

A Backpropagation Algorithm for Inferring Disentangled Nodal Dynamics and Connectivity Structure of Dynamical Networks

Eugene Tan, Débora Corrêa, Thomas Stemler, Michael Small

Abstract—Dynamical networks are versatile models that describe a variety of behaviours such as synchronisation and feedback in networks of coupled dynamical components. However, applying these models in real systems is difficult as prior information of the connectivity structure or local dynamics is often unknown and must be inferred from node state observations. Additionally, the influence of coupling interactions complicates the isolation of local node dynamics. Given the architectural similarities between dynamical networks and recurrent neural networks (RNNs), we propose a network inference method based on the backpropagation through time (BPTT) algorithm used to train RNNs. This method aims to simultaneously infer both the connectivity structure and isolated local node dynamics from node state observations. An approximation of local node dynamics is first constructed using a neural network. This is alternated with an adapted BPTT algorithm to regress corresponding network weights by minimising prediction errors of the network based on the previously constructed local models until convergence. This method was successful in identifying the connectivity structure for coupled networks of chaotic oscillators. Freerun prediction performance with the resulting local models and weights was comparable to the true system with noisy initial conditions. The method is also extended to asymmetric negative coupling.

Index Terms—dynamical networks, network inference, backpropagation, neural networks, machine learning

1 INTRODUCTION

Dynamical networks are a common occurrence when studying complex systems where the goal is to model large systems of multiple interacting components [1]. In its simplest form, a dynamical network may be defined as having three main components: local node dynamics, coupling dynamics and connectivity structure (see Equation 1). One key feature of dynamical networks is its ability to recreate and describe rich and interesting dynamics such as chimera states, synchronisation [2], [3], [4] and cascading failure commonly encountered in real world systems [5]. They have also been applied to other systems such as neuron networks [5], [6], power grids [7], [8], epidemic spread [9], and cardiac arrhythmia [10].

$$\dot{\mathbf{x}}_i(t) = f(\mathbf{x}_i(t)) + \sum_{i \neq j} c_{ij} g(\mathbf{x}_i(t), \mathbf{x}_j(t)). \quad (1)$$

In Eq 1, nodes are indexed with i , the coefficient c_{ij} represents coupling between nodes i and j , and f and g are the local and coupling dynamics respectively. Despite its relatively simple form, data-driven applications of this framework to model real world systems can be difficult if information regarding the coupling structure and dynamics is not accessible. In these cases, these components need to be inferred before a dynamical networks framework can be applied. Specifically, there is the inverse problem of inferring network characteristics from data. However, this problem

is difficult due to the complicated interplay between local models and coupling behaviour. This is problematic in systems where the network can only be observed in its entirety, and local dynamics and coupling structure are not clearly separable [11]. For example, in modelling brain connectivity [12], it may be desirable to infer the component dynamics of local brain regions. However, isolating these regions for observation may not be physically possible. If modelling epidemic spread between regions, it may be desirable to infer which transport connections influence the dynamics the most.

Many existing methods of inferring dynamical network properties require either the local or coupling dynamics to be at least partially known. Alternatively, some methods restrict inferences to statistical arguments with uncertainty in order to remove the dependence on a priori information. For the interested reader, a comprehensive review and discussion on data-driven network inference has also been given by Gao & Yan [13].

Numerous approximation and inference methods for inferring network connectivity have been developed each with varying success. These methods may be classified into three main approaches:

- 1) **Direct Approach** - For systems where both local f and coupling dynamics g is explicitly known, a direct algebraic approach using derivatives have been used to recover connectivity structure [14].
- 2) **Perturbation Methods** - These methods try and recreate a proxy dynamical network that performs similarly to the target system. Small perturbations are applied to individual nodes in the network and observed to see how each signal propagates. These

- E. Tan, T. Stemler and M. Small are with the Department of Mathematics & Statistics, The University of Western Australia, Crawley, WA 6009, Australia
E-mail: eugene.tan@uwa.edu.au
- D. Corrêa is with the Department of Computer Science & Software Engineering, The University of Western Australia, Crawley, WA 6009.

Manuscript received 1 Nov 2022; revised 9 Jun 2023.

propagations are then used to infer the network connectivity structure [15], [16], [17].

- 3) **Signal Correlation** - This approach focuses on quantifying the correlation between node signals [18]. Node pairs with highly correlated signals are used to infer the connectivity structure. Maximum route entropy has also been used as an alternative to achieve the same result. [19]. An extension of this approach is the employment of some notion of causality such as Granger causality [20], [21] and short-term causal dependence. [15].

The inference of connectivity structure also poses an additional challenge of false positive couplings and remains an open research area. Granger causality and similar statistical based approaches have been found to particularly struggle with this problem, even for simple network structures such as rings and chains [21], [22].

The second problem of constructing models to describe the local dynamics f in multi-node networks has been studied for pseudo-periodic systems [10], [20]. Outside these, many approaches focus on the case of a single node without external interactions and utilise a time-delay embedding with appropriately chosen delay and embedding dimension [23]. Proposed modelling methods for long-term prediction include neural network predictors [24], radial basis predictors [25] and reservoir computing [26] among several others. Assessing the quality of these local models relies on the comparison of dynamical invariants such as Lyapunov exponents or correlation dimension [27], [28], [29] and attractor homology [30].

There is also a third related problem of identifying coupling equations g from signals. Solutions have been proposed using phase dynamics for networks exhibiting pseudo-periodic behaviour, but not necessarily for aperiodic behaviour [10]. For simple oscillator networks, Panaggio et al. [31] propose a gradient descent method to regress coupling functions of a given form. This approach draws similarities with [14].

For low dimensional systems, symbolic approaches aimed at tackling either the dynamics or connectivity structure inference problem have gained increased traction in recent times. A method based on the ideas of compressed sensing [32] and sparse optimisation methods (SINDy) [33] have been proposed to reconstruct the functional form of the dynamics equations using a library of basis functions (e.g. power series, Fourier series) for isolated dynamical systems with no interactions. Compressive sensing tries to find the sparsest combination of coefficients for some power series basis via convex optimisation with respect to the L_1 -norm. Link and coupling weight identification can also be included into this approach [32], [34]. However, we note that this method may be limited to simple systems with analytical or well-behaved equations.

