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

Abstract

Obtaining repeated measurements through observables of underlying physical and natural systems to build dynamical models is engrained in modern science. A key to the success of such methods is that the dynamics in the observed space can often be described by a map that has much lower functional complexity than the one that describes the unknown underlying system. Finding observables that can empirically reduce the functional complexity of the map to be learned, and at the same time, theoretically guarantee exact reconstruction in the new phase space is an open challenge. Here, we determine a set of observables for the Koopman operator of the inverse-limit system of a dynamical system that guarantees exact reconstruction of the underlying dynamical system. Similar to the delay coordinate maps being universal observables in Takens delay embedding, the observables we determine are universal, and hence do not need to be changed while the underlying system is changed. They are determined by a class of driven systems that are comparable to those used in reservoir computing, but which also can causally embed a dynamical system, a phenomenon which we newly describe. Dynamics in the observed space is then shown to be topologically conjugate to the underlying system. Deep learning methods can be used to learn accurate equations from data as a consequence of the topological conjugacy. Besides stability, amenability for hardware implementations, causal embedding-based models provide long-term consistency even for systems that have failed with previously reported data-driven or machine learning methods.

Keywords Data-driven dynamical systems, Koopman Operator, Causal Embedding, Nonautonomous Dynamical Systems.

Introduction

Since Poincare’s pioneering work, we have had over a hundred years of the profound theory of dynamical systems that allows analyzing tractable models. With the advancement of technology,
our ability to record rich and complex data, and forecast it has increased profoundly. Often, machine learning algorithms can accomplish a prediction task rather than build a model to explain the dynamics behind the data, and hence have poor long-term consistency. From a philosophical outlook of science, finding meeting points between rationalism (dynamical systems theory) and empiricism (machine learning) could yield important results in modeling. When observed data originates from a dynamical system, we solve a theoretical problem in data-driven forecasting that entails equations from data and exceptionally long-term consistent models.

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 just to minimize the point-wise prediction error. Although better forecasts do help better manage service interruptions and resource management, a model with long-term consistency concerning time-averaged characteristics is useful for understanding the response to changes in the statistical properties of the perturbations of models that have potentially distinguished applications, for instance in climate science (e.g., [1, 2] and references therein). Attempts made to build model equations from complex data such as sunspot cycles date back to the 1920s in [3], but a major breakthrough came through the Takens embedding theorem [1] that guaranteed an equivalent state-space reconstruction, a valuable tool for any kind of analysis. Often, experimental measurements of dynamical systems are not directly the system’s states, but a univariate time-series whose span is smaller than the underlying system’s dynamics. The Takens embedding theorem and their various generalizations (e.g., [5, 6, 7, 8, 8]), under some generic conditions, establish the learnability of a system that is created out of concatenating sufficiently large previous observations of a dynamical system into a vector (called delay coordinates). The system determined by such vectors is equivalent (topologically conjugate) (e.g., [9]) to the system from which the observed time-series was derived. While topological conjugacy ensures an alternate representation of the underlying system, the quality of the representation remains sensitive to various parameters making learning the dynamics unreliable (e.g., [10]). One of the reasons for this fragility is that the embedded attractor in the reconstruction space is, without doubt, an invariant set, but not necessarily an attractor. For learning the map on the invariant set, it is desirable to have locally conservative but globally dissipative dynamics i.e., there is an attractor containing the invariant set so that small errors do not lead to the future iterates completely veering off from the invariant set before predictions totally fail [11, Fig. 1 and Fig. 2]. Theoretical conditions under which the geometry of the attractor can be preserved (e.g., [12]) 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.

Recently, machine learning algorithms like reservoir computing [13, 14] and data-driven algorithms based on Koopman Operator (e.g., [15, 16]) have shown greater quantitative accuracy in forecasting dynamical systems. While the reservoir computing methods fail to establish a learnable map often yielding poor consistency [17, 18, 19, 20], the Koopman theory and other data-driven based approaches suffer from either lack of determining the right observables [21, 22] or demand full knowledge of the underlying state space variables [23, 24] that govern the underlying system or data obtained at high sampling-rates. Attempts of combining principles of such data-driven methods with delay embedding [25, 26] are yet to theoretically guarantee exact reconstruction.
Since studies are suggestive that with large enough time-lagged observations \cite{12,27} one gets closer to isometric embeddings, here we determine observables for the Koopman operator of the inverse-limit system \cite{28} instead of the actual system. Remarkably, such observables need not be changed when the underlying dynamical system is changed. The methodology provides exact reconstruction (or finite-faithful representation \cite{29}), global dissipativity while forecasting, and equations from data. We demonstrate long-term topological consistency (attractor learning) 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 intermittent systems exhibit bursts, and signify great sensitivity to the distant past much more than the immediate past and is a feature of transition to turbulence and convection in confined spaces \cite{30}, seismic data \cite{31} and anomalous diffusion in biology \cite{32}, and are extremely difficult to model from data. Our approach is to learn the action of the Koopman operator on observables of the inverse-limit system of the underlying dynamical system so that the observables depend on past history. We determine these observables by a specific kind of driven dynamical system and using the given data from the underlying system.

