y_0, y_1) is plotted on the left. On the right is plotted the vectors (x_{n-1}, x_n) where (x_n) is the solution to the actual data, i.e. it satisfies the update equation $x_{n+1} = g(x_n, u_n)$. The Lorenz data from the main article ([**?**, Fig. **??**B]) was used in the update equation and learn G_T.

Intuitively from Fig. **??**, if the diameter of the set Y_T gets larger, the diameter of $X_{U^+} \times X_{U^+}$ gets larger as well. A larger diameter of $X_{U^+} \times X_{U^+}$ would ensure that the subset indicated in yellow in Fig. **??** would shrink in size since the set $X_{U^+} \times X_{U^+}$ is always contained in the compact set $X \times X$ when $g^+ : (U^+ \times X^+) \to X$. This prevents large errors during forecasting while trajectories slip out of it. To get the diameter of Y_T larger we must make $g(u, \cdot)$ not "too contractive" – recall the examples that $g(u, x) = ux/2$ would have Y_T to be a set with a single point $\{0\} \times \{0\}$. In the case of a RNN as in (3), a large diameter of Y_T can be realized by setting α that is large enough so that it also simultaneously satisfies the USP (see [**?**] for a detailed study). In summary, when we employ learning Γ instead of learning G_T, the commutativity as indicated in Fig. **??** holds. The reader may compare this with Fig. **??**. In the discussion above, we have not considered the effect of errors made while learning Γ. This would entirely depend on the functional complexity of the map and the sophisticated method to find it.

Figure was here

Figure was here

8 Delay Coordinates as Inputs

We next consider a more general problem of learning a dynamical system when only observations of an orbit are available. Explicitly, if (W, T) is a dynamical system with dynamics generated by $w_{n+1} = Tw_n$, and if $\theta : W \to \mathbb{R}$ is an observable, then the task is to learn a system that is topologically conjugate to (W, T) and predict $\theta(w_{m+1}), \theta(w_{m+2}), ...$ using the data $\theta(w_0), \theta(w_1), \ldots, \theta(w_m)$. Suppose the input from the delay-coordinate map $\Phi_{\theta,2d}(\theta(w_n)) := (\theta(w_{n-2d}), \ldots, \theta(w_{n-1}), \theta(w_n))$ is fed into the driven system as u_n and Takens delay embedding theorem (see Theorem 1) holds, in which case, there exists a homeomorphism $F_\theta : \Phi_{\theta,2d}(\theta(w_n)) \mapsto \Phi_{\theta,2d}(\theta(w_{n+1}))$. Hence if the input values $u_n := \Phi_{\theta,2d}(\theta(w_n))$ are fed to a driven system g that is SI-invertible and has USP, the induced dynamical system (Y_F, G_F) would be topologically conjugate to the inverse-limit space of $(\Phi_{\theta,2d}(W), F_\theta)$ due to Theorem 4, and one could forecast $u_n, u_{n+1}, \ldots$, and hence the values $\theta(w_n), \theta(w_{n+1}), \ldots$. The advantage of feeding delay-coordinates to a driven system is that the embedding is stable in the sense of global dissipativity that we have described in Theorem 5. From the Koopman operator standpoint, when $X \subset \mathbb{R}^N$, the component functions of H_2 are the set of observables of the inverse-limit system of $(\Phi_{\theta,2d}(W), F)$ on which the action of the Koopman operator can be learnt from data.

One of the perplexities of employing the delay coordinate map is that the required delay to embed the attractor is not known. In the case when the required delay is not known, we demonstrated appealing numerical results ([**?**, Fig. **??**D]) where the theory is based on a conjecture below.

Conjecture 1. *Let g be a driven system that is SI-invertible and has the USP. Let (W, T) be a dynamical system and $\theta : W \to \mathbb{R}$ be an observable so that Takens delay embedding theorem holds and the the component functions of H_2 can embed*

$$\Theta(\widehat{W}_T) := \{(\ldots, \theta(w_{-2}), \theta(w_{-1})) : w_{n+1} = Tw_n\}$$

in $X_U \times X_U$,. Then there exists a homeomorphism on $\widetilde{\mathcal{F}}_\theta$ on $\Theta(\widehat{W}_T)$ and a self-map $G_{\Theta,T}$ on $H_2(\Theta(\widehat{W}_T))$ so that the following diagram commutes:

(21)