The symbolic approach with basis functions was also extended to local dynamics and coupling functions by utilising prior known information of the network connectivity structure [11]. A similar approach (ARNI) [35] using basis functions has also been proposed to reconstruct the coupling function and connectivity structure of dynamical networks. ARNI is also applicable to heterogeneous networks where

local dynamics f differ between nodes. We note that ARNI can also be adapted to infer local dynamics instead [11].

Despite these topics being well studied, the literature on inferring both connectivity structure and node dynamics from time-series observations is not rich. Methods for reconstructing local dynamics (e.g. SINDy [33] and Gao [11]) are either restricted to single node systems, or rely on prior information about the connectivity structure. In cases where the aim is to infer connectivity structure, either local dynamics are known prior [14] or are not directly inferred [35]. There remains the need of more methods that tackle the inference of both of these properties simultaneously.

We note that a method utilising a mean field approach has been proposed to tackle the simultaneous inference of local dynamics and connectivity structure [36]. This method constructs an 'effective network' that acts as an approximate proxy for the real dynamical network, reproducing many of its properties and behaviours. Under specific assumptions such as weak coupling, network heterogeneity and limited links, this method has been shown to perform well in identifying critical transitions and the connectivity structure of a cat cortex. However, the 'effective network' relies on the network sparsity and presence of hubs to recover an approximation of the local dynamics. Additionally, this approximation of local dynamics is not fully disentangled from the interaction dynamics. An extension and formalisation of this approach was subsequently presented [37] and demonstrated the capacity for this approach in predicting emergence behaviour.

Another dynamical systems approach was proposed by Novaes et al. [38] for analysing sparse networks of interacting oscillators. Their approach assumes dynamics occur near a Hopf-Andronov bifurcation and focuses on analysing the phases of trajectories to recover the requisite network properties.

In contrast to the more theoretical approaches previously discussed, there has recently been a growing interest in applying machine learning methods to solve this problem. Many existing methods utilise a graph neural network (GNN) architecture to perform network inference. One of the earlier formulations consists of two neural networks that are trained simultaneously to learn the interaction structure and resulting dynamics (conditioned on said interaction) [39]. Similar ideas using GNNs have also been proposed by Cranmer et al. [40] to predict the movement of many body particles, given the interaction structure is known a priori. Even more recently, Zhang et al. [41] presented a neural architecture consisting of a network generation layer and three dynamics layers to model a generalised dynamical network. This network is then subsequently trained end-to-end for the purposes of prediction. Whilst these methods are able to infer network dynamics, these approaches produce black box models that combine both the interaction and local dynamics together. That is, the learned local dynamics are conditional on a given network connection structure and are not necessarily separable.

In this paper, we present a proof of principle of a data-driven approach for simultaneously inferring both the local node dynamics and weighted connectivity structure from the observation of node signals, assuming only prior knowledge of the coupling function. This method aims to decouple

the time-series observations of node states to reconstruct the separated components of local dynamics $f(x_i)$ and coupling connectivity structure c_{ij} . The novelty of this method lies in the separated output approximations of local dynamics and coupling structure. This differs from the ‘effective network’ method and various machine learning approaches that infer local dynamics without fully removing interaction dynamics.

Recognising the structural similarities between dynamical networks and RNNs, we propose an adaptation of the backpropagation through time (BPTT) algorithm used to train RNNs [42], as a method of regressing the connectivity structure of a dynamical network. This regression is then used to decouple the observed node signals and simultaneously construct a model of local dynamics. This method combines the tasks achieved by several of the abovementioned methods, whilst also dealing instances of dynamical networks with weighted edges. The proposed method is formulated on the following assumptions: (1) the coupling function $g(x_i, x_j)$ is known, (2) all nodes states are observable, (3) the network dynamics are not fully synchronised, and (4) the magnitude of coupling effects is small in compared to the local node dynamics. In our analyses, assumptions (1) and (2) are encoded as apriori information (i.e. g is diffusive coupling, and all node states are inputs for regression). Assumptions (3) and (4) can be achieved by either considering a sparsely connected network, or weak coupling magnitudes c_{ij} . However, we note that it is not needed for networks to be both sparse and have weak coupling for these assumptions to be met.

We test the performance of the proposed method on a dynamical network of Lorenz [43] and Chua oscillators [44] with a randomly initialised weighted connectivity structure. We find that an adapted backpropagation regression method was able to reproduce the network coupling structure and local dynamics for a 64 node network. Additional cases with asymmetric and negatively weighted coupling are also explored, yielding similar results. To test the performance on a more realistic application, the proposed backpropagation regression method was applied to a network of driven FitzHugh-Nagumo oscillators operating in the chaotic regime to simulate a simple biological neuron network (see Appendix). Similar to the previous cases, the backpropagation method was able to successfully reproduce both the network structure and local dynamics.

2 BACKGROUND

2.1 Dynamical Networks

Dynamical networks can be defined as a network $G = (f, g, C)$ consisting of three main components, local dynamics f , coupling dynamics g and connectivity structure C . Node states evolve according to Equation 1 with local behaviour governed by f and additional external influences from neighbours due to coupling components g and C .

The effect of coupling between nodes results in a high dimensional system with multiple interacting sub-systems. For small or highly connected networks, sufficiently large coupling can cause network-wide synchronisation of node states. As coupling is decreased, chimera states and group

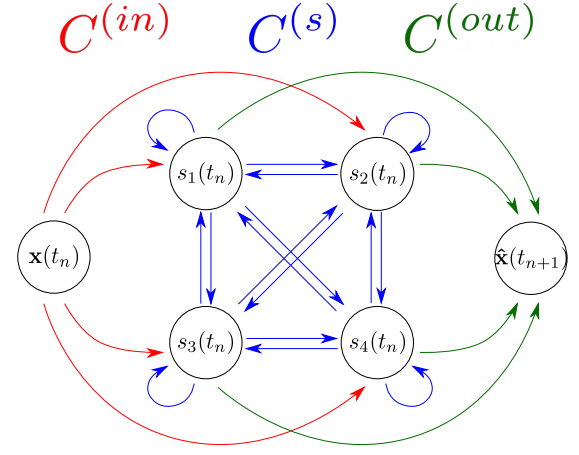


Fig. 1. Illustration of the forward pass and one step prediction of RNNs with hidden node state evolutions given by Equation 2

synchronisation emerges [2], [3], [4]. When no coupling is present, all nodes behave independent of each other.