Causal Embedding and Learning the Koopman Operator

A dynamical system in this work is a tuple $(U, T)$ where $U$ is a compact metric space and $T : U \rightarrow U$ is a surjective function (that is not necessarily continuous). In fact, if $A$ is any invariant set, i.e., if $T(A) := \bigcup_{u \in A} Tu = A$, 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 a bi-infinite 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/trajectory of $T$, and since $T$ is surjective there is at least one orbit through each point in $U$. Throughout, we say that a space $Z$ is embedded in $X$ if there exists $Y \subset X$ and a function $f$ so that $f : Z \rightarrow Y$ is a homeomorphism.

The problem of forecasting a dynamical system involves predicting $u_{m+1}, u_{m+2}, \ldots$ given the finite segment of an orbit $u_0, u_1, \ldots, u_m$ of an unknown $T$. By a problem of learning a dynamical system from data, we mean learning model equations that can be used to regenerate and forecast the data. The model equations could involve other hidden or auxiliary variables. 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 $\phi : U \rightarrow V$ with the property that $\phi \circ T = S \circ \phi$. Such a map $\phi$ is called a conjugacy and we say that $(V, S)$ is conjugate to $(U, T)$ or simply $S$ is conjugate to $T$. If we relax the condition on $\phi$ where, instead of having a homeomorphism, we only require $\phi$ to be continuous, then we call $\phi$ a semi-conjugacy, and say that $S$ is semi-conjugate to $T$. When $S$ is conjugate to $T$ it means that there is a one-to-one correspondence (e.g., \cite{9}) in the dynamics between the two systems, whereas when $S$ is semi-conjugate to $T$ with $\phi$ a many-to-one mapping then $(V, S)$ provides a coarse-grained description of the $(U, T)$. When $S$ is semi-conjugate to $T$ then it is customary to say that $S$ is a factor of $T$ or that $T$ is an extension of $S$. In essence, an extension is a larger system that captures all the important dynamics of its factor (e.g., \cite{9}).
Given a segment of an orbit \( (U, T) \), our approach is to build model equations of a system that is topologically conjugate to an extension of \( (U, T) \). To define such an extension, we first let the left-infinite countable cartesian product of \( U \) be denoted by \( \hat{U} := \cdots \times U \times U \) (and equip it with the product topology). Any dynamical system \( (U, T) \) (since we assume \( T \) is surjective) determines a nonempty subspace \( \hat{U}_T \) of \( \hat{U} \) comprising the left-infinite orbits, i.e.,

\[
\hat{U}_T := \{(\ldots, u_{-2}, u_{-1}) : Tu_n = u_{n+1}\},
\]

and \( \hat{U}_T \) is called the inverse-limit space of \( (U, T) \) in the literature (e.g., [28]); in the literature, the elements of \( \hat{U}_T \) are rather written as right-infinite sequences. The map \( T \) also induces a self-map \( \hat{T} \) on \( \hat{U}_T \) defined by \( \hat{T} : (\ldots, u_{-2}, u_{-1}) \mapsto (\ldots, u_{-2}, u_{-1}, T(u_{-1})) \). It is straightforward (see [11, Eqn. 11]) to find that the dynamical system \( (U, T) \) is semi-conjugate to \( (\hat{U}_T, \hat{T}) \) and whenever, \( (U, T) \) is invertible, \( (\hat{U}_T, \hat{T}) \) is conjugate to \( (U, T) \).

Towards the end of constructing a dynamical system conjugate to \( (\hat{U}_T, \hat{T}) \), we consider the notion of a driven (dynamical) system. A driven system comprises of two compact metric spaces \( U \) and \( X \), and a continuous function \( g : U \times X \to X \). We call \( U \) and \( X \), respectively the input and state space of the driven system. For brevity, we refer to \( g \) as a driven system with all underlying entities quietly understood. Since \( X \) is assumed to be compact, it follows (e.g., [33, 34, 35]) that for every chosen (bi-infinite) input \( \vec{u} = \{u_n\} \subset U \) there exists at least one sequence \( \{x_n\} \) that satisfies \( x_{n+1} = g(u_n, x_n) \) for all \( n \in \mathbb{Z} \). Any such bi-infinite sequence \( \{x_n\} \) is called a solution of \( g \) for the input \( \vec{u} \). In particular, if we denote a bi-infinite sequence \( \{u_n\}_{n \in \mathbb{Z}} \subset U \) by \( \vec{u} \), and a left-infinite part by \( \vec{u}^n := (\ldots, u_{n-2}, u_{n-1}) \), we note that \( \vec{u}^n \) belongs to \( \hat{U} \) regardless of \( n \in \mathbb{Z} \).