If the conjecture is true then one can learn the single-delay lag dynamics $G_{\Theta,T}$ by feeding $\theta(w_0), \theta(w_1), \ldots, \theta(w_m)$ into the driven system g as before to forecast $\theta(w_m), \theta(w_{m+11}), \ldots,$. The premise as to why the conjecture can be true can explained as follows. If Takens embedding theorem for the system (W, T) and the observable θ, then it easily follows that there is a homeomorphism

$$\widetilde{\mathcal{F}}_\theta := (\ldots, \theta(w_{k-2}), \theta(w_{k-1})) \mapsto (\ldots, \theta(w_{k-1}), \theta(w_k)).$$

Next, it can be shown that the component functions of H_2 are also sufficiently smooth [**?**, Theorem III.1] when g has the USP and is sufficiently smooth. If U is a smooth manifold of dimension M, and there are at least $2M+1$ component functions of H_2 that are generic in the sense of Whitney's embedding theorem – then H_2 would embed $\Theta(\widehat{W}_T)$ in $X_U \times X_U$. In this case, $G_{\Theta,T} : (x_{n-1}, x_n) \mapsto (x_n, x_{n+1})$ exists and is given by $G_{\Theta,T} = H_2 \circ \widetilde{\mathcal{F}}_\theta \circ H_2^{-1}$. However, it is an open question whether there are enough independent component functions in H_2 that are generic in the sense of Whitney so that H_2 embeds $\Theta(\widehat{W}_T)$. The only result the authors are currently aware of is that there are enough independent functions in the universal semi-conjugacy h (actually h restricted to $\Theta(\widehat{W}_T)$) that makes h a limit point of embeddings in the sense of Whitney [**?**, Corollary 2.3.2]; h true turns out to be the so-called echo state map in [**?**].

9 Methods and Forecasting Results

We demonstrate numerical results after learning the map Γ and using equations (16)–(17) for forecasting. We realize g through a RNN of the type (3) with the causal embedding property. Regardless of how the map Γ is learnt and implemented, we call a system with a RNN that implements g in (16)–(17) as a *recurrent conjugate network* (RCN) in view of G_T being conjugate/semi-conjugate to the inverse-limit system as in (15). A schematic of a RCN is shown in Fig. **??**. Throughout, the matrices A and B in the RNN are randomly initialized, and in particular, we set the spectral radius of the matrix αB to be α by using a matrix B with spectral radius 1. For α sufficiently small, usually in $(0, 1)$, RNNs have the USP and this can also be verified empirically (using the parameter-stability plot in [**?**]) when needed. Unless mentioned otherwise, we retain the same parameter values $a = 0.5$ and $\alpha = 0.99$ as in [**?**] in the experiments. We describe the systems employed in [**?**] for the experiments in detail and also supplement them with more numerical results.

Figure was here

In our RCN implementation, we learn $\Gamma : (x_n, x_{n-1}) \mapsto u_n$ by first obtaining the principal components of the states x_n and then learning a feedforward network (NN) on these principal components with the target u_n. The principal components are used only for an efficient state representation possibly to reduce the errors while learning and is not for a lossy approximation since all principal components are used. To be explicit, we denote the matrix with the first N states of the network data as row vectors by $X_{1:N}$. If $X_{1:N} = U\Sigma P^T$ denotes the singular value decomposition of $X_{1:N}$, then the principal component matrix is given by P and the principal components are given by $Z_{1:N} = X_{1:N}P$. These principal components are used to train a feedforward

neural network. If we denote the row vectors of $Z_{1:N}$ by $z_i^T, i = 1, 2, \cdots, N$ and the neural network by NN, then we learn an approximation of the map $NN : (z_{n-1}, z_n) \mapsto u_n$. We use the learnt approximation of NN to approximate Γ since

$$NN\left(\begin{bmatrix} P^T x_{n-1} \\ P^T x_n \end{bmatrix}\right) = NN \circ \begin{bmatrix} P^T & 0 \\ 0 & P^T \end{bmatrix} \begin{bmatrix} x_{n-1} \\ x_n \end{bmatrix} = u_n = \Gamma(x_{n-1}, x_n).$$

The feedforward neural network is implemented in *Python* using the *Keras* [**?**] library built on *Tensorflow*. Throughout our experiments, the network is constructed with 12 hidden layers with a layer dimension equal to 128. The activation function on the hidden layers is the $ReLU$ function built into *Keras*, whereas the output layer has a *tanh* output. Note that the output of the *tanh* activation function lies within $(-1, 1)$, so before training, the sequence (u_n) needs to be re-scaled as to fit inside $(-1, 1)$. We typically accomplish this by first subtracting the mean $\overline{u}$, and then multiplying the input by a scalar so that the input range is inside $[-0.5, 0.5]$ so as to keep the range of (u_n) well within the bounds of the *tanh* function.