Here, we focus on dynamical networks based on two different 3-dimensional chaotic systems, Lorenz [45] and Chua, with diffusive coupling in the first component. For the latter, we will focus on the Chua system with cubic nonlinearity ϕ operating in the single scroll mode [44] to analyse dynamics contrasting with the Lorenz system.

2.2 Recurrent Neural Networks (RNN)

Recurrent neural networks (RNNs) are a modelling architecture developed in the field of machine learning that is widely used for sequence and time-series prediction [46], [47], [48]. RNNs consist of an input layer with weights $C^{(in)}$, a fully connected hidden layer of nodes with time varying state $s(t)$ and feedback weights $C^{(s)}$, and an output layer with readout weights $C^{(out)}$. For prediction, an input sequence $\mathbf{x}_{in} = \{\mathbf{x}(t_0), \mathbf{x}(t_1), \dots, \mathbf{x}(t_n)\}$ is fed into the RNN via the input layer with input weights $C^{(in)}$. These inputs are usually discretely sampled at regular time intervals t_n and then sequentially fed into the recurrent hidden layer whose nodes feedback to each other according to weights $C^{(s)}$ before applying a nonlinear transformation $\sigma(\cdot)$. In its continuous time formulation, this allows network to approximate nonlinear dependencies between states with nodes evolving according to Equation 2,

$$\mathbf{s}(t_n) = \sigma(C^{(in)}\mathbf{x}(t_n) + C^{(s)}\mathbf{s}(t_{n-1}) + \mathbf{b}), \quad (2)$$

where $\mathbf{s}(t_n) = (s_1(t_n), s_2(t_n), \dots, s_N(t_n))$ represents the state values of all N hidden nodes in the RNN.

Predictions $\hat{\mathbf{x}}(t)$ are calculated using the trained readout weights and hidden node states as given in Equation 3 and Figure 1,

$$\hat{\mathbf{x}}(t_{n+1}) = C^{(out)} \cdot \mathbf{s}(t_n). \quad (3)$$

The training of RNNs involve the small adjustment of network weights $C^{(in)}$, $C^{(s)}$, $C^{(out)}$ and biases $\mathbf{b}$ such that the error between $\mathbf{x}(t_n)$ and $\hat{\mathbf{x}}(t_n)$ is minimised. This is achieved via the backpropagation through time (BPTT) algorithm [20]. The BPTT algorithm can be interpreted as an adaptation of the gradient descent training algorithm

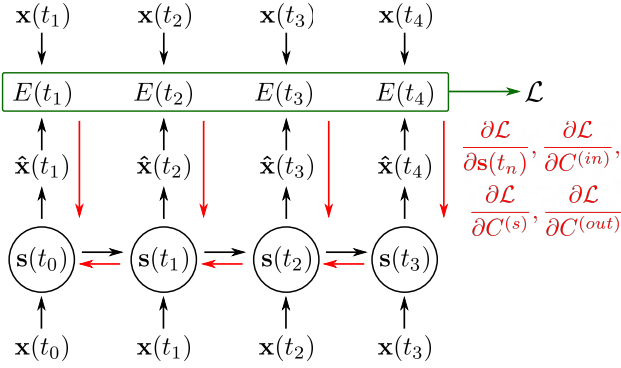


Fig. 2. Schematic of typical backpropagation through time (BPTT) calculation for training recurrent neural networks with an input length of 4. Input time-series $\mathbf{x}(t_n)$ is used to evolve node states $\mathbf{s}(t_n)$ during the forward pass. For brevity, each node in the figure represents all the state of all hidden nodes in the network at a given time. Node states $\mathbf{s}(t_n)$ are used to calculate a value for the errors $E(t_{n+1})$ and $\mathcal{L}$. Partial derivatives for $\mathcal{L}$ are taken with respect to the node states $\mathbf{s}(t_n)$ and edge weights $C^{(in)}$, $C^{(s)}$, $C^{(out)}$ and subsequently used to adjust edge weights such that prediction error $\mathcal{L}$ is minimised.

for neural networks that accounts for the propagation of network weight effects through the history of the RNN's hidden node states $\mathbf{s}(t_n)$.

2.3 Backpropagation

The BPTT algorithm works by unfolding the predictions of the recurrent network at each time step. This process reframes the forward pass of an RNN into a feedforward network where each layer corresponds to the node states of the RNN at a specific time. For a given RNN model, an input sequence $\mathbf{x}_{in} = \{\mathbf{x}(t_0), \mathbf{x}(t_1), \dots, \mathbf{x}(t_n)\}$ is fed in to produce a predicted sequence of outputs $\hat{\mathbf{x}}_{out} = \{\hat{\mathbf{x}}(t_1), \hat{\mathbf{x}}(t_2), \dots, \hat{\mathbf{x}}(t_{n+1})\}$. Prediction losses are calculated with respect to some metric d between the real output $\mathbf{x}_{out}$ and the predictions $\hat{\mathbf{x}}_{out}$. For an RNN with N hidden nodes, the total loss $\mathcal{L}$ for an n step prediction with respect to the L^2 norm is given by

$$\mathcal{L} = \sum_{t=t_1}^{t=t_n} E(t) = \sum_{t=t_0}^{t=t_n} \|\hat{\mathbf{x}}_{out}(t) - \mathbf{x}_{out}(t)\|_2, \quad (4)$$

where $E(t_n)$ is the error of the n^{th} prediction step.

The gradient of the loss taken with respect to node weights is calculated by propagating the error backward through time steps and used to update node weights (see Figure 2). During the update step, each weight c_{ij} (input, output or hidden layer weights) is updated according to the calculated partial derivatives proportional to some learning rate α ,

$$c_{ij} \leftarrow c_{ij} - \alpha \frac{\partial \mathcal{L}}{\partial c_{ij}}. \quad (5)$$

From observation, the hidden layer of an RNN functions identically to a dynamical network. We aim to investigate if the BPTT approach can be adapted to regress the weights of a dynamical network given a set of input node states.