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

\[
X_U := \{x \in X : x = x_k \text{ where } \{x_n\} \text{ is a solution for } \vec{u}\}.
\]

For example, when \( U = [0, 1] \) and \( X = [0, 1] \), for the driven system \( g(u, x) := \frac{u x}{2} \) regardless of any sequence in \( U \), \( x_n \equiv 0 \) for \( n \in \mathbb{Z} \) is the only solution of \( g \), and hence the reachable set \( X_U = \{0\} \). In this case, it is not possible to relate (this will be made precise later) the temporal variation in the input to the reachable set because it is singleton subset. We would like the reachable sets of driven systems to be such that the inverse-limit space \( \hat{U}_T \) of \( (U, T) \) can be topologically embedded into the reachable set or embedded into a finite self-product of the reachable set. To achieve such a desired embedding, we consider driven systems that are also state-input invertible (SI-invertible) that we define next. We say \( g \) is SI-invertible if \( g(\cdot, x) : U \to X \) is invertible for all \( x \), or equivalently if \( x_n \) and the \( x_{n+1} \) are related by \( x_{n+1} = g(u_n, x_n) \), then \( u_n \) can be uniquely determined from \( x_n \) and \( x_{n+1} \). A recurrent neural network (RNN) with \( N \) artificial neurons e.g., [36] is an example of \( g \) with \( U \subset \mathbb{R}^N \) and \( X = [-1, 1]^N \) (the cartesian product of \( N \) copies of \( [-1, 1] \)) defined by

\[
g(u, x) = (1 - a)x + atanh(Au + \alpha Bx), \tag{1}
\]

where \( A \) is a \( N \times N \) matrix that represents input connections to the neurons called the input matrix. The matrix \( B \) is also of dimension \( N \times N \) representing the strength of the interconnections
between the neurons (known as the reservoir matrix in reservoir computing), and \(a\) and \(\alpha\) are real-valued parameter customarily called the leak rate and scaling of the reservoir \(B\) respectively and \(\tanh(\cdot)\) is (the nonlinear activation) \(\tanh\) performed component-wise on \(\cdot\). The RNN accepts inputs as sequence of vectors with \(N\) elements, and it can be readily verified that when both \(A\) and \(B\) are invertible, then \(g\) has SI-invertibility (see [11 Eqn. 4]). We also note that if a real vector \(v_n\) is of dimension \(K < N\), SI-invertibility can still be realized by (isometrically) embedding \(v_n\) into \(\mathbb{R}^N\), for instance by padding \(N - K\) zeroes to obtain an input of dimension \(N\) i.e., \(v_n \mapsto (v_n^1, v_n^2, \ldots, v_n^K, 0, 0, \ldots 0) =: u_n\).

Given a driven system \(g\) that is SI-invertible and a dynamical system \((U, T)\), we define a relation on \(X_U\) (a relation on \(X_U\) is a subset defined on \(X_U \times X_U\)) by

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

and call \(Y_T\) relation induced by \((U, T)\). It easily follows [11 Theorem 3] that there exists a map \(G_T : Y_T \to Y_T\) defined by \((x_{n-1}, x_n) \mapsto (x_n, x_{n+1})\) that describes the the single-delay lag dynamics on the driven system’s states.

In general, there could be no well-founded relationship like a topological conjugacy or a semi-conjugacy between the maps \(T\) and \(G_T\) since the dynamics of the driven system could be more complex than in the input ([11 Fig. 3]). However, it is possible to identify a class of driven systems for which we can relate \(T\) and \(G_T\). First, we pause to ask a question: is the complexity of the solution of \(g\) exclusively determined by the complexity in the input? To make this precise, we first consider the mapping that defines the evolution of the input by appending a new input value \(\bar{u} = v \in U\) at time \(n\), \(\sigma_v : \bar{u}^n \mapsto \bar{u}^n v\), where symbolically \(\bar{u}^n v := (\ldots, u_{n-2}, u_{n-1}, v)\), and then ask if the dynamics of the driven system is topologically semi-conjugate to \(\sigma_v\)? This is equivalent to asking if there exists a continuous surjective map \(h : \overset{\sim}{U} \to X_U\) so that the following diagram commutes:

\[
\begin{array}{ccc}
\overset{\sim}{U} & \xrightarrow{\sigma_v} & \overset{\sim}{U} \\
\downarrow h & & \downarrow h \\
X_U & \xrightarrow{g(\bar{u}^n)} & X_U.
\end{array}
\]

If the driven system \(g\) is such that the above commutativity holds then we call \(h\) a universal semi-conjugacy. Remarkably, \(g\) satisfies the above commutativity (see [11 Theorem 2]) when \(g\) satisfies the unique solution property: we 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 \(\Psi\) so that \(\Psi(\bar{u})\) denotes the unique solution obtained from the input \(\bar{u}\). Also in this context, the USP is equivalent to saying \(g\) is a topological contraction [11 Eqn. 6], and this notion is independent of SI-invertibility. This topological contraction is often also called the echo state property [37], and systems with this property have other important useful properties [38, 39] for information processing. Attempts have been made in the literature to algorithmically construct a semi-conjugacy between the driven dynamics in the special case of periodic inputs [40], but we remark that we are considering a very general problem by considering all possible left-infinite sequences in [2]. For more illumination, we recall that the diagram in (2) commutes if and only if the system \(g\) has the USP (see [11 Lemma 5]). Furthermore, when (2) holds, \(h(\bar{u}^k) = x_k\), where \(x_k\) is the value of the solution at the \(k^{th}\) instant for any input \(\bar{u}\) whose left-infinite segment is \(\bar{u}^k\).
With inputs to \( g \) being restricted to be orbits of \( T \), in the context of left-infinite spaces as in [2], the system on the top in [2] turns out to be the inverse-limit system \( (\hat{U}_T, \hat{T}) \). This is since if we restrict the map \( \sigma_v \) to \( \hat{U}_T \), and select \( v = Tu_{-1} \), then we have \( \sigma_v(u_0, \ldots, u_{-2}, u_{-1}) = \sigma_{Tu_{-1}}(u_0, \ldots, u_{-3}, u_{-2}, u_{-1}) = \hat{T}(u_0, \ldots, u_{-3}, u_{-2}, u_{-1}) \). Next, we denote \( r : \hat{U} \to \hat{U} \) to be the right-shift map, i.e., \( r : (\cdot, u_{-2}, u_{-1}) \mapsto (\cdot, u_{-3}, u_{-2}) \). We then can establish that the two maps \( G_T \) and \( \hat{T} \) are related (a formal proof is in [11, Theorem 4]) through the function \( H_2(\bar{u}) := (h(\tau \bar{u}), h(\bar{u})) \) (strictly speaking, it is \( H_2 \) restricted to \( \hat{U}_T \)) by the following commutativity diagram printed in black:

\[
\begin{array}{ccc}
\hat{U}_T & \xrightarrow{T} & \hat{U}_T \\
\downarrow{H_2} & & \downarrow{H_2} \\
Y_T & \xrightarrow{G_T} & Y_T \\
\downarrow{(r_2, \pi_2)} & & \downarrow{(r_2, g)} \\
U \times X & & \\
\end{array}
\]

Note that \( H_2 \) maps an entire-left infinite sequence into an element in \( X \times X \). Specifically if \( \{x_n\} \) is the solution obtained with the input \( \{u_n\} \), then \( H_2(\bar{u}_k) = (x_{k-1}, x_k) \) for all \( k \in \mathbb{Z} \) since \( h(\tau \bar{u}_k) = x_k \).

To qualify \( H_2 \) further, 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 [2] commutes (and thus, [3] also commutes) (ii) \( H_2(\bar{u}) := (h(\tau \bar{u}), h(\bar{u})) \) embeds the inverse-limit space \( \hat{U}_T \) in \( X \times X \).

Now we list the consequences of our definitions and the commutativity in [3]. When \( g \) is SI-invertible and \( \{u_n\} \subset U \) is an orbit of an dynamical system \((U, T)\), the following results (formal proofs in [11, Theorem 3, Theorem 4]) hold: (i) there exists a map \( G_T : Y_T \to Y_T \) defined by \( (x_{n-1}, x_n) \mapsto (x_n, x_{n+1}) \) [11, Theorem 3] (ii) further, if \( g \) has the USP then \((Y_T, G_T)\) is topologically semi-conjugate to \((\hat{U}_T, \hat{T})\) (iii) furthermore, if \( T : U \to U \) is a homeomorphism, then \((Y_T, G_T)\) is topologically conjugate to \((\hat{U}_T, \hat{T})\) (and hence also conjugate to \((U, T)\)) and hence \( g \) causally embeds \((U, T)\).

Since two successive points \((x_{n-1}, x_n)\) of a solution of \( g \) with an input \( \bar{u} \) satisfy \( (x_{n-1}, x_n) = H_2(\bar{u}_n) \) and lie in \( Y_T \), one can learn the single-delay lag dynamics of the driven states through the map \( G_T : (x_{n-1}, x_n) \mapsto (x_n, x_{n+1}) \) from sufficiently large finite set of data points \((x_0, x_1), (x_1, x_2), \ldots, (x_{m-1}, x_m)\). Once \( G_T \) is learnt, then applying the iterates of \( G_T \), all future successive points on a solution can be forecasted. 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 [11, Eqn. 14]. In particular 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 \((\hat{U}_T, \hat{T})\). Synoptically, restricting \( H_2 \) to inverse-limit spaces of different dynamical systems contained in \( \hat{U} \) 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 [15] of any \((\hat{U}_T, \hat{T})\) as long as \( \hat{U}_T \) is contained in \( \hat{U} \).

Given a dynamical system \((U, T)\) and some vector space \( V \) of functions \( f \) whose domain is \( U \), the
(linear) operator $\mathcal{K} : V \rightarrow V$ so that $\mathcal{K} f = f \circ T$ holds is called the Koopman operator of $T$. If one knows the action of the Koopman operator on an observable $f$, then since $\mathcal{K} f(u) = f(T(u))$ and likewise $\mathcal{K} f(T(u)) = f(T(2)u)$, one can forecast the observed values $(f(Tx), f(T(2)u), f(T(3)u), \ldots)$, where $T^{(n)}$ denote the $n$-fold composition with itself. In Koopman’s theory (e.g., [15, 32, 16]) the dynamics of $(U, T)$ is inferred by the study of the action of the linear operator $\mathcal{K}$ on functions in $V$. It is customary to consider a class of observables where $V$ is some Hilbert space or some $L^p$ space, i.e., an observable $f$ represents an equivalence class of measurable functions. One of the central ideas in using Koopman’s theory for forecasting dynamical systems is to obtain a collection of observables $\mathcal{U}, T$ when it is a Hilbert space, and the span of the observables is invariant under $F$ map, say $L$.