Training is accomplished using the *Adam optimizer*, optimizing the *Mean Squared Error* loss function. We train the network three times, using three different learning rates for the *Adam optimizer* equal to 0.001,0.0001, and 0.00001. Each time we learn Γ, we use 150 training epochs and a batch size of 128.

Chaotic systems form an abstraction of complex behaviors in large-dimensional systems. Hence, chaotic systems provide an amiable set of toy examples on which to test our forecasting theory behind causal embedding. We now describe the chaotic system used in the numerical simulations in the main article [**?**]. The well-known system of ordinary differential equations developed by Edward Lorenz:

$$\begin{aligned} \dot{x} &= \sigma(x - y) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z, \end{aligned} \tag{22}$$

with parameters $\sigma = 10, \beta = 8/3$ and $\rho = 28$ exhibits chaos. We sample the flow of this Lorenz system by using numerical integration techniques (implemented using the *SciPy* library's ODE solver [**?**]), and in our experiment we use a step size of 0.1 between successive samples. Next, we consider the discrete time dynamical system driven by the (full) logistic map, given by:

$$v_{n+1} = 4v_n(1 - v_n). \tag{23}$$

Note that the logistic map is not invertible, which means the theory (Theorem 4) only guarantees a semi-conjugacy. For chaotic systems, even negligible errors in estimating Γ would make it hard to minimize the long-term point-wise prediction

Time-steps	1000	2000	5000	10000	20000	50000
(w_n) is the Lorenz states, sampled every 0.1 timestamps. $u_n = \frac{1}{100} w_n$						
x-coordinate	0.0088	0.0144	0.0211	0.0329	0.0452	0.0686
y-coordinate	0.0088	0.0144	0.0211	0.0329	0.0452	0.0686
z-coordinate	0.0022	0.0065	0.0136	0.0204	0.0285	0.0715

Table 2: Wasserstein distances between each coordinate of the forecasted and the actual data. The table illustrates long-term consistency - the data point distribution remains close even after 50,000 prediction steps. Furthermore, it showcases the learnt attractor being close to the actual one.

error. However long-term topological consistency can be achieved: the orbit of the prediction lies on the attractor.

The behavior of deterministic dynamical systems can alternatively be described macroscopically (e.g., [**?**]). This study involves describing how the density of an ensemble of initial conditions evolves upon iterating the map on such an ensemble. A useful tool is the Perron Frobenius operator that determines the evolution of an initial density [**?**]. An important concept in studying stochastic dynamics is the notion of an invariant density p that is a fixed point of the Perron Frobenius operator and when it exists it can determine the visitation frequency of typical orbits to any (measurable) set. We refer the reader to [**?**] for more details. The accuracy of the reconstructed or forecasted data relies on how the statistical properties like the invariant density are retained by the forecasted data. In the main article, we exhibited invariant densities of the logistic map [**?**, Fig. **??**]. The invariant density of the full logistic map on $[0, 1]$ is found to be $\frac{1}{\pi\sqrt{x(1-x)}}$ (e.g., [**?**].). There exist arbitrary small perturbations of the map T under which the invariant density is much different than that of the invariant density of the logistic map, i.e., there is a non-smooth change in the invariant density. In more technical terms the map T lacks the linear response [**?**]. Despite this statistical instability, our forecasting results from the RCNs show greater numerical accuracy while the invariant density was simulated (see Fig. **??**E). Wherever we cannot graphically illustrate the reconstructed invariant density of a higher dimensional system like the Lorenz system, we tabulate the Wasserstein distance (Table 2) between the numerically found densities of the actual and predicted. In this case, we use the 1-Wasserstein distance [**?**] which is a metric between one-dimensional probability distributions, implemented programmatically through the *SciPy* library [**?**]. In addition, we exhibit the cumulative distribution functions (Figure **??**) of the various coordinates of the numerical experiment in [**?**, Fig. **??**B].