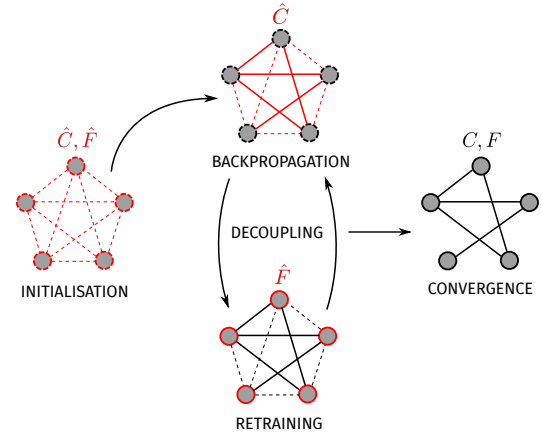


Fig. 3. Overview of the steps in the proposed backpropagation regression method. Components being trained (red) in each step. Regressed components are given with solid lines.

3 METHODS

This paper proposes a novel method of simultaneously inferring the local dynamics f and connectivity structure C of a dynamical network with a known coupling function g by only observing node states. The proposed method consists of three main stages, namely initialisation, backpropagation and decoupling. A flowchart outlining this process is provided (see Figure 3).

3.1 Backpropagation Regression Algorithm

Initialisation

The initialisation stage consists of constructing a local model $\hat{f}$ that approximates the true local dynamics f . Due to the complex interplay between the local f and coupling g dynamics, getting an accurate coupling-free estimation of f directly from node signals is not possible.

The estimated local model $\hat{f}$ is defined as a map from state space $\mathbf{x}(t)$ to vector field $\hat{\mathbf{x}}(t)$. The time-series is numerically differentiated in order to retrieve $\hat{\mathbf{x}}(t)$,

$$\hat{\mathbf{x}}(t) \approx \frac{\mathbf{x}(t + \delta t) - \mathbf{x}(t)}{\delta t}. \quad (6)$$

However, Equation 6 is only valid for a single isolated node and does not account for coupling effects with neighbouring nodes. For this stage of the method, we assume that the magnitude of coupling effects from g is relatively small compared to local dynamics f . By small, we mean that the 'attractor' of each node's time-series is not too distorted from the true attractor of the non-coupled case. This is because coupling with other neighbours in the dynamical network may cause spurious intersections of trajectories. This allows a mean field approach to be taken when constructing an approximate local model $\hat{f}$. Here, we calculate an estimate $\hat{\mathbf{x}}(t)$ of the true vector field at each point in state space as an average over its K nearest neighbours (see Figure 4),

$$\hat{\mathbf{x}}(t) = \frac{1}{K} \sum_{i=1}^K \frac{\mathbf{x}^{(i)}(t + \delta t) - \mathbf{x}^{(i)}(t)}{\delta t}, \quad (7)$$

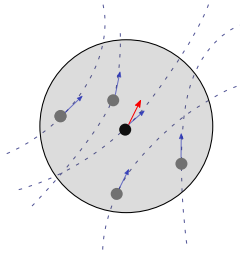


Fig. 4. Mean field approach where effective vector field (blue) at a point in phase space is taken as the average of K nearest neighbours' vector field influenced with neighbouring node interactions.

where $\dot{\mathbf{x}}^{(i)}(t)$ corresponds to the velocity of the i^{th} closest neighbour. The identification of neighbours are done with respect to the Euclidean distance with all points in the time-series (i.e. at all times t) being considered potential candidates of nearest neighbours.

The estimated local dynamics $\hat{f}$ is defined as a mapping $\hat{f} : \mathbf{x}(t) \rightarrow \hat{\mathbf{x}}(t)$. A feedforward network with 2 hidden layers was used to learn the mapping due to its simplicity and ease of implementation. Each hidden layer is composed of 128 neurons with a sigmoid activation function. The estimated local dynamics $\hat{f}$ calculated from the mean field approach acts as an initial proxy for the true local dynamics, which will then be further refined to remove coupling effects on the vector field during the decoupling and retraining stage of the algorithm. In addition to the local dynamics $\hat{f}$, values for the estimated coupling adjacency matrix $\hat{C}$ must also be initialised. However, there is no restriction on what values of $\hat{C}$ may be taken.

Backpropagation

The main aim of the backpropagation stage is to provide an improved estimate for the coupling adjacency matrix $\hat{C}$. Similar to the BPTT algorithm, a forward pass consisting of freerun predictions is first calculated using the estimated $\hat{f}$ and $\hat{C}$ with randomly selected initial values from the observed time-series in order to properly sample across the whole attractor,

$$\dot{\hat{\mathbf{x}}}_i(t) = \hat{f}(\mathbf{x}_i(t)) + \sum_{j \neq i} \hat{c}_{ij} g(\hat{\mathbf{x}}_i(t), \hat{\mathbf{x}}_j(t)), \quad (8a)$$

$$\hat{\mathbf{x}}_i(t + \delta t) = \hat{F}(\hat{\mathbf{x}}_i(t)) \quad (8b)$$

$$= \hat{\mathbf{x}}_i(t) + \delta t \dot{\hat{\mathbf{x}}}_i(t). \quad (8c)$$

For simplicity, we assume coupling only in the first component of each time-series.