For instance, $\mathcal{K}$ portion of the spectrum of $\mathcal{K}$ linear operator $\mathcal{K}$ is the Koopman operator of $\mathcal{K}$ operator of the inverse-limit system. Besides being able to learn the action of the Koopman operator exactly, our setup has other advantages over selecting or optimizing the choice of observables as in the Extended Dynamic Mode Decomposition (EDMD) algorithm (e.g., [22, 43]). We can alter the functional complexity of $G_T$ by changing the number of observables or equivalently by altering the dimension of $X$ – in the case of a RNN implementation of $g$, by simply changing the number of neurons in the network. Empirically, increasing the dimension of $X$ is found to increase the linear relationship (or intuitively reduces the functional complexity of $G_T$) that is measured as a generalization of the Pearson correlation coefficient to random vectors (e.g., [44]) between $(x_{n-1}, x_n)$ and $G_T(x_{n-1}, x_n)$, and such numerical evidence is tabulated in [11, Table 1]. In contrast, expanding the set of observables while using EDMD to learn a map with a lower functional complexity is a highly involved task, not least because so much that one often does not know how to guess a new observable, as well as the fact that adding observables does not necessarily retain the invariance of the span of the observables under the Koopman operator that is needed [22, 43] to capture a portion of the spectrum of the operator. Also, it is a fact that the spectrum of the Koopman operator of conjugate systems are identical while the spectrum of the Koopman operator of a factor is contained in its extension (e.g. [12]). These spectra are equal when $T$ is a homeomorphism. In any case the spectrum of the Koopman operator of $G_T$ contains that of the Koopman operator of $T$, and thus we have presented a methodology that avoids capturing only a portion of the spectrum of $T$ for forecasting.
Equations from Data and Forecasting

We learn single-delay lag dynamics defined by $G_T$ indirectly by learning $\Gamma : (x_{k-1}, x_k) \mapsto u_k$ – the map $\Gamma$ always exists when $G_T$ exists (see [3] and [11 Theorem 3]). The map $G_T$ entails another map (see the vertical red line and the diagonal line in red in that order in Eqn. (3) or in Fig. 1A), $S : (u_k, x_k) \mapsto (u_{k+1}, x_{k+1})$ defined by

\[
\begin{align*}
    u_{k+1} &= \pi_1 \circ (\pi_2) \circ (\pi_2, g)(u_k, x_k) \quad (4) \\
    x_{k+1} &= \pi_2 \circ (\pi_2) \circ (\pi_2, g)(u_k, x_k), \quad (5)
\end{align*}
\]

where $\pi_i : (a_1, a_2) \mapsto a_i$. The expressions (4)-(5) are equations constructed from data! Although the equations are coupled, (4) per se represents (possibly nonlinear) difference equation of infinite order since we can replace $x_k$ by $h(\hat{u}^k)$, where $h$ is the universal semi-conjugacy. While realizing (4)-(5) in practice, $x_k$ acts as a vector with several auxiliary variables as its components.

Learning the map $\Gamma$ instead of $G_T$ not only saves computational resources since $u_n$ lies in a low-dimensional subspace due to the zero-padding, but owing to the USP of the driven system $g$ one gains input-related, parameter-related stability [39], and importantly it prevents large numerical errors due to input noise due to global dissipativity ([11 Theorem 5]) that is not guaranteed for learning $G_T$ (see [11 Section 7] for details). Also, in practice, one does not need the entire-left infinite history of the input; by initializing the driven system with an arbitrary initial value $y_m \in X$, then the sequence $y_{m+1}, y_{m+2}, y_{m+3}, \ldots$ satisfying $y_{k+1} = g(u_k, y_k)$ for $k \geq m$ approximates the corresponding elements of the actual solution $\{x_n\}$ uniformly thanks to the uniform attraction property (see [11, Theorem 1] or [11, Eqn. 18]).

When only observations $\theta(w_0), \theta(w_1), \ldots, \theta(w_m)$ from a dynamical system $(W, T)$ where $\theta : W \rightarrow \mathbb{R}$ is an observable are available, 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 $u_n$. Assuming $(W, T)$ and $\theta$ are such that Takens embedding holds, i.e., there exists a map $F : \Phi_{\theta,2d}(\theta(w_n)) \mapsto \Phi_{\theta,2d}(\theta(w_{n+1}))$, then the single-delay lag dynamics obtained through $g$ is topologically conjugate to the inverse-limit system $(\Phi_{\theta,2d}(W), F)$, and forecasting can be carried out as before. The dimension of the input-vector is increased since delay-coordinates are fed into $g$, but the major gain is global dissipativity [11, Fig. 6].

One of the perplexities of employing the delay coordinate map is that the required delay to embed the attractor is not known immediately. In this case, by feeding the observations directly into the same driven system $g$ instead of the delay coordinates, ceteris paribus, we demonstrate appealing numerical results where the theory is based on a conjecture (see [11 Conjecture 1]). When $H_2$ has at least $2m + 1$ generic observables so that Whitney’s embedding theorem holds and $H_2$ embeds $\Theta(W_T) := \{(\ldots, \theta(w_{-2}), \theta(w_{-1})) : w_{n+1} = Tw_n\}$ in $X \times X$ then the dynamics of the single-delay lag dynamics becomes conjugate to that of $T$ and forecasting can be carried out as before (see [11 Section 8] for more details).
Methods and Forecasting Results

We realize $g$ through a RNN of the type (1) that is also SI-invertible and has the USP. Such RNNs fall into the reservoir computing framework and are amenable for hardware implementations (e.g., [45, 46, 47]). We call a forecasting system described by (4)–(5) a recurrent conjugate network (RCN), regardless of how the map $\Gamma$ is implemented as long as $g$ is a RNN. A schematic of a RCN is shown in [11, Fig. 8]. We set the spectral radius of the matrix $\alpha B$ to be $\alpha$ by using a matrix $B$ with spectral radius 1. For $\alpha$ sufficiently small, and usually in $(0, 1)$, RNNs have the USP that can be verified empirically (e.g., the parameter-stability plot in [39]).