Figure was here

We next illustrate the forecasting results obtained by feeding the delay-coordinates of an observation of (22) in Figure **??**. We consider observations $\theta(w_0), \theta(w_1), \ldots, \theta(w_m)$ from a dynamical system determined by sampling the flow of the Lorenz system (W, T), and where $\theta : W \to \mathbb{R}$ is an observation – observation we use is the x-coordinate of (22). The delay-coordinates $\Phi_{\theta,2d}(\theta(w_n)) := (\theta(w_{n-2d}), \ldots, \theta(w_{n-1}), \theta(w_n))$ is fed into the driven system as u_n and the attractor is learnt by learning a map $(x_{n-1}, x_n) \mapsto w_n$. The original and reconstructed attractor for the Lorenz system is shown in Fig. **??**. To obtain this result, data (w_n) is generated from the Lorenz system, sampled every 0.1 time-steps. The input into the network is the 10-delayed first coordinate of (w_n), scaled down to fit inside $[-0.5, 0.5]$. Specifically $u_n = \frac{1}{100}(w_x^{(n-9)}, w_x^{(n-8)}, \cdots, w_x^{(n)})$. Training of Γ was accomplished using 2000 data points, after discarding the first 500 to allow the RNN to forget its initial state.

Figure was here

One of the great challenges in learning dynamics comes from those systems where the present is greatly sensitive to the distant past much more than the immediate past, that is the past does not pass away. Examples of systems that show intermittency occurs in dynamical systems whenever the system appears to switch back and forth between two qualitatively different behaviors. We consider a type of intermittency (type-I intermittency) that has been found difficult to forecast, one in which the orbits slowly drift from a fixed/periodic point and then move to a chaotic regime but then return to the slowly drifting neighborhood of the fixed point. The time spent by the orbit in a neighborhood of the fixed/periodic point depends rather sensitively on how closely the system entered its vicinity and the length of this slow drifting phase is unpredictable. Such intermittency is a feature of transition to turbulence and convection in confined spaces [**?**], earthquake occurence [**?**], and anamolous diffusion in biology [**?**]. We consider an invertible map of type-I intermittency in the constant-J sub-families of the Hènon maps: [**?**]

$$w_{n+1} = \begin{bmatrix} w_{n,y} \\ -Jw_{n,x} + \mu + w_{n,y}^2 \end{bmatrix}. \tag{24}$$

The parameters J and μ can be adjusted to alter the intermittent behavior, and in our case, we used $J = 0.015$ and $\mu = 1.76$ in our experiment in [**?**, Fig. **??**A,B,C]. We next consider the non-invertible Pomeau-Manneville family of maps [**?**] in which the intermittency is more pronounced:

$$w_{n+1} = \begin{cases} w_n(1 + 2^\gamma w_n^\gamma) & \text{if } w_n \leq 0.5 \\ 2w_n - 1 & \text{if } 0.5 < w_n, \end{cases} \tag{25}$$

where $0 < \gamma < 1$. A larger value of γ makes the on-off intermittent behaviour more pronounced, and in the main article we had employed $\gamma = 0.6$.

The change in qualitative behavior as the parameter γ in the Pomeau-Manneville

map is tuned is illustrated in Fig. **??**. In the main article, we opted to keep the hyperparameters for our choice of RNN in (3) constant. We note that we are able to obtain results for maps with stronger intermittency by increasing the parameter α in the RNN slightly greater than 1 in (3). On an intuitive level, this makes the size (diameter) of Y_T^+ larger. With $\alpha = 1.2$ instead of $\alpha = 0.99$ in [**?**], we can obtain both topological and statistical consistency while forecasting for even more pronounced intermittency.

Authors in [**?**] have called the behavior exhibited by such intermittent maps as sporadicity that "fills in a gap between multi-periodic and chaotic dynamical behaviors or equivalently between predictable and random patterns". Approximations of the Koopman operators of such dynamical systems or feedforward neural network-based approaches are amnesiacs regarding their distant past. We have demonstrated that RCNs can reconstruct the attractor of a map in the Pomeau-Manneville family [**?**, Fig. **??**D]. This can be attributed to the fact that RCNs can remember nostalgically the distant past as we consider observables of the inverse-limit systems.

Figure was here

We also show the effectiveness of our methods to forecast physical data of temperatures across South Africa that shows variations on multiple time scales. Data considered for forecasting is the real-world weather data observed at weather stations and then interpolated using physical models to allow for finer detail. In particular, the data is the *ERA5 hourly data on single levels collected from 1979* available from *the Copernicus Climate Change Service (C3S) Climate Data Store (CDS)* [**?**], and we use the temperature 2 meters above the earth's surface and mean sea level pressure data sampled every 6 hours. The data takes the form of a grid of 65×105 data points, where grid points refer to a location in South Africa. Due to computational constraints, we scale down the resolution by only considering the 5th data point in each dimension, leaving a grid of size 13×21 (see top-panel Fig. **??**). Note that associated with each grid point, there are two numbers: the temperature (in K), and the Pressure (in Pa). For each time-step, we unroll the grid into two vectors of length 173 ($= 13 \times 21$) each. After normalizing each vector by subtracting the mean and scaling down, we stack these vectors atop each other. This final vector of length 546 is fed as input to the network.