Freerun predictions for all nodes $\hat{\mathbf{x}}_i(t)$ are calculated by recursively evaluating Equations 8a and 8b. The loss in freerun predictions due to errors in the local model $\hat{f}$ and network weights $\hat{C}$ can be calculated using some metric. Here, n is the length of the freerun prediction, N is the number of nodes in the dynamical network. This loss at each node, E_i , is the squared L^2 error across the total freerun trajectory $\{\hat{\mathbf{x}}_i(t_0), \dots, \hat{\mathbf{x}}_i(t_n)\}$,

$$\mathcal{L} = \sum_{t=t_0}^{t=t_n} \sum_{i=1}^N E_i(t) = \sum_{t=t_0}^{t=t_n} \sum_{i=1}^N \|\hat{\mathbf{x}}_i(t) - \mathbf{x}_i(t)\|^2. \quad (9)$$

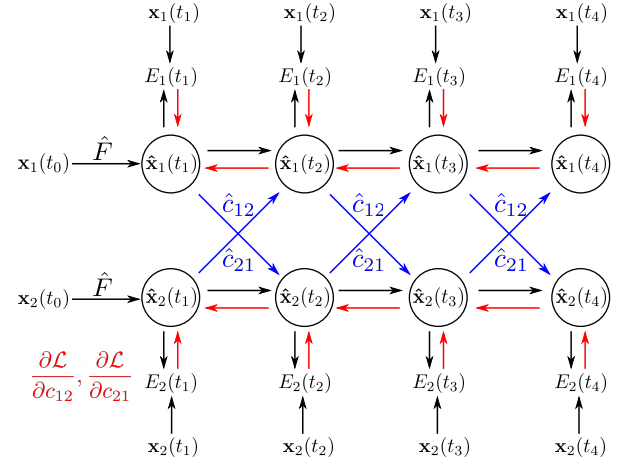


Fig. 5. Schematic of the two node modified backpropagation algorithm for calculating weight gradients. Forward pass (blue and black), and backpropagation (red).

For regressing network weights, the loss gradient with respect to $\hat{C}$ must be calculated and used to update the estimate of $\hat{C}$,

$$\frac{d\mathcal{L}}{d\hat{C}} = \sum_{t=t_0}^{t=t_n} \sum_{i=1}^N \frac{\partial E_i(t)}{\partial \hat{C}} = \sum_{t=t_0}^{t=t_n} \sum_{i=1}^N \sum_d 2(\hat{\mathbf{x}}_i^{(d)}(t) - \mathbf{x}_i^{(d)}(t)) \frac{\partial \hat{\mathbf{x}}_i^{(d)}(t)}{\partial \hat{C}}, \quad (10)$$

where $\hat{\mathbf{x}}_i^{(d)}(t)$ is the d^{th} component of the predicted state of node i at time t .

This requires the estimation of partial derivatives with respect to the coupling weights c_{ij} . The recursive expressions for these derivatives are given in the Appendix. To provide more stability in the regression, the gradient for each step of backpropagation is averaged over the multiple randomly chosen initial inputs $\mathbf{x}(t_0)$. An overview of the modified backpropagation algorithm is given in Figure 5 with a more detailed illustration given in the Appendix.

Decoupling and Retraining

The use of a mean field approximation to construct an initial local model $\hat{f}$ results in significant errors in the estimation of the vector field for the coupled dimensions. These errors limit the ability for the algorithm to estimate the true coupling weights as accurate regressions assume that $f = \hat{f}$. By using estimated coupling values $\hat{C}$ to partially decouple calculated values of the vector field, the output space of $\hat{f}$ changes to better reflect the true local dynamics f (i.e. without coupling effects). To adapt to this change, the model must be retrained on data of the new partially decoupled output space. The process of decoupling the vector field and truncating coupling interaction effects allows for gradual correction of the errors introduced from the mean field approximation of the vector field.

After attaining an improved estimate of the weighted adjacency matrix $\hat{C}$, the calculated coupling weights can be used to partially decouple the observed node time-series according to Equation 11.

$$\hat{\mathbf{x}}(t_n) \approx \frac{\mathbf{x}_i(t_{n+1}) - \mathbf{x}_i(t_n) - \delta t \sum_{j \neq i} \hat{c}_{ij} g(\mathbf{x}_i(t_n), \mathbf{x}_j(t_n))}{\delta t}, \quad (11)$$

for all selected nodes i . The new calculated value of the vector field $\hat{\mathbf{x}}$ can then be used to improve the previously constructed local dynamics model. The final decoupling step is alternated with the backpropagation step to gradually improve both the local model and estimate weighted adjacency matrix $\hat{C}_{ij}$. A much lower model learning rate $\eta' < \eta$ must be used to ensure that the structure of the initial model is preserved. Using a small learning rate also has the added benefit of providing additional stability to the algorithm as it minimises drastic changes in the loss landscape that can affect convergence.

3.2 Performance Assessment

Several measures need to be introduced to assess performance for the proposed backpropagation method. Because the backpropagation method aims to infer two separated components: the static network connectivity C and the local node dynamics f , the measures should assess the quality of each inference separately, as well as their performance in the combined network model.

The error in the inferred network connectivity $\hat{C}$ can be measured using the norm of the weight error matrix ϵ_C :

$$\epsilon_C = \frac{\|\hat{C} - C\|^2}{\|C\|^2}, \quad (12)$$

and is normalised by the true weight values C .

The quality of the inferred local dynamics can be measured by the prediction horizon and mutual information. Prediction horizon assesses the accuracy of predictions given by the combined model inclusive of local dynamics and coupling weights. The prediction horizon for node i was defined as the time $t_{p,i}$ where prediction errors exceed a set threshold ϵ . The overall prediction horizon t_p for the entire dynamical network is taken as the average prediction horizon across all nodes i ,

$$t_p = \langle t_{p,i} \mid \|\hat{\mathbf{x}}_i(t_{p,i}) - \mathbf{x}_i(t_{p,i})\| < \epsilon \rangle_{i=1}^N. \quad (13)$$

Prediction horizon is simple to calculate and provides an intuitive measurement of performance if the intended application is for exact future prediction. However, it is defined with respect to a hard limit ϵ and does not distinguish between more gradual prediction failures such as slow changes in phase, which may be temporary or permanent.

