

Learning iterated function systems from time series of partial observations

Emilia Gibson* and Jeroen S.W. Lamb*^{†‡}

August 20, 2025

Abstract

We develop a methodology to learn finitely generated random iterated function systems from time-series of partial observations using delay embeddings. We obtain a minimal model representation for the observed dynamics, using a hidden variable representation, that is diffeomorphic to the original system.

1 Introduction

In data-driven modelling of dynamical systems, learning equations of motion from time-series observations is a central objective and many methodologies have been developed to learn ordinary and partial differential equations (and their discrete time analogues), as well as transfer and Koopman operators in these settings.

In recent decades, there has been a steadily growing interest in dynamical systems whose equations of motion contain a probabilistic element, so-called *random dynamical systems* (RDSs). These kind of dynamical systems represent dynamical systems driven by “noise”, reductions of (higher dimensional) complex dynamical systems where ignored degrees of freedom are represented in a probabilistic way, or express models with an inherent uncertainty. Models with noise are state-of-the-art in topical applications in the life sciences, physical sciences, social sciences and finance [10]. Whereas admittedly the most common modelling framework for random dynamical systems is stochastic (partial) differential equations (S(P)DEs), there are many other types of random dynamical systems (that are more suitable than S(P)DEs in certain contexts).

In this paper we address the task of learning one of the most elementary types of random dynamical systems: finitely generated random iterated function systems. A random iterated function system refers to a discrete-time dynamical system, whose evolution consists of the sequential application of self-maps (functions) on a (metric) state space, where the selection process of the sequence of maps is governed by probability. We assume that the number of functions is finite and the probabilistic process determining the sequence(s) is that of an irreducible finite state Markov Chain (MC), with the states of this MC representing the generating functions.

Let Σ_P^+ denote an irreducible finite (k) state Markov chain with $k \times k$ transition matrix P , and let the sequence $\omega := \omega_0\omega_1\omega_2 \dots$ denote a sample. Then, we consider the time-evolution of the random iterated function system generated by a finite set of functions with compact domain M

$$f_i : M \rightarrow M, \quad i = 1, \dots, k. \quad (1)$$

Given a sample $\omega \in \Sigma_P^+$ and an initial condition $x_0 \in M$, the evolution of the system is given by

$$x_{n+1} = f_{\omega_n}(x_n), \quad n = 0, 1, 2, \dots \quad (2)$$

The transition matrix P governs the probability of the occurrence of different types of sequences ω . We refer to a random dynamical system of the above type as a *finitely generated random Iterated*

*Department of Mathematics, Imperial College London, London SW7 2AZ, UK (emilia.gibson19@imperial.ac.uk, jeroen.lamb@imperial.ac.uk).

[†]International Research Center for Neurointelligence, The University of Tokyo, Tokyo, 113-0033 Japan.

[‡]Centre for Applied Mathematics and Bioinformatics, Department of Mathematics and Natural Sciences, Gulf University for Science and Technology, Halwally, 32093 Kuwait.

Function System, and in abbreviation as IFS.¹ The functions f_1, \dots, f_k are thus referred to as the generators of the IFS.

We consider the setting in which the trajectories of the IFS $x_0x_1x_2x_3\dots$ are observed through a smooth observable $\psi : M \rightarrow \mathbb{R}^s$, and the task of learning the ground truth IFS from observations $z_0z_1z_2z_3\dots$, where $z_i := \psi(x_i)$. Since from observations through the lens of ψ , we have inherently no knowledge about the coordinate representation of the IFS on M , the task of *learning* at best produces an IFS that is diffeomorphic to the the ground truth IFS and accurately (re)produces the time-series of observations. Of course, the extent to which one is able to learn the system also intrinsically depends on the amount of time-series data we have available, but we will not dwell on estimating the amount of data needed to obtain an approximation with given tolerance, so our abstract results refer to the idealisation in which we have unlimited data.

Our main result is summarized as follows:

Theorem 1.1. *Under mild smoothness and non-degeneracy conditions, a finitely generated random iterated function system can be learnt from time-series of partial observations.*

This result is constructive and builds on delay-embeddings for iterated function systems [13], [3], hidden variable inference [14] and the identifiability of delay-embedded finite state MCs. The precise conditions are given in [Hypothesis 2.3](#).

We sketch our approach in [Figure 1](#), with some illustrations of data from an example that we discuss later on in this paper (Hénon IFS, [subsection 5.2](#)). Our methodology consists of three stages:

- Finding a delay embedding of the observations so as to reveal the existence of a finite number of generators of the delay-embedded IFS, with associated transition probabilities.
- Reconstruct the MC structure of the ground truth IFS from the MC structure of the delay-embedded IFS.
- Construct generators for an IFS that is bundle diffeomorphic to the ground truth IFS using hidden variable representations.

We develop the methodology step by step in the following sections: [sections 2](#) to [4](#), with each section dedicated to one of the three stages listed above. In these sections, we discuss the feasibility of each of these steps, as well as the theory that underlies [Theorem 1.1](#). Taken together, [sections 2](#) to [4](#) provide the foundation for [Theorem 1.1](#), for further details see [subsection 4.1](#). In [section 5](#) we provide some numerical examples to demonstrate the potential of our approach, and follow up with conclusions in [section 6](#).

2 Detection and separation of generators

We first discuss how an IFS presents itself in time series data. This question was previously studied in [1] for the case of full observations ($\psi = Id$). We propose an alternative framework involving multi-manifold learning, that is well-suited for general observation functions.

2.1 IFSs and multi-manifold learning

Consider first the setting of full observations, i.e., $\psi = Id$. Given the existence of k different generators, a natural first step is to divide the observations $\{x_n\}_{n \geq 0}$ into k clusters, so that each point is identified as an iterate under one of the different generators. An intuitive approach is to reference the graphs of the different functions in the IFS. We assume the following.

Hypothesis 2.1. *We assume that M is a r -dimensional manifold and that the set of generators in (1) are C^1 smooth and transversal.*

We map time series data to the graphs of the different functions by introducing the lag-1 vectors (x_n, x_{n+1}) . Then clustering points in $M \times M$ which belong to a common manifold together (see literature on multi-manifold clustering for possible algorithms, e.g. [16]) and using relation (2), we

¹IFSs have been studied extensively in the case that all the maps f_i are contractions, see e.g. [2], but in a broader RDS setting (as in this paper), we do not require any properties of the functions f_i other than some mild smoothness and non-degeneracy conditions.