In our forecasting experiments using RCNs, we learn $\Gamma : (x_{n-1}, x_n) \mapsto u_n$ by learning a feedforward network (NN) on the principal components of states $x_n$ of $g$. The principal components are optional and used only for an efficient state representation possibly to reduce the errors while learning, and is not intended for a lossy approximation since all principal components are used. To be explicit, we denote the matrix with the first $N$ states of $g$ 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

Fig. 1. Forecasting results using a RCN. In all experiments, matrices $A$ and $B$ of the RNN in Eqn. (1) were initialized randomly, with $B$ having unit spectral radius; without explicitly mentioning the input is always padded with the required number of zeroes to match the row number of $A$. We set the parameters $\alpha = 0.5$ and $\alpha = 0.99$ and a 1000 dimensional RNN, i.e. $X = [-1, 1]^{1000}$. Training to learn $\Gamma$ was accomplished with 2000 data points after 500 of these are discarded to allow the network to forget its initial state. A. The inner workings of forecasting. (B) Data $(w_n)$ is generated from the Lorenz system [11, Eqn. 22], sampled every 0.1 time-steps. The input into the network is scaled down to fit inside $[-1, 1]$, and perturbed with noise. Specifically, the input $(u_n)$ is given by $u_n = \frac{1}{100} w_n + \epsilon_n$ where $\epsilon_n$ is normally distributed with 0 mean and standard deviation equal to 0.01. This translates to a signal to noise ratio of roughly 18dB. 50,000 points of actual and predicted data.

(C) Two different epochs of the actual and forecasted time-series in (C). (D) Full states of the Lorenz system with the input in C is reconstructed by learning a map $\Gamma_{\text{full}} : (x_{n-1}, x_n) \mapsto w_n$. (E,F) Data $(w_n)$ is generated from the full Logistic Map [11, Eqn. 23], then normalized to have zero mean, and perturbed with noise. Specifically, the input $(u_n)$ is given by $u_n = w_n - \text{mean}(w) + \epsilon_n$, where $\epsilon_n$ is normally distributed of zero mean and standard deviation equal to 0.01. This translates to a signal-to-noise ratio of roughly 30dB. Plotted is the phase portrait in (E) and the smoothened invariant density $\mathcal{F}$ for actual (red) and predicted (blue) data, over 10000 prediction steps.
a feedforward neural network implemented in Python using the Keras library built on Tensorflow ([48]). Training is accomplished using the Adam optimizer, optimizing the mean squared error loss function. 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 $NN : (z_{n-1}, z_n) \mapsto u_n$. Hence $\Gamma$ that we learn is composed as

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

The dynamical systems chosen are a Lorenz system that exhibits turbulence, the full logistic map, a chaotic system without statistical stability as described by the linear response (e.g., [49]), a map in the Hénon family that exhibits intermittency and chaos [50], a map in the Pomeau-Manneville family [30] that exhibits weak chaos and intermittency. Intermittency is a feature of transition to turbulence and convection in confined spaces [30], earthquake occurrence [31], and anomalous diffusion in biology [32]. We note that the logistic map and the maps from the Pomeau-Manneville are both non-invertible, and hence the single-delay lag dynamics of the RCN could possibly be just semi-conjugate to the inverse-limit space. For the precise description of these dynamical systems we refer to [11, Eqns. 22, 23, 24, 25]. Even negligible deviations due to computational noise would make it hard to minimize the long-term pointwise prediction error for such chaotic systems, but long-term topological consistency, that is the requirement that the orbit lies on the attractor can be achieved. The ability to reconstruct or forecast a dynamical system also relies on how the statistical properties like an invariant density [51] are retained by the forecasted data.

We use the same driven system, a RNN in our RCN for all the experiments, and demonstrate the forecasting results in Fig. 1 and Fig. 2. A notable feature is that despite the very low sampling rate in Fig. 1B (step size of 0.1) in the discretization of the Lorenz system, the RCN predictions help reconstruct the attractor. To test our conjecture [11 Conjecture 1], we also feed a univariate time-series $\{\theta(w_n)\}$ obtained from an orbit $w_n$ of the Lorenz system, and observe that the hidden states in the entire-attractor can be recovered by training the states in $Y_T$ by $(x_{n-1}, x_n) \mapsto w_n$. We note that without relying on any such conjecture if the input is fed as delay-coordinates the attractor is learnt (see [11 Fig. 10]) as per the theory. In Fig. 1F, despite the logistic map not having the linear response [49], the forecasted invariant density shows a good resemblance to the actual invariant density. Since each node in a RNN receives a weighted sum of the external inputs and states as its input, heuristically such summing results in some noise cancellation, and it is found that the reconstructed attractor has lesser noise than the noisy attractor of the original. This for instance is evident in Fig. 2A. Obtaining models that capture the statistical behavior for the data from the Pomeau-Manneville map as illustrated in Fig. 2F is a hallmark of the methodology used here since the length of a slow drifting phase in the intermittency is unpredictable (details in [11 Section 9]), and to the best of our knowledge, no machine learning algorithm has succeeded so far on such maps. Also, we remark that we have not explored the parameters of the RCN that achieves the best forecasting results, and instead have used the same RNN to demonstrate the universality of the observables rendered by $H_2$. As an example to show that the results could be improved, we show that with different choices of $\alpha$ in the RCN we can forecast more pronounced on-off intermittency of maps in the Pomeau-Manneville family in [11 Section 9]. The interested reader may refer to the effectiveness of a RCN in accurately reproducing seasonal variations in the temperatures from physical data of the South African climate dynamics in [11 Section 9].
Fig. 2. More Forecasting results with the same RCN setup as in Fig. 1. However, to capture intermittency in noisy data, longer training data was used; 1000 samples were discarded to allow the network to forget its initial conditions, and 5000 were used for training. (A,B,C) Data \((w_n)\) comes from an intermittent realisation from the Henon family of maps [11, Eqn. 24]. Input \((u_n)\) was normalized to have mean zero, scaled to fit inside \([-1,1]^2\), and perturbed with noise.