Mutual information $I(X, Y)$ is a nonlinear measure of the correlation between two random variables X and Y . In simple terms, describes the amount of information that is shared between any two random variables X and Y . Low mutual information implies independence between X and Y . Mutual information is usually calculated as a function of marginal and joint probabilities,

$$I(X, Y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} p_{(X,Y)}(x, y) \ln \left(\frac{p_{(X,Y)}(x, y)}{p_X(x)p_Y(y)} \right) dydx. \quad (14)$$

The quality of a model can be accessed by comparing the mutual information between the true states of a trajectory $\mathbf{x}(t)$, and those predicted by the model $\hat{\mathbf{x}}(t)$ at each point in time t . Hence, we can define the average mutual information

$I(\tau)$ with respect to some chosen freerun prediction length τ ,

$$I(\tau) = \frac{1}{\tau} \sum_{t=1}^{\tau} p(\mathbf{x}(t), \hat{\mathbf{x}}(t)) \ln \left(\frac{p(\mathbf{x}(t), \hat{\mathbf{x}}(t))}{p(\mathbf{x}(t))p(\hat{\mathbf{x}}(t))} \right), \quad (15)$$

where $p(\mathbf{x}(t))$, $p(\hat{\mathbf{x}}(t))$ and $p(\mathbf{x}(t), \hat{\mathbf{x}}(t))$ are the joint and marginal probabilities of observing (or predicting for the case of $\hat{\mathbf{x}}$) a given state at time t . For each value of τ in $I(\tau)$, the distributions of $p(\mathbf{x}(t))$, $p(\hat{\mathbf{x}}(t))$ and $p(\mathbf{x}(t), \hat{\mathbf{x}}(t))$ are calculated empirically using histograms based on observations of $\mathbf{x}(t)$ and $\hat{\mathbf{x}}(t)$ for $t \in [1, \tau]$. For chaotic systems, $I(\tau)$ of an imperfect model will be a decreasing function. The contour of $I(\tau)$ tracks the collapse of the local model's prediction performance for increasing prediction lengths τ .

The performance of a trained local model $\hat{f}$ can be assessed by using it to produce freerun predictions of separate system with only a single node (i.e. no external interactions). A score S related to the mutual information between the trajectories from the real local model $\mathbf{x}(t)$ and trained local model $\hat{\mathbf{x}}(t)$ can be calculated using the cumulative area under the average mutual information curve,

$$S = \int_1^{\tau_{max}} I(\tau) d\tau, \quad (16)$$

The score S can also be interpreted as a measure of the prediction performance of the model for a prediction length of τ steps where larger values of S indicate more persistent mutual information and a better predictive local model. Because the score S is calculated as a sum across τ , it is able account for the limitations in prediction horizon and better describe how well the dynamics of the local model reflects the real system.

The mutual information based score S is only calculated with respect to only one component of the predicted time-series (i.e. x component). This is sufficient for the systems analysed as all components of the time-series (x, y, z) feedback to each other. Hence, collapse in the prediction performance of one component will quickly result in poor predictions in all other components. Additionally, $I(\tau)$ for each component cannot be summed together as it assumes that all components evolve independently.

To assess the mutual information of the combined model of both regressed weights and trained local model, S is calculated as the sum of the scores across all nodes. Whilst nodes are not independent, the sparsity of network connections mean that collapse in the prediction performance of one node does necessarily propagate quickly to all other nodes in the network.

4 RESULTS

Three tests were done to assess the performance of the proposed adapted backpropagation regression algorithm. These were the cases of the small 16 node network, large 64 node network, and networks with non-positive asymmetric coupling weights. All tests were conducted on random networks with each edge existing with probability $p = \log(N)/N$ to minimise the likelihood of disjoint networks. Edges are bidirectional in all networks except the asymmetric network case. Values for the node coupling strengths were normally distributed $(\mu, \sigma) = (0.15, 0.02)$.

The mean coupling strength μ is assumed to be non-zero to ensure that regressed values clearly distinguish between the absence and presence of edges.

Input data of 25000 steps with an additional 2000 steps washout were generated with time step $dt = 0.02$ using the standard RK4 algorithm. Washout steps were discarded exclude any transient dynamics. An initial learning rate of $\eta = 0.001$ was used for training the feedforward model during initialisation. A lower learning rate of $\eta' = 0.0002$ was used for all subsequent retraining iterations with each training period consisting of 30 epochs. A preset number of 40 or 80 refit iterations were chosen before ending regression. However, we note that it may be possible to implement a stopping criterion based on plateauing of either the prediction errors or regressed weight matrix $\hat{C}$.

Performance measures (S, t_p) for each model were evaluated over 8 randomly chosen initial conditions. To assess the performance of regressed models, each performance measure was compared against a control case where the entire network structure and local model is fully known but initial conditions are perturbed by $\xi \sim N(0, 0.005^2)$ where ξ is in scaled units. This method was chosen over the calculation of dynamical invariants such as Lyapunov exponents due to its ease of computation, especially in high dimensional systems such as dynamical networks. Due the chaotic nature of the system's oscillators, small deviations in initial conditions will grow exponentially. Deviations the initial conditions that are sufficiently small are able to track nearby trajectories for some period of time before diverging. Therefore, scores that match those of the control case imply that the regressed models are able to perform almost as well as the case where the system dynamics are fully known, but contaminated with noise in its initial conditions.

4.1 Positive Symmetric Networks

The fully observable case assumes that all node time-series are accessible for regression. Gradient losses $\mathcal{L}$ are calculated at each backpropagation step and used to regress the network adjacency matrix $\hat{C}$. The regression algorithm was run for a total of 40 refit iterations each consisting of 300 backpropagation steps followed by decoupling and retraining of the model. A total of 8 randomly generated 16 node networks were tested. For each network, the quality of the regressed weights and local model was assessed by running freerun predictions with 8 randomly generated initial conditions.

The backpropagation method was able to accurately regress the weights of both the Lorenz (Figure 6) and Chua (Figure 8) dynamical networks. The average normalised error in the regressed weights for these systems at the end of 30 iterations was 9% and 7% for the Lorenz and Chua networks respectively. The constructed trained local model for both systems also showed a large improvement in S with 71% increase for Lorenz and a 38% increase for Chua when compared to the initial local model based purely on the mean field approach (i.e. Iteration 0). The mutual information of the combined local model and coupling structure increased by 65% and 59% for the Lorenz and Chua networks respectively. The prediction horizons for both systems were found to increase with each additional iteration. Both Chua