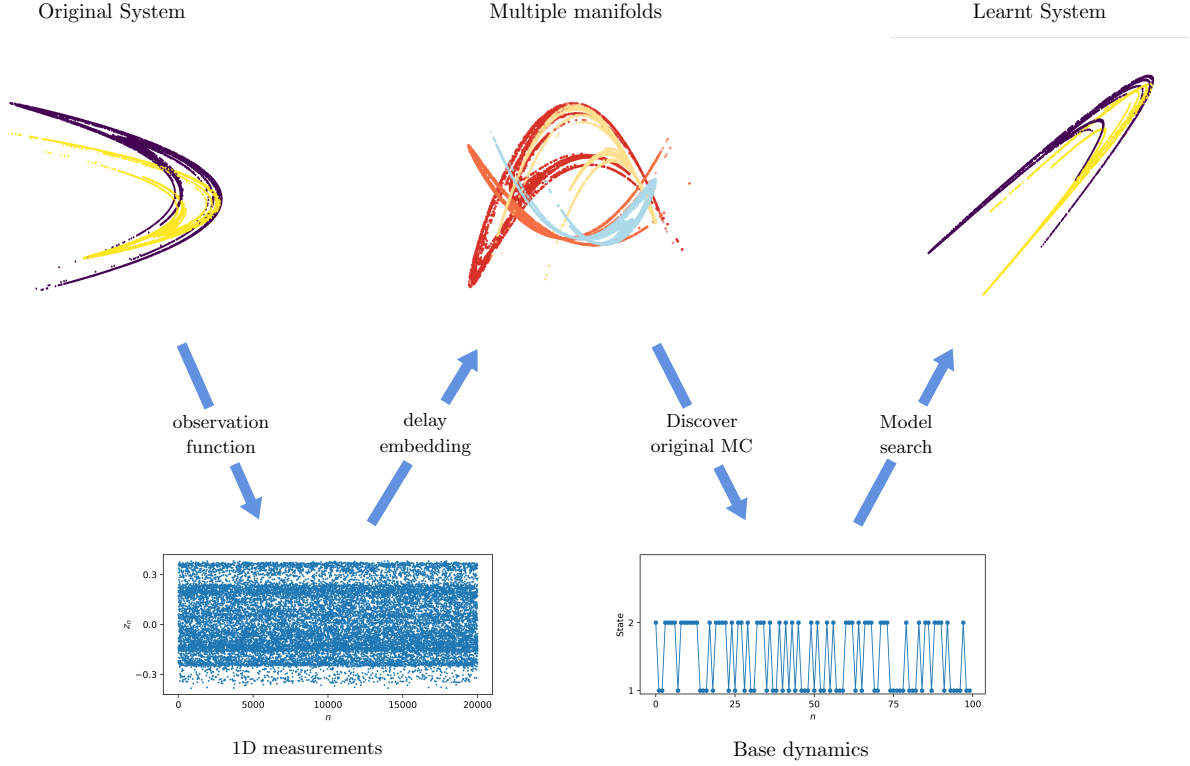


Figure 1: Flow diagram sketch of our approach towards learning IFSs, with reference to the Hénon IFS defined in [subsection 5.2](#). Points identified as iterates under different generators are shown in different colours. **Top, left:** a sample trajectory from the underlying system, which here consists of two generating functions. **Bottom, left:** scalar observation data. **Top, middle:** the 1D measurements delay embedded in \mathbb{R}^3 . In the delay space there are multiple manifolds appearing, here four, which we separate. Using transitions between different manifolds, we discover the sequence of generators from the sampled trajectory. **Bottom, right:** The discovered sequence of generators used in the sample trajectory. We fit an IFS model to the observed time series and its discovered sequence of generators. **Top, right:** A sample trajectory from the learnt IFS model.

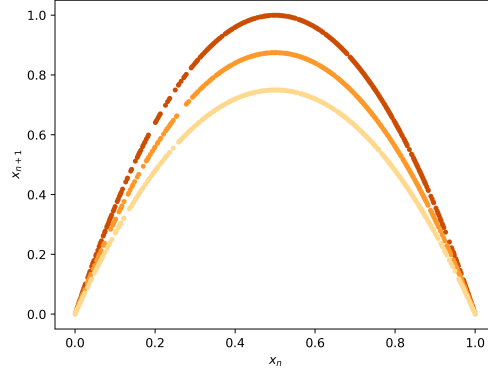


Figure 2: (Family of logistic maps). A lag-1 plot of a sample trajectory consisting of 2200 steps. The lag-1 vectors lie on a collection of manifolds, representing the graphs of the functions in the IFS, shown in different colours.

recover the sample ω which generated the data. Expressions for each of the functions in the IFS are found using standard function fitting techniques. This straightforward approach can be very effective. Multi-manifold clustering methods work best when data points are evenly distributed across each of the different manifolds. Note that prior knowledge of k is not essential.

Hypothesis 2.1 ensures that each graph is a smooth topological manifold in the product space, and that these graphs are far apart in the Whitney C^1 topology and can hence be separated. Instead of requiring that the set of generators to be transversal, it suffices to assume that the different generators are separable in the Whitney C^q topology for some $q \geq 1$ finite.

2.1.1 Family of logistic maps

Consider the 1D IFS consisting of 3 logistic maps:

$$f_1(x) = 3x(1-x), \quad f_2(x) = 3.5x(1-x), \quad f_3(x) = 4x(1-x),$$

where each map has an equal probability of being chosen at any given time step. The lagged vectors $(x_n, x_{n+1}) \in [0, 1]^2$ of a sample trajectory are displayed in [Figure 2](#). Points are confined to the three graphs of the different functions. Using the different cluster, we then recover the sequence of generators ω of the time series.

2.2 Delay embeddings of IFSs

In the previous subsection, we assumed that we could measure the entire state space. This is a strict requirement, and in practical settings, is often an unrealistic assumption. We relax this condition and assume instead that we observe the system through some measurement function $\psi : M \rightarrow \mathbb{R}^s$. We think of these ‘partial observations’ as a mapping to a lower dimensional space.

In the deterministic setting it is often assumed that the ground truth can be reconstructed from partial observations. A standard approach for reconstructing the ground truth system is to use time-delay embeddings [11, 15]. In previous work by J. Stark and D. S. Broomhead, it was shown that a ground truth IFS can also be reconstructed using time-delay embeddings. Our contribution is to use the reconstructed system $\mathbf{z}_n(l) := (z_n, z_{n+1}, \dots, z_{n+l-1})$ to find an analytic model for the original system. To simplify notation, we formulate the results for continuous scalar observables $\psi : M \rightarrow \mathbb{R}$, but these results readily extended to higher dimensional observations [12, Remark 2.9].