To be explicit, \(u_n = \frac{1}{10}(w_n - \text{mean}(w)) + \varepsilon_n\) with \(\varepsilon_n\), normally distributed with zero mean and standard deviation equal to 0.001, resulting in a signal to noise ratio of 40dB. (A) Actual (red) and learnt (blue) attractors; predicted attractor is observed to have less noise especially along “hairpin bend”. (B) represents the learnt (blue) and actual (red) densities of the first coordinates of the data. (C) Time-series of actual and predicted the \(v_y\) coordinates in an epoch. (D,E,F) Analogous plots for data \((w_n)\) coming from the Pomeau-Manneville family of maps [11, Eqn. 25]. Input \((u_n)\) is given by \(u_n = w_n - \text{mean}(w) + \varepsilon_n\) with \(\varepsilon_n\), distributed normally with zero mean and standard deviation equal to 0.01, roughly resulting in a signal to noise ratio of 26dB.

Discussion

Finding the 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 a pursuit 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., [16, 43]). 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 exactly. In particular, we produce a topological conjugacy (semi-conjugacy) between a map that describes the single-delay lag dynamics and the underlying homeomorphism (non-invertible map that can be discontinuous) dynamical system. Also, the map that describes the single-delay lag dynamics is not found in other reservoir computing algorithms.

With our methodology, one does not need expert human intuition or physical insights to decide on the observables and library functions that are employed in other data-driven approaches. This is very useful in high-dimensional forecasting tasks where such insights are rare. Besides exact
reconstruction when conjugacy holds, we have the universality of observables, equations from data, and robustness to input noise. All this helps in obtaining high-fidelity models that give exceptional forecasting results with long-term topological and statistical consistency on chaotic data and even on data showing intermittency.

Acknowledgements We wish to thank Herbert Jaeger, Markus Haase, Igor Mezić for their helpful comments during the course of our work. The research was supported by the National Research Foundation of South Africa under Grant UID 115940.

References

[1] Majda AJ, Abramov R, Gershgorin B (2010) High skill in low-frequency climate response through fluctuation dissipation theorems despite structural instability. *Proceedings of the National Academy of Sciences* 107(2):581–586.

[2] Ghil M, Lucarini V (2020) The physics of climate variability and climate change. *Reviews of Modern Physics* 92(3):035002.

[3] Yule GU (1927) Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer’s sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 226(636-646):267–298.

[4] Takens F (1981) Detecting strange attractors in turbulence in *Dynamical systems and turbulence, Warwick 1980*. (Springer), pp. 366–381.

[5] Sauer T, Yorke JA, Casdagli M (1991) Embedology. *Journal of statistical Physics* 65(3):579–616.

[6] Stark J (1999) Delay embeddings for forced systems. i. deterministic forcing. *Journal of Nonlinear Science* 9(3):255–332.

[7] Robinson JC (2010) *Dimensions, embeddings, and attractors*. (Cambridge University Press) Vol. 186.

[8] Gutman Y, Qiao Y, Szabó G (2018) The embedding problem in topological dynamics and takens’ theorem. *Nonlinearity* 31(2):597.

[9] De Vries J (2013) *Elements of topological dynamics*. (Springer Science & Business Media) Vol. 257.

[10] Principe JC, Rathie A, Kuo JM (1992) Prediction of chaotic time series with neural networks and the issue of dynamic modeling. *International Journal of Bifurcation and Chaos* 2(04):989–996.

[11] Manjunath G, Clrecq A (2021). *Supporting Information that is available online along with this publication.*

[12] Eftekhari A, Yap HL, Wakin MB, Rozell CJ (2018) Stabilizing embedology: Geometry-preserving delay-coordinate maps. *Physical Review E* 97(2):022222.
[13] Jaeger H, Haas H (2004) Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science 304(5667):78–80.

[14] Lu Z, Hunt BR, Ott E (2018) Attractor reconstruction by machine learning. Chaos: An Interdisciplinary Journal of Nonlinear Science 28(6):061104.

[15] Koopman B, Neumann Jv (1932) Dynamical systems of continuous spectra. Proceedings of the National Academy of Sciences of the United States of America 18(3):255.