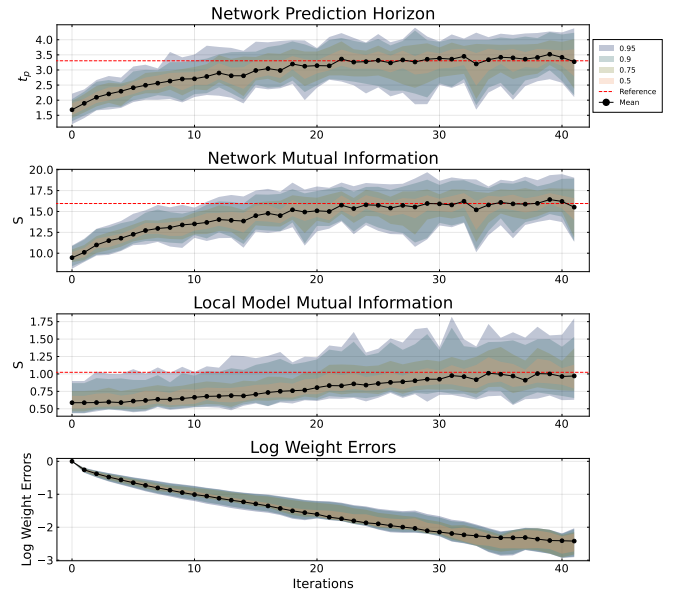


Fig. 6. Regression results for the Lorenz 16 node system normally distributed coupling weights $(\mu, \sigma^2) = (0.15, 0.02)$ and connection probability p over 40 iterations. First two data points correspond to the values attained during initialisation. Coloured bands contain the 95%, 90%, 75% and 50% quantiles across 64 different combinations of dynamical network and random initial conditions. The result from the noise control case is in red.

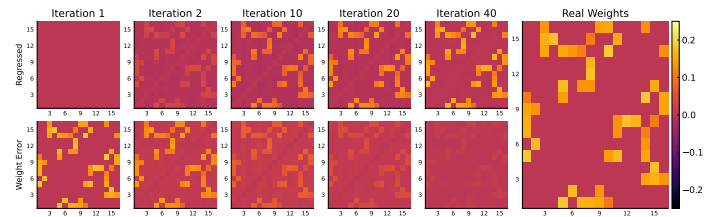


Fig. 7. Regression of Lorenz 16 node network weights (top) at different number of iterations compared to the error in the weights (bottom). True weights (right) are given for comparison showing good agreement with the regressed results. Array entries coloured based on value of weights and error.

and Lorenz networks were found to perform as well or better than the control case where $\xi = 0.005$.

4.2 Large 64-node Network

A single larger system of 64 random coupled Lorenz oscillators was used to test the scalability of the system. The algorithm was run for 80 iterations to account for slightly slower convergence. Performance metrics identical to the regular 16 node case was used with the network tested against 16 different initial conditions (see Figures 10 and 11).

We find that the backpropagation method was also able to accurately regress the desired weights of the coupled system with a final normalised weight error of 9%. The mutual information of the local model and combined network system was also found to increase by 51% and 81% by the end of the iterations respectively.

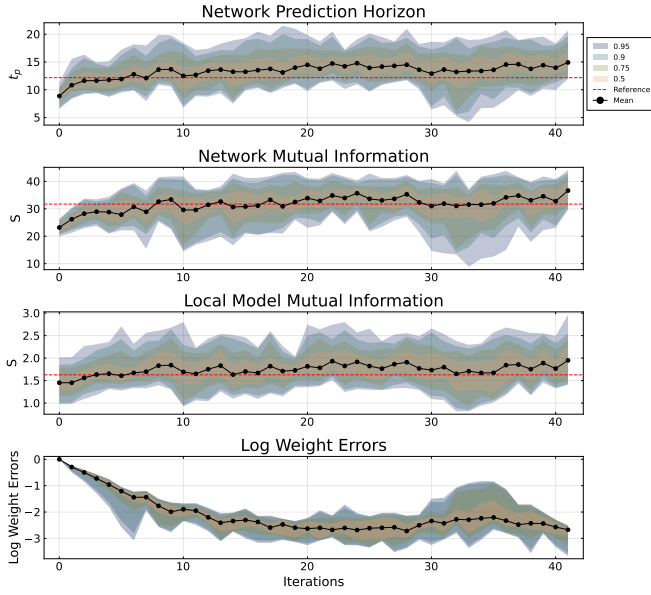


Fig. 8. Regression results for the Chua 16 node system operating in the single scroll regime. Coupling weights are normally distributed with $(\mu, \sigma^2) = (0.15, 0.02)$ and connection probability p over 40 iterations. First two data points correspond to the values attained during initialisation. Coloured bands contain the 95%, 90%, 75% and 50% quantiles across 64 different combinations of dynamical network and random initial conditions. The result from the noise control case is in red.

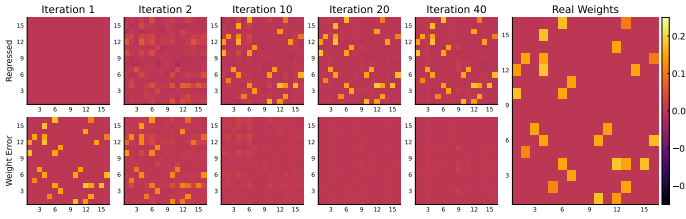


Fig. 9. Regression of Chua 16 node network weights (top) at different number of iterations compared to the error in the weights (bottom). True weights (right) are given for comparison showing good agreement with the regressed results. Array entries coloured based on value of weights and error.

4.3 Negative Asymmetric Coupling

Many dynamical systems assume the case of positive coupling resulting in the attraction of nearby trajectories. In addition to this common case, we consider dynamical systems with both negative and positive coupling where repulsive behaviour between nodes is possible. Additionally, we also consider the effects where the adjacency matrix is not symmetrical. To do this, a 16 node adjacency matrix was generated with each non diagonal entry being non-zero with a probability $p = \log(N)/N$ and is not guaranteed to be symmetric. However, each node has a 0.25 probability of being changed to a negative value. The time-series was then generated similarly to the original test case with the Lorenz equations (see Figures 12 and 13).