Given an IFS (2), the delay vector $\mathbf{z}_n(l)$ depends on the observation x_n as follows:

$$\mathbf{z}_n(l) = (\psi(x_n), \psi(f_{\omega_n}(x_n)), \dots, \psi(f_{\omega_{n+l-2}} \circ \dots \circ f_{\omega_n}(x_n))). \quad (3)$$

Hence, $\mathbf{z}_n(l)$ depends on ω through $\Omega_n := (\omega_n, \omega_{n+1}, \dots, \omega_{n+l-2})$. Let $\Phi_{\psi, f, \Omega_n} : M \rightarrow \mathbb{R}^l$ be the delay coordinate map given by

$$\Phi_{\psi, f, \Omega_n}(x_n) = (\psi(x_n), \psi(f_{\omega_n}(x_n)), \dots, \psi(f_{\omega_{n+l-2}} \circ \dots \circ f_{\omega_n}(x_n))), \quad (4)$$

then we write each delay vector in terms of its delay map as $\mathbf{z}_n(l) = \Phi_{\psi,f,\Omega_n}(x_n)$. A generic IFS delay embedded in \mathbb{R}^l admits up to k^{l-1} different delay maps, and each map has an associated $(l-1)$ -tuple $\underline{i} \in \Sigma_P^{l-1}$.

Let M be a smooth compact r -dimensional manifold. It follows from the embedding result in [13], which we state below for completeness, that if $l \geq 2d + 1$, then each delay map $\Phi_{f,\psi,\underline{i}}$ is generically an embedding.

Theorem 2.2 (Takens' theorem for IFSs [13]). *Let M be a $1 \leq r$ -dimensional smooth compact manifold. Let $C^1(M, \mathbb{R})$ be the space of C^1 real-valued functions on M and $D(M)$ be the space of diffeomorphisms of M . Suppose $l \geq 2r + 1$. Then there exists an open and dense set of $(\psi, f) \in C^1(M) \times [D(M)]^k$ such that, for any (ψ, f) in this set, $\Phi_{\psi,f,\underline{i}}$ is an embedding for every $\underline{i} \in \Sigma_P^{l-1}$.*

The dynamics in the delay space takes the form

$$\mathbf{z}_{n+1}(l) = \Phi_{\psi,f,\Omega_{n+1}} \circ f_{\omega_n} \circ \Phi_{\psi,f,\Omega_n}^{-1}(\mathbf{z}_n(l)) =: F_{\omega_n, \dots, \omega_{n+l-1}}(\mathbf{z}_n(l)), \quad (5)$$

where the subscript $(\omega_n, \dots, \omega_{n+l-1})$ refers to the fact that one time step in the delay space depends on the l most recent entries of ω . By Theorem 2.2, the ground truth system and the reconstructed dynamics in the delay space are bundle diffeomorphic. Here, the term *bundle diffeomorphism* refers to the fact that the smooth coordinate transformation Φ in (5) is dependent on ω .

By learning closed form expressions for each of the generators of the delay embedded IFS, we can use the dynamics in (5) to model the ground truth IFS. However, as the original system is non-autonomous, the symbolic dynamics of the delay embedded system also have a delay embedded structure. Therefore, model representations of the delay embedded system have an increased complexity that is not reflected in the original dynamics. This leads to the following question. From the delay embedded system, can we find a parsimonious representation in which the ground truth IFS and the inferred IFS have the same number of generators?

To this end, we first aim to discover the associated sample(s) ω that generated the observed time series. We begin by identifying each delay vector with its corresponding delay map. We assume the following.

Hypothesis 2.3 (Generic IFS).

1. The generators of the IFS are diffeomorphisms of M ,
2. the observable $\psi : M \rightarrow \mathbb{R}$ is continuous,
3. there exist integers $l, q \geq 1$ finite, and $\epsilon > 0$ such that the delay maps $\Phi_{\psi,f,\underline{i}}$ are embeddings for every $\underline{i} \in \Sigma_P^{l-1}$, and the pairwise distance between delay maps is at least ϵ in the Whitney C^q topology for all pairs excluding the diagonal.

Since each delay map is an embedding, the images of M under different delay map correspond to different r -dimensional sub-manifolds in the delay space [3], and so the delay vectors $\mathbf{z}_n(l)$ belong to the following collection of manifolds:

$$\bigcup_{\underline{i} \in \Sigma_P^{l-1}} \Phi_{\psi,f,\underline{i}}(M), \quad \forall n \geq 0.$$

Therefore, we propose to use multi-manifold learning to cluster together delay vectors that belong to the same delay map. While delay vectors are assigned to different manifolds, we still need resolve their associated $(l-1)$ -tuples. This problem is the focus of section 3.

Naturally, our strategy is also contingent upon finding a suitable delay l . There exist various techniques for determining the optimal delay size in the deterministic setting. However, these methods do not readily extend to IFSs. We want to find the smallest $l > 0$ such that each of the different delay maps are an embedding. However, popular methods for determining the minimum embedding dimension such as False Nearest Neighbours [9], or adaptations thereof [4], are designed to check if a single delay map is an embedding, and so, can only be used when the base dynamics are known a priori. Therefore, we need a new methodology to check if multiple embeddings are present or not. For the numerical examples presented in section 5, we use an adhoc, brute force approach. We search for the smallest delay which transforms the partial observations to a multi-manifold dataset by gradually increasing l . This works well for simple systems but for high dimensional and strongly nonlinear systems it may be nontrivial and remains an open problem beyond the aims of this paper.

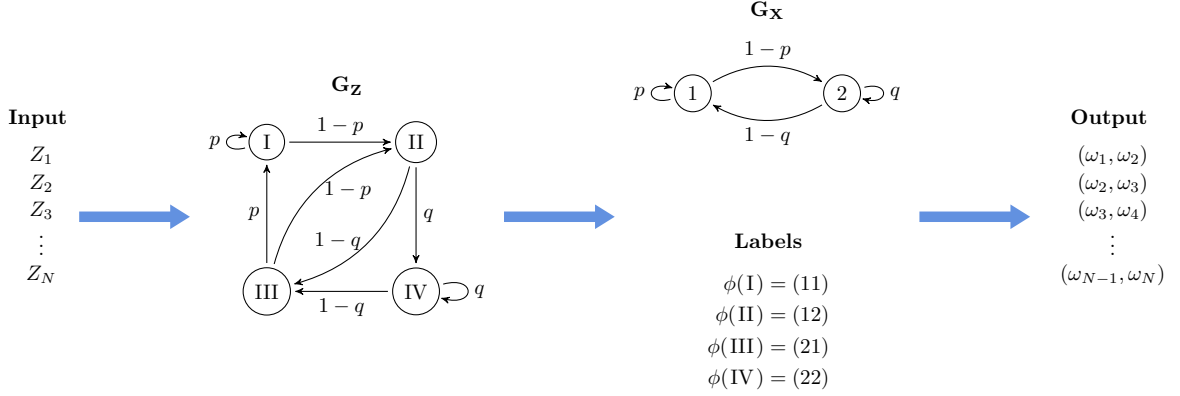


Figure 3: Summary of our procedure for recovering a time delay embedded Markov chain. Using the input sequence Z_n of delay maps, we estimate the transition graph G_Z of the underlying TDEMC. After inferring G_Z , we apply Algorithm 3.1 to discover the transition graph G_X of the original MC that drives the ground truth IFS, along with the associated $(l-1)$ -tuples of each delay map. Using ϕ we obtain the sequence of maps ω which generated the observation data.