As training, we use 26,000 data points, the first 1000 of which are discarded to allow the RCN to forget its initial state values. At a 6-hour sample rate, this translates to training on around 12 year's worth of climate data from late 2005 to 2017. We then use the trained network to predict 5,000 time-steps into the future, which is roughly 3 years, ending in April 2021. Fig. **??** illustrates the result of the prediction of the temperatures (which is of interest to us). As can be seen, even though the short-term prediction quickly fails (bottom panel of Fig. **??**), long-term characteristics

of the data set have been modeled with the seasonal variation in temperature of the predicted and actual data clearly visible (middle panel of Fig. **??**). This is despite using only one in five samples from the dataset in our experiment due to computational constraints.

Figure was here

10 Conclusions

Finding a set of observables of data that determine a less functionally complex learnable map so that its dynamics in the observed space gets closer to that of the action of the Koopman operator on the observables has been pursued by many researchers for the last decade. In practice, however, the Koopman operators of complex and chaotic systems tend to have significantly more complicated spectral properties (e.g., non-isolated eigenvalues and/or continuous spectra) hindering the performance of data-driven approximation techniques that capture only a portion of the spectrum (e.g., [**?**, **?**]). Also, in the temporal domain, approximations of the Koopman operators of the underlying dynamical systems are amnesiacs regarding their distant past. Here, we show a driven dynamical system that can causally embed dynamical systems is capable of determining a set of observables of the inverse-limit system of the underlying dynamical system and learning the Koopman operator rather than learning an approximation of it. In particular, we produce a topological conjugacy (semi-conjugacy) between the single-delay lag dynamics of the driven system's data and the underlying homeomorphism (non-invertible map that can be discontinuous) dynamical system.

This methodology induces equations from data. In particular, we have shown that for data arising from a self-map on a compact metric space, a first-order system of difference equations with auxiliary variables or a single difference equation of infinite order can be obtained. The states of the driven system are affected to an arbitrarily small extent by that left-infinite segment of the past of the input beyond some arbitrarily large but finite time thanks to the continuity of the universal semi-conjugacy of the driven system. As a consequence finite length of input data is adequate for forecasting. Empirically it is found that the map with the single-delay lag dynamics has a stronger linear relationship and intuitively less functional complexity than the map that describes the data or the map the delay coordinates induce. As a consequence, through the use of recurrent conjugate networks (RCNs), we obtain exceptional forecasting results with long-term topological and statistical consistency which is illustrated on chaotic maps and data showing intermittency. The robustness of forecasting against external noise is also observed.

The advantage over other known data-driven approaches is not only long-term consistency, but it does not need expert human intuition or physical insights to decide on observables and library functions that are commonly employed in data-driven approaches. This is very useful in high-dimensional forecasting tasks where such insights are rare. The performance of such data-driven approaches is affected in the absence of such physical precognition, and when one overcomes it through combining delay-embedding techniques still exact reconstruction is lacking. Hence, whenever high-fidelity models are needed, the RCNs out-perform such data-driven methods considering that the step size of a discretization can be even up to **100 times** than that have been used in some algorithms like SINDy and other echo state network methods. Lastly, and remarkably, we do not have to resort to finding new observables or guessing the library of governing equations for new temporal datasets as the observables induced by the driven systems are universal.

From the perspective of the reservoir computing approaches, we show the existence of a learnable map in a RCN if single-delay lag dynamics is chosen. Although learning a function that the single-delay dynamics entails is more computationally expensive than performing a simple linear regression as in a customary echo state network training, we overcome extensive ad hoc adaptations, like introducing feedback conditions and other parameters, which make the art of training in reservoir computing methods a craft rather than a science. The driven system in the RCN has proven properties of robustness to perturbations of the input and parameters in the system, making these networks attractive for hardware implementations. The theory of causally embedding a dynamical system is general and hence does not depend on a particular choice of the driven system. A future exploration into designing more sophisticated driven dynamical systems to reduce the functional complexity of the map that the single-delay dynamics entails could help in furthering the accuracy that we have obtained with a RCN.

Data availability. All the data used is either computer-generatable or obtained from publicly accessible sources.

Code availability. The Python/Matlab code employed for all the simulations in the supplement is also submitted.