[16] Budišić M, Mohr R, Mezić I (2012) Applied koopmanism. Chaos: An Interdisciplinary Journal of Nonlinear Science 22(4):047510.

[17] Herteux J, Räth C (2020) Breaking symmetries of the reservoir equations in echo state networks. Chaos: An Interdisciplinary Journal of Nonlinear Science 30(12):123142.

[18] Manjunath G (2021) An echo state network imparts a curve fitting. IEEE Transaction on Neural Networks and Learning Systems; doi: 10.1109/TNNLS.2021.3099091.

[19] Wu Q, Fokoue E, Kudithipudi D (2018) On the statistical challenges of echo state networks and some potential remedies. arXiv preprint arXiv:1802.07369.

[20] Faranda D, et al. (2019) Boosting performance in machine learning of turbulent and geophysical flows via scale separation.

[21] Schmid PJ (2010) Dynamic mode decomposition of numerical and experimental data. Journal of fluid mechanics 656:5–28.

[22] Williams MO, Kevrekidis IG, Rowley CW (2015) A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. Journal of Nonlinear Science 25(6):1307–1346.

[23] Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the national academy of sciences 113(15):3932–3937.

[24] Champion K, Lusch B, Kutz JN, Brunton SL (2019) Data–driven discovery of coordinates and governing equations. Proceedings of the National Academy of Sciences 116(45):22445–22451.

[25] Brunton SL, Brunton BW, Proctor JL, Kaiser E, Kutz JN (2017) Chaos as an intermittently forced linear system. Nature communications 8(1):1–9.

[26] Champion KP, Brunton SL, Kutz JN (2019) Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. SIAM Journal on Applied Dynamical Systems 18(1):312–333.

[27] Yair O, Talmon R, Coifman RR, Kevrekidis IG (2017) Reconstruction of normal forms by learning informed observation geometries from data. Proceedings of the National Academy of Sciences 114(38):E7865–E7874.

[28] Ingram WT, Mahavier WS (2011) Inverse limits: From continua to Chaos. (Springer Science & Business Media) Vol. 25.

[29] Mezić I (2020) Koopman operator, geometry, and learning. arXiv preprint arXiv:2010.05377.
[30] Pomeau Y, Manneville P (1980) Intermittent transition to turbulence in dissipative dynamical systems. *Communications in Mathematical Physics* 74(2):189–197.

[31] Bottiglieri M, Godano C (2007) On-off intermittency in earthquake occurrence. *Physical Review E* 75(2):026101.

[32] Klages R (2013) Weak chaos, infinite ergodic theory, and anomalous dynamics in *From Hamiltonian chaos to complex systems*. (Springer), pp. 3–42.

[33] Kloeden PE, Rasmussen M (2011) *Nonautonomous dynamical systems*. (American Mathematical Soc.) No. 176.

[34] Manjunath G, Jaeger H (2014) The dynamics of random difference equations is remodeled by closed relations. *SIAM Journal on Mathematical Analysis* 46(1):459–483.

[35] Manjunath G, Jaeger H (2013) Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. *Neural computation* 25(3):671–696.

[36] Jaeger H (2001) The “echo state” approach to analysing and training recurrent neural networks— with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148(34):13.

[37] Manjunath G, Jaeger H (2013) Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. *Neural computation* 25(3):671–696.

[38] Grigoryeva L, Ortega JP (2018) Echo state networks are universal. *Neural Networks* 108:495–508.

[39] Manjunath G (2020) Stability and memory-loss go hand-in-hand: three results in dynamics and computation. *Proceedings of the Royal Society A* 476(2242):20200563.

[40] Sauer T, et al. (2009) Detection of periodic driving in nonautonomous difference equations in *Advances in Discrete Dynamical Systems*. (Mathematical Society of Japan), pp. 301–309.

[41] Manjunath G (2021) Embedding information onto a dynamical system. *arXiv preprint arXiv:2105.10766*.

[42] Eisner T, Farkas B, Haase M, Nagel R (2015) *Operator theoretic aspects of ergodic theory*. (Springer) Vol. 272.

[43] Korda M, Putinar M, Mezić I (2020) Data-driven spectral analysis of the koopman operator. *Applied and Computational Harmonic Analysis* 48(2):599–629.

[44] Puccetti G (2019) Measuring linear correlation between random vectors. *Available at SSRN 3116066*.

[45] Appeltant L, et al. (2011) Information processing using a single dynamical node as complex system. *Nature communications* 2:468.

[46] Kudithipudi D, Saleh Q, Merkel C, Thesing J, Wysocki B (2016) Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing. *Frontiers in neuroscience* 9:502.
[47] Vandoorne K, et al. (2014) Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature communications* 5:3541.

[48] Chollet F, et al. (2015) Keras (GitHub Repository [https://github.com/fchollet/keras](https://github.com/fchollet/keras)).

[49] Baladi V (2014) Linear response, or else. *Proceedings, The International Congress of Mathematicians* (III):525–545.

[50] Kaplan H (1993) Type-i intermittency for the hénon-map family. *Physical Review E* 48(3):1655.

[51] Lasota A, Mackey MC (2013) *Chaos, fractals, and noise: stochastic aspects of dynamics*. (Springer Science & Business Media) Vol. 97.