3 Time delay embedded Markov chains

Following on from the previous section, our goal in this section is to discover the base dynamics ω that generated the observed time series. In subsection 3.1, we describe an algorithm that assigns to each delay map its corresponding $(l-1)$ -tuple and in subsection 3.2 prove that this algorithm gives the correct output. We use an abstract approach involving *time delay embedded Markov chains* (TDEMCs).

Definition 3.1 (TDEMC). *Let (X_0, X_1, \dots) be a time homogeneous, irreducible Markov chain on k symbols with transition matrix P . We define a time delay embedded Markov chain of delay $m > 1$ as*

$$Y_n = X_n X_{n+1} \cdots X_{n+m-1}. \quad (6)$$

Suppose we label the different delay maps $\Phi_{\psi, f, i}$ in (4) as I, II, \dots , $|\Sigma_P^m|$, where for notational convenience, we set $m := l-1$. Then each delay vector $\mathbf{z}_n(l)$ is assigned a label $Z_n \in \{I, II, \dots, |\Sigma_P^m|\}$. We let $\phi(i)$ denote the associated m -tuple of the i^{th} delay map for $i = I, II, \dots, |\Sigma_P^m|$, and so $\phi(Z_n) = (\omega_n, \dots, \omega_{n+l-2})$. By learning $\phi(Z_n)$ we recover the original sequence of maps ω which generated the observation data, which we then use to estimate the base dynamics' transition matrix P .

Using the sequence Z_n , we can estimate transition probabilities between different labels and store them in a transition graph. The transition graph forms the basis of our method to recover ω , outlined in Figure 3 using a simple example. From the topological structure of G_Z , we infer the topological structure of G_X . Finally, we need to verify that the edge weights of G_Z are compatible and consistent with possible edge weights on G_X , see for instance Figure 3.

3.1 Procedure to unravel a TDEMC

We use the following set up. Let (X_0, X_1, \dots) be a time homogeneous, irreducible Markov chain on k symbols with transition matrix P , and Y_n be the corresponding TDEMC of delay size m . Additionally, let $\phi: \{1, \dots, |\Sigma_P^m|\} \rightarrow \Sigma_P^m$ be a bijection. We assume we are given information on the Markov chain Z_n satisfying $\phi(Z_n) = Y_n$, for some TDEMC Y_n whose delay size m is known, but ϕ , X_n and Y_n themselves are not known. Our goal is to recover X_n from the observed MC Z_n .

Let G_X refer to the transition graph of X_n , and G_Z refer to the transition graph of Z_n . Our approach for recovering the original Markov chain X_n requires knowledge on the *elementary circuits* of G_Z , which we define below.

Definition 3.2 (Closed walk). *A walk on a graph G refers to a sequence of nodes $\{v_1, \dots, v_n\}$, $n \geq 2$, such that for all $i = 1, \dots, n-1$, there exists a directed edge in G from v_i to v_{i+1} . A walk is closed if $v_n = v_1$.*

Definition 3.3 (Elementary circuit). *A circuit is a closed walk, in which each edge is traversed only once. An elementary circuit is a closed walk, in which each node is visited only once (except the starting node).*

The set of all elementary circuits Ξ of a graph can be obtained using Johnson's circuit finding algorithm [8]. Let Ξ_X and Ξ_Z refer to the sets of elementary circuits of G_X and G_Z respectively. Going forwards, we shall omit writing the final vertex in a closed walk, since the final node in the walk is immediately determined by its initial node.

We use the standard notion of distance on directed graphs [5].

Definition 3.4 (Directed distance). *Let \mathcal{N} denote the set of nodes of a graph G and $U, V \subset \mathcal{N}$. We define the directed distance from U to V in a graph G , as*

$$d(U, V) = \min_{u \in U, v \in V} d(u, v),$$

where $d(u, v)$ is the length of any shortest path starting at u and ending at v . Note that in a directed graph $d(U, V) \neq d(V, U)$.

We also need the following properties on nodes $v \in G_Z$, which follow directly from the definition of G_Z .

Proposition 3.5.

1. Let $\xi \in \Xi_Z$ be an elementary circuit of length $l < m$. Let $\{v_1, \dots, v_l\}$ denote the nodes of ξ . Let $\phi(v_i) = a_{i_1} \cdots a_{i_m}$. Then for all $v_i \in \xi$, for all $1 \leq j \leq m - l$,

$$a_{i_j} = a_{i_{j+l}}.$$

2. Fix $u, v \in G_Z$ and let $\phi(u) = a_1 \cdots a_m$.

(a) If $d(v, u) < m$ then $\phi(v) = (\cdots, a_1, \dots, a_{m-d(v, u)})$.

(b) If $d(u, v) < m$ then $\phi(v) = (a_{d(u, v)+1}, \dots, a_m, \cdots)$.

The procedure for recovering X_n from Z_n is as follows. We use [Algorithm 3.1](#) to determine ϕ and hence recover Y_n . From Y_n we obtain the original Markov chain X_n (up to a possible relabelling).

Algorithm 3.1 Unembedding a TDE Markov chain

Inputs.

1. Delay size $m \in \mathbb{N}$.
2. List of elementary circuits $\Xi = [\xi_1, \dots, \xi_S]$ in G_Z sorted by length, from shortest to longest.
3. Weighted adjacency matrix Q of G_Z .

Ouputs.

1. A unique m -tuple for each node $v \in G_Z$.
2. The unembedded graph G_X .
3. Weighted adjacency matrix P of G_X .

Initialise.

1. A null graph G_X , i.e., a graph with 0 nodes and 0 edges.
2. $L = []$. A list storing a subset of nodes in G_Z which belong to elementary cycles and have an assigned tuple.

Steps. For $k = 1, \dots, S$:

1. Remove ξ_k from Ξ . Let $\{v_1, \dots, v_{n_k}\}$ denote the nodes of ξ_k in order along the circuit.
 2. Assign a tuple to each state in ξ_k using [Algorithm 3.2](#).
 3. Let $\beta(v_i)$ denote the last element of its associated tuple (of length m).
 - 3.1 If the closed walk $\{\beta(v_1), \dots, \beta(v_{n_k})\} \subseteq G_X$, do not update L .
 - 3.2 Else, add the elementary circuit $\{\beta(v_1), \dots, \beta(v_{n_k})\}$ to G_X .
For each $i = 1, \dots, n_k$, set $P_{\beta(v_i), \beta(v_j)} = Q_{v_i, v_j} > 0$, where $j := (i + 1) \bmod n_k$.
Append to L the set $\{v_i : v_i \notin L\}$.
-

Theorem 3.6. *Algorithm 3.1 outputs G_X, ϕ .*

We defer the proof of Theorem 3.6 to subsection 3.2.