The backpropagation method was found to also work equally well in regressing cases with negative weights and asymmetric coupling. The results for the network prediction horizon, log weight errors and mutual information measures show a similar trend to their symmetric, positively

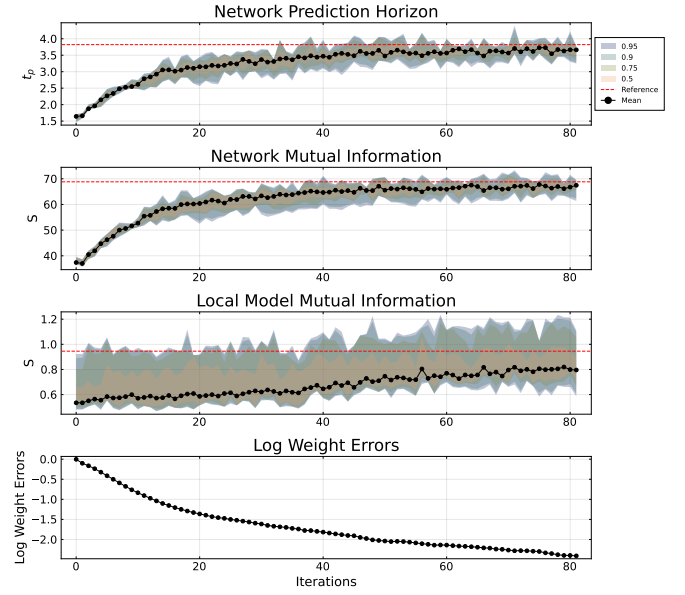


Fig. 10. Regression results for the larger Lorenz 64 node system normally distributed coupling weights $(\mu, \sigma^2) = (0.15, 0.02)$ and connection probability p over 80 iterations. First two data points correspond to the values attained during initialisation. Coloured bands contain the 95%, 90%, 75% and 50% quantiles across 8 random initial conditions. The result from the noise control case is in red.

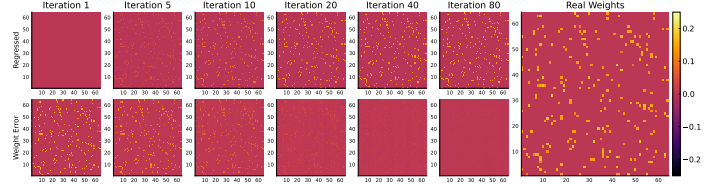


Fig. 11. Regression of Lorenz 64 node network weights (top) at different number of iterations compared to the error in the weights (bottom). True weights (right) are given for comparison showing good agreement with the regressed results. Array entries coloured based on value of weights and error.

weighted counterparts with a final normalised weight error of 5%, and an 58% relative increase in mutual information for the constructed local model. The mutual information of the combined network was found to increase by 75% when compared to the first iteration.

4.4 Noise Effects

In order to test the robustness of the method, the backpropagation approach was applied to time-series from a 16 node Lorenz dynamical network with additive Gaussian noise to simulate measurement noise. The Gaussian noise term $\epsilon_\xi(t) \sim N(0, \xi)$ was added after normalising the data to zero mean and unit standard deviation. Because backpropagation regression requires the evaluation of numerical derivatives, the noisy input data was first smoothed using spline regression [49], [50] with a regularisation parameter $\lambda = 10$. The usage of cubic splines also has the advantage of producing relatively smooth derivatives.

The noise magnitude was varied with $\xi \in [0, 0.1]$ with the maximum ξ value corresponding to a signal-noise ratio (SNR) of 20dB. Backpropagation regression was run for 40

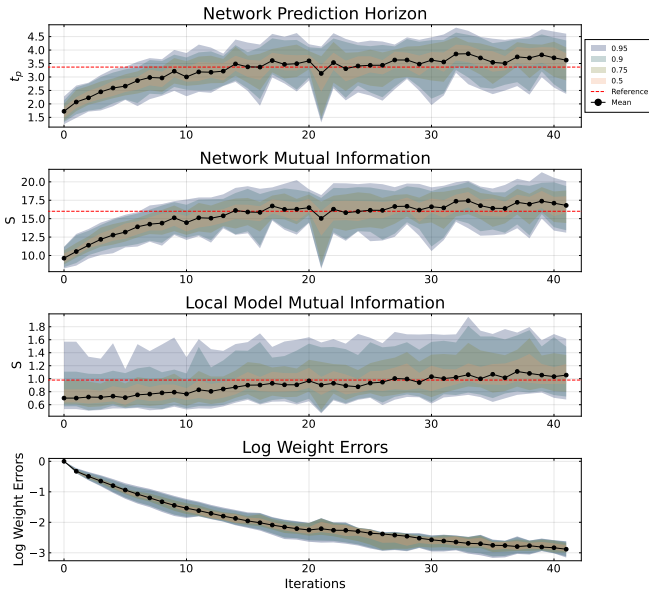


Fig. 12. Regression results for the Lorenz 16 node system normally distributed coupling weights $(\mu, \sigma^2) = (0.15, 0.02)$ and connection probability p over 40 iterations. Weights are asymmetric and changed to its negative value with probability of 0.25. First two data points correspond to the values attained during initialisation. Coloured bands contain the 95%, 90%, 75% and 50% quantiles across 64 different combinations of dynamical network and random initial conditions. The result from the noise control case is in red.

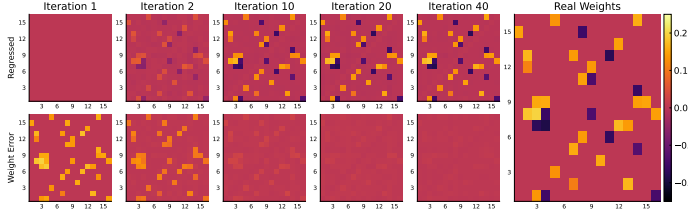


Fig. 13. Regression of Lorenz 16 node network with negative and positive weights (top) at different number of iterations compared to the error in the weights (bottom). True weights (right) are given for comparison showing good agreement with the regressed results. Array entries coloured based on value of weights and error.

iterations in line with those for the noiseless Lorenz and Chua 16 node systems. Each configuration was repeated for 8 randomly chosen network and initial conditions.

The effect of noise hampers performance and results in the addition of spurious incorrect weights during regression (see Figure 14). To address this, a threshold value of 0.04 is selected to filter the weights following regression. Regressed weights that do not exceed this value