Remark 3.7. It suffices to consider only a minimum circuit basis (see [7] for details) of G_Z as input to Algorithm 3.1 rather than all elementary circuits of G_Z . This may reduce the size of the computation.

Algorithm 3.2 Assigning m -tuples to an elementary circuit

Inputs.

1. Delay size $m \in \mathbb{N}$.
2. Elementary circuit $\xi_k = \{v_{k_1}, \dots, v_{k_{n_k}}\} \subset G_Z$, with the nodes of ξ_k listed in order along the circuit.
3. A list L containing a subset of nodes in G_Z which belong to elementary cycles and have an assigned tuple.
4. Local variable N_V which counts the number of nodes in G_X .

Outputs.

1. A unique m -tuple for each of the nodes $v_{k_i} \in \xi_k$.

Steps.

1. If $L \neq \emptyset$, for each $v_{k_i} \in \xi_k$ with $d(v_{k_i}, L) < m$ or $d(L, v_{k_i}) < m$, update its tuple according to Proposition 3.5.2.
 - 2 While there are nodes v_{k_i} with incomplete tuples:
 - 2.1 Take any node v_{k_j} with an incomplete tuple.
Set the first missing element in its tuple to $N_V + 1$.
 - 2.2 Update the tuple of v_{k_j} according to Proposition 3.5.1.
 - 2.3 Update the tuples of the other nodes in ξ_k using Proposition 3.5.2.
 - 2.4 Set $N_V = N_V + 1$.
-

We demonstrate Algorithm 3.1 on the TDE Markov chain whose transition graph G_Z is shown in Figure 3 and has delay size $m = 2$. The elementary circuits of G_Z , sorted by length, are

$$\Xi = \{\{1\}, \{IV\}, \{II, III\}, \{I, II, III\}, \{II, IV, III\}, \{I, II, IV, III\}\}.$$

Applying Algorithm 3.1, we take the first circuit in Ξ . Since this circuit has length 1, we assign to state 1 the tuple (11) and append the state to L . Next, we take the circuit $\{IV\}$. As $d(\{IV\}, \{I\}) = d(\{I\}, \{IV\}) \not< m$ and $|\{IV\}| = 1$, we assign to state IV the tuple (22) and append this state also to L . The next smallest circuit in Ξ is $\{II, III\}$. Using existing knowledge on assigned tuples to states in L and Proposition 3.5.2, we infer that state II is associated to the tuple (12) and state III is associated to (21). Having assigned tuples to each of the states in G_Z , by taking into account the edge weights of the embedded graph, we obtain the transition graph G_X shown in Figure 3.

3.2 Proof of Theorem 3.6

We begin by showing that the TDE Markov chain is irreducible if and only if the original Markov chain is irreducible.

Proposition 3.8. X_n is irreducible iff Z_n is irreducible.

Proof. (\implies) Suppose X_n is irreducible. Fix nodes $u, v \in G_Z$. WLOG write $\phi(u) = a_1 \cdots a_m$ and $\phi(v) = b_1 \cdots b_m$. We first show v is accessible from u . Since X_n is irreducible, we know b_1 is accessible from a_m in X . Hence, there exists $n \in \mathbb{N}$ such that

$$\mathbb{P}(X_n = b_1 | X_m = a_m, \dots, X_1 = a_1) > 0.$$

Since $v \in G_Z$, we have $\mathbb{P}(X_m = b_m, X_{m-1} = b_{m-1}, \dots, X_2 = b_2 | X_1 = b_1) > 0$. Hence, by homogeneity and the Markov property, $\exists n \in \mathbb{N}$ such that $\mathbb{P}(Z_n = b | Z_1 = a) > 0$. Analogously, we can show that a is accessible from b and hence Z_n is irreducible.

(\impliedby) Suppose X_n is not irreducible. By definition, there exist nodes $a_m, b_m \in G_X$ such that

$$\mathbb{P}(X_n = b_m | X_1 = a_m) = 0 \quad \forall n > 0.$$

Fix nodes $a_1, \dots, a_{m-1}, b_1, \dots, b_{m-1} \in G_X$ such that $\mathbb{P}(X_2 = a_2, \dots, X_m = a_m | X_1 = a_1) > 0$ and $\mathbb{P}(X_2 = b_2, \dots, X_m = b_m | X_1 = b_1) > 0$. Then there exists nodes $\phi^{-1}(a_1 a_2 \dots a_m), \phi^{-1}(b_1 b_2 \dots b_m) \in G_Z$, however,

$$\mathbb{P}(Z_n = \phi^{-1}(b_1 b_2 \dots b_m) | Z_1 = \phi^{-1}(a_1 a_2 \dots a_m)) = 0 \quad \forall n > 1.$$

Therefore, Z_n is not irreducible. This completes the proof. \square

As X_n is irreducible by assumption, then by [Proposition 3.8](#), Z_n is also irreducible. By definition of irreducibility, this implies that the graphs G_X and G_Z are both strongly connected. As G_X is strongly connected, each edge in G_X belongs to an elementary circuit [\[7\]](#). Therefore, to recover G_X , it is sufficient to identify each of the elementary circuits in G_X .

The next two propositions describe the relationship between elementary circuits in the two topological graphs G_X and G_Z .

Proposition 3.9. *The number of elementary circuits in G_Z is greater than or equal to the number of elementary circuits in G_X .*

Proof. Let $\{a_1, a_2, \dots, a_p\}$ be an elementary circuit of size p in G_X . Note that necessarily $p \leq k$, the number of states in the original Markov chain. Let v_i satisfy

$$\phi(v_i) := a_i a_{(i+1) \bmod p} a_{(i+2) \bmod p} \dots a_{(i+m-1) \bmod p},$$

for each $i = 1, \dots, p$. Then, by construction, $\{v_1, \dots, v_p\}$ is a elementary circuit of G_Z . \square

Proposition 3.10. *Elementary circuits in G_Z are formed by delay embedding closed walks in G_X which are not necessarily elementary circuits.*

Proof. Let $\xi = \{v_1, \dots, v_s\}$ be an elementary circuit in G_Z and let $\phi(v_i) := a_{i_1} \dots a_{i_m}$ for each $i = 1, \dots, s$. Then it follows that $\{a_{i_1}, \dots, a_{i_{s_1}}\}$ is a closed walk on G_X . If the closed walk is not an elementary circuit in G_X it must be a combination of two or more elementary circuits in G_X [\[7, Lemma 2\]](#). \square

As an example, consider again the TDE Markov chain depicted in [Figure 3](#). G_X admits three elementary circuits: $\{1\}$, $\{2\}$, $\{1, 2\}$, while G_Z admits six elementary circuits:

$$\{1\}, \{IV\}, \{II, III\}, \{I, II, III\}, \{II, IV, III\}, \{I, II, IV, III\},$$

where

$$\phi(11) = I, \phi(12) = II, \phi(21) = III, \phi(22) = IV.$$

The extra elementary circuits of G_Z correspond to the (delay embedded) closed walks $\{1, 1, 2\}$, $\{1, 2, 2\}$, $\{1, 1, 2, 2\}$ on G_X .

Following [Proposition 3.10](#), we shall say that an elementary circuit ξ_Z in G_Z corresponds to a closed walk ξ_X in G_X if embedding the path $\{\xi_X, \xi_X, \dots\}$ on G_X forms the path $\{\xi_Z, \xi_Z, \dots\}$ on G_Z , and vice versa. It remains to find the elementary circuits in G_Z that also correspond to elementary circuits in G_X . Some of the shared elementary circuits of the two graphs can be easily identified, these circuits are characterised in [Observation 3.11](#).

Observation 3.11. *Let l denote the length of the smallest circuit(s) in G_X . If a circuit $\xi \in G_Z$ has length l , then $\xi \in \Xi_Z$.*

Proof. This follows directly from [Proposition 3.10](#). If ξ does not correspond to an elementary circuit in G_X it must be a combination of two or more elementary circuit in G_X and therefore have length greater than l . \square

We can naturally extend [Observation 3.11](#) as follows. Let Π denote the set of elementary circuits in G_Z which also correspond to elementary circuits in G_X . Denote by Π_l the set of elementary circuits in Π of length $\leq l$.

Corollary 3.12. *Let $\xi^* \in \Xi_Z$ with $|\xi^*| = \min_{\xi \in \Xi_Z} \{|\xi| : |\xi| > l\}$. Then either $\xi^* \in \Pi$ or ξ^* is a combination of two or more elementary circuits in Π_l .*

Lastly, we need the following proposition, which states that if two elementary circuits in G_Z are separated by a distance of at least m , then the two corresponding closed walks in G_X do not share any common nodes.

Proposition 3.13. *Let ξ_X and ξ'_X be closed walks on G_X . Suppose ξ_X and ξ'_X share no states in common. Let ξ_Z, ξ'_Z be the corresponding walks embedded on G_Z of ξ_X and ξ'_X respectively. Then $d(\xi_Z, \xi'_Z) \geq m$.*

Proof. Suppose for contradiction, that there exists $u \in \xi_Z, v \in \xi'_Z$ such that $d(u, v) < m$. Without loss of generality, let $\phi(u) = a_1 \cdots a_m$ and $\phi(v) = b_1 \cdots b_m$. From Proposition 3.5.2, we have $b_1 = a_{d(u,v)+1}$ and, hence, ξ_X and ξ'_X must share at least one state in common. This contradicts the initial assumption, we therefore conclude that $d(u, v) \geq m$. \square

We combine the above results to prove Theorem 3.6.

Proof of Theorem 3.6. Every edge in G_X belongs to at least one elementary circuit. In the proof of Proposition 3.10 we show that every elementary circuit in G_X also appears embedded in G_Z . It follows by induction using Proposition 3.5, Corollary 3.12 and Proposition 3.13 that Algorithm 3.1 is able to uniquely identify every edge in G_X and recover ϕ , after a possible relabelling, from elementary circuits of G_Z . \square

4 Identifying diffeomorphic iterated function systems

We discuss the final step in our methodology: how to construct generators for an IFS that is bundle diffeomorphic to the ground truth IFS. As established in sections 2 and 3, we can learn the different generators in (5) and their underlying symbol structure. We want to learn a minimal number of generators, equal to the number of generators in the ground truth model. It is tempting to try learning such a model by reformulating each mapping $F_{\omega_n, \dots, \omega_{n+l-1}}$ in (5) as a composition $g_{\omega_{n+l-1}} \circ \dots \circ g_{\omega_n} : \mathbb{R}^l \rightarrow \mathbb{R}^l$. In general this is not possible because the non-autonomous dynamics cannot be disentangled in the delay space.

However, we can learn generators g_i that are compatible with the partial observations. Following section 3, we start with the knowledge of the sample ω associated to the time series of partial observations. In subsection 4.1 we investigate the relationship between the ground truth IFS and other IFSs that satisfy (5) and in subsection 4.2 we discuss a particular method for finding such models.

4.1 Dynamical Equivalence

Suppose there exists an IFS, different to the ground truth IFS but with the same Markov chain in the base, defined on some compact manifold M' and a function $\psi' : M' \rightarrow \mathbb{R}$ such that the new observable and ground truth IFS observations both evolve according to (5). In other words, we assume that both IFSs satisfy the same “reduced model” [14]. In this case, for generic observations the two systems are bundle diffeomorphic. This follows directly from Takens’ theorem for IFSs [13].

Theorem 4.1. *Consider two IFSs*

$$x_{n+1} = f_{\omega_n}(x_n), \quad y_{n+1} = g_{\omega_n}(y_n), \quad n = 0, 1, 2, \dots$$

where $f_i : M \rightarrow M, g_i : M' \rightarrow M'$ for each $i = 1, \dots, k$ and $\omega_0 \omega_1 \omega_2 \dots \in \Sigma_P^+$. Let $\Omega_n = (\omega_n, \dots, \omega_{n+l-1})$. Suppose there exists observation functions $\psi : M \rightarrow \mathbb{R}, \psi' : M' \rightarrow \mathbb{R}$ and $l \geq 1$ such that Φ_{ψ, f, Ω_n} and $\Phi_{\psi', g, \Omega_n}$ are embeddings of M and M' resp. and that

$$\Phi_{\psi, f, \Omega_{n+1}} \circ f_{\omega_n} \circ \Phi_{\psi, f, \Omega_n}^{-1} = \Phi_{\psi', g, \Omega_{n+1}} \circ g_{\omega_n} \circ \Phi_{\psi', g, \Omega_n}^{-1}.$$

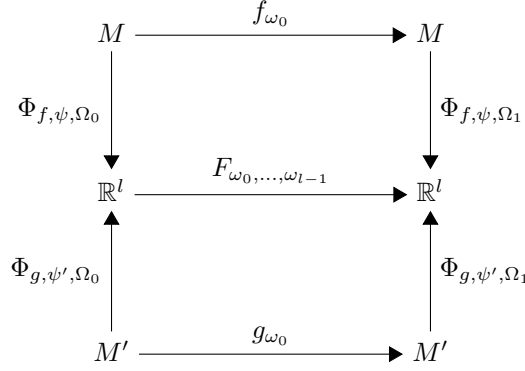
for each admissible Ω_n . Then the both systems reduce to (5) and hence they are bundle diffeomorphic.

Proof. Let F denote the delay reconstructed system (5). Using Takens theorem for IFSs, we have $(f, \omega) \longleftrightarrow (F, \omega)$, with \longleftrightarrow denoting the equivalence relation of (smooth) bundle conjugacy [13].

Similarly, we have $(g, \omega) \longleftrightarrow (F, \omega)$. Applying transitivity of \longleftrightarrow , we obtain

$$(f, \omega) \longleftrightarrow (g, \omega).$$

We thus have the following conjugacy diagram:



□

Remark 4.2. [Theorem 4.1](#) and its proof hold analogously for autonomous deterministic dynamical systems with (smooth) topological conjugacies replacing (smooth) bundle conjugacies.

We finally show that the results in [sections 3](#) and [4](#) above constitute a proof of our main result in [Theorem 1.1](#) for an IFS satisfying [Hypothesis 2.3](#) and the idealisation of unlimited data.

Proof of Theorem 1.1. the first thing we observe is that in a suitable delay space, we can separate the time series into different delay maps. Through [section 3](#) we can identify the original driving signal. From [Section 4](#) we can associate the driving signal to a set of generators that is bundle diffeomorphic to the ground truth IFS.

The final assumption in [Hypothesis 2.3](#) ensures that the different delay maps are topologically distinct. Therefore in theory, given unlimited data, the different delay maps can be separated (with any ambiguous delay vectors which lie in the intersection of two manifolds being removed). It follows from [Theorem 3.6](#) that after separating the different delay maps, we can recover the base dynamics that generated the data. Using optimisation techniques, such as those described in [subsection 4.2](#), we can discover a set of generators whose evolution in the delay coordinates reduces to [\(5\)](#). [Theorem 4.1](#) implies that learnt system is bundle diffeomorphic to the ground truth IFS.

□

4.2 Hidden Dynamics Inference

Inspired by [\[14\]](#), we use hidden dynamic inference to identify IFS decompositions of the delay embedded system that satisfy [\(5\)](#).

We define a discrete time, multi-functional HDI model with V latent variables and k different functions as follows:

$$\begin{aligned} z_{n+1} &= g_{\omega_n}(\omega_n; z_n, h_n^1, \dots, h_n^V), \\ h_{n+1}^i &= g_{\omega_n}^i(\omega_n; z_n, h_n^1, \dots, h_n^V), \quad i = 1, \dots, V, \end{aligned} \tag{7}$$

where $\omega_n \in \{1, \dots, k\}$.

As proposed in [\[14\]](#), the initial condition can be included in the set of parameters to be optimised. Otherwise, the first few entries of the observed time series can be used instead. Additionally, each function in the model is assumed to be a linear combination of some pre-defined basis functions whose coefficients minimise the Mean Square Error (MSE) loss defined below, and are found using gradient-based optimization and automatic differentiation.

Let \mathbf{p} represent the discrete time multifunctional HDI model parameters. We define the MSE loss, given an input sequence of labels $\underline{\omega} = \{\omega_n\}_{n=0}^{N-1}$ as

$$\text{MSE}(\mathbf{p}) = \sum_{n=1}^{N-1} \|g_{\omega_n}(z_n, h_n^1, \dots, h_n^V) - z_{n+1}\|^2. \tag{8}$$

By minimising the MSE and by [Theorem 4.1](#), we effectively learnt an IFS model that is diffeomorphic to the ground truth IFS.

For a partially observed IFS, we set $V + 1$ equal to the (maximum) estimated dimension of the sub-manifolds $\Phi_{\psi,f,i}(M)$ in [\(4\)](#). However, models that preserve the dynamics of the observed variable can still be found for various values of V .

5 Numerical Results

In this section we illustrate our methodology on some numerical examples. We demonstrate the ability of the proposed framework to discover bundle diffeomorphic systems from time series of partial observations from two different IFSs. Both systems have nonlinear dynamics. The first example uses a Mobius IFS, the second features a hyperbolic IFS. The results can be reproduced using the source code in [6].

5.1 Curvilinear Sierpinski gasket

We first study the following Mobius IFS with compact domain $M = \{z \in \mathbb{C} : |z| \leq 1\}$. The IFS consists of the following generators,

$$\{f, R \circ f, R^2 \circ f\},$$

where $f(z) = \frac{(\sqrt{3}-1)z+1}{-z+\sqrt{3}+1}$ and $R(z) = e^{2\pi i/3}z$. The attractor of the IFS is referred to as the curvilinear Sierpinski gasket. We assume that each map in the IFS has an equal probability of being chosen and use observation function $\psi(z) = \text{Im}(z)$. The entire simulation is shown in Figure 4b.

We first embed the observations in \mathbb{R}^3 , and so there are 9 different delay maps which we separate using multi-manifold clustering. The delay embedding is shown in Figure 4a. We then unembed the TDE Markov chain and estimate the original Markov chain which generated the data using Algorithm 3.1. Lastly, we use HDI to find an analytic model representation for the observations. We are successfully able to identify an IFS comprised of 3 functions that is approximately bundle diffeomorphic to the original system. For this system we introduce a single hidden variable and use a basis of cubic polynomials. A sample trajectory from the learnt IFS model is displayed in Figure 4c.

5.2 Hénon IFS

We consider the following Hénon IFS, adapted from [1]. The generators of the IFS are:

$$f_1(x, y) = (y + 1 - 1.2x^2, 0.3x), \quad f_2(x, y) = (y + 1 - 1.2(x - 0.2)^2, -0.2x). \quad (9)$$

We let $\psi(x, y) = x$ be the observation function and simulate a sample trajectory, shown in Figure 5a. For visualisation purposes, the different Hénon maps in the IFS are displayed in different colours.

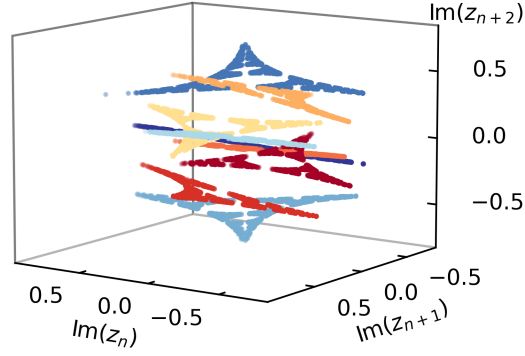
Once again, we embed the observed time series in \mathbb{R}^3 , where we are able to cluster and separate 4 different 2-dimensional sub-manifolds and obtain an estimate the original sequence of maps used. Here, we take the basis functions to be quadratic polynomials and create one hidden variable. The results of a sample trajectory from the learnt model are shown in Figure 5b.

6 Conclusions

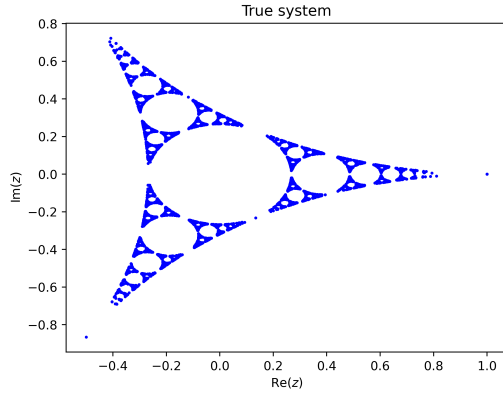
In this work we develop a data-driven methodology for learning finitely generated iterated function systems from time series data of partial observations. Although we are motivated by learning random systems, our methodology is directly applicable to non-autonomous systems with a finite number of generators. Our approach draws upon results and methods from various different areas of mathematics including, time-delay embeddings, manifold learning, directed graphs and Markov chains and model identification. We highlight differences between delay embeddings of autonomous deterministic systems and partial observations of IFSs. This study is the first step towards generalising model discovery toolkits to broader classes of random dynamical systems.

Acknowledgments

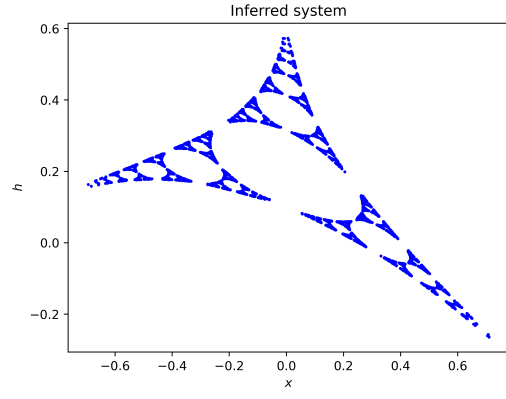
The authors are grateful to Matthew Levine, Matheus M. Castro and Kevin Webster for useful discussions and acknowledge support from the EPSRC Centre for Doctoral Training in Mathematics of Random Systems: Analysis, Modelling and Simulation (EP/S023925/1). The research of JSWL has furthermore been supported by EPSRC grants EP/W009455/1, EP/Y020669/1, EP/Y028872/1 and EP/Z533658/1, as well as by the JST (Moonshot R & D Grant Number JPMJMS2021) and GUST (Kuwait).



(a) Delay embedding of partially observed Curvilinear Sierpinski IFS. Different colours are used to identify the different delay maps.

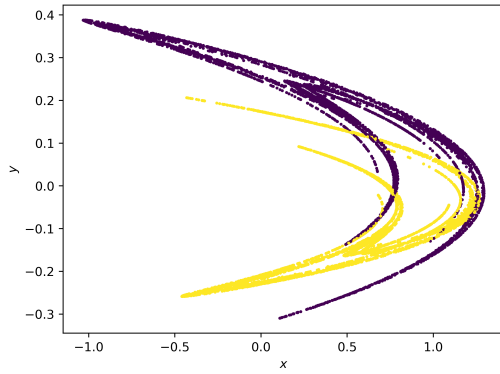


(b) Original system

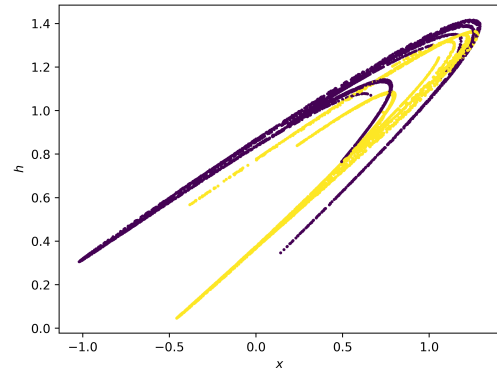


(c) Learnt system, $\text{MSE}(\mathbf{p}) = 1.3 \times 10^{-5}$

Figure 4: Results from the partially observed curvilinear Sierpinski gasket IFS.



(a) Original system



(b) Learnt system, $\text{MSE}(\mathbf{p}) = 4.5 \times 10^{-5}$

Figure 5: Sample trajectories from the ground truth Hénon IFS and the learnt IFS model.

References

- [1] Zachary Alexander, James D. Meiss, Elizabeth Bradley, and Joshua Garland. Iterated function system models in data analysis: Detection and separation. *Chaos*, 22(023103), 2012.
- [2] Michael F. Barnsley. *Fractals everywhere*. Academic Press, 2 edition, 1993.
- [3] D. S. Broomhead, J. P. Huke, M. R. Muldoon, and J. Stark. Iterated function system models of digital channels. *Proc. A.*, 460:3123–3142, 2004.
- [4] Liangyue Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Phys. D*, 110(1):43–50, 1997.
- [5] Gary Chartrand and Songlin Tian. Distance in digraphs. *Comput. Math. Appl.*, 34(11):15–23, 1997.
- [6] Emilia Gibson. Detecting-IFS, 2025.
- [7] Petra M. Gleiss, Josef Leydold, and Peter F. Stadler. Circuit bases of strongly connected digraphs. *Discuss. Math. Graph Theory*, 23:241–260, 2003.
- [8] Donald B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.*, 4(1):77–84, 1975.
- [9] Matthew B Kennel, Reggie Brown, and Henry DI Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, 45:3403–3411, 1992.
- [10] Hisashi Kobayashi, Brian L Mark, and William Turin. *Probability, random processes, and statistical analysis: Applications to communications, signal processing, queueing theory and mathematical finance*. Cambridge University Press, 2011.
- [11] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Phys. Rev. Lett.*, 45:712–716, 1980.
- [12] Tim Sauer, James A Yorke, and Martin Casdagli. Embedology. *J. Stat. Phys.*, 65(3):579–616, 1991.
- [13] J. Stark, D. S. Broomhead, M. E. Davies, and J. Huke. Delay embeddings for forced systems. ii. stochastic forcing. *J. Nonlinear Sci.*, 13:519–577, 2003.
- [14] George Stepaniants, Alasdair D Hastewell, Dominic J Skinner, Jan F Totz, and Jörn Dunkel. Discovering dynamics and parameters of nonlinear oscillatory and chaotic systems from partial observations. *Phys. Rev. Res.*, 6(043062), 2024.
- [15] Floris Takens. Detecting strange attractors in turbulence. In David Rand and Lai-Sang Young, editors, *Dynamical Systems and Turbulence*, volume 898 of *Lecture notes in Math.*, pages 366–381. Springer, Berlin, Heidelberg, 1981.
- [16] Yong Wang, Yuan Jiang, Yi Wu, and Zhi-Hua Zhou. Multi-manifold clustering. In Byoung-Tak Zhang and Mehmet A. Orgun, editors, *PRICAI 2010: Trends in Artificial Intelligence*, volume 6230 of *Lecture Notes in Computer Science*, pages 280–291. Springer, Berlin, Heidelberg, 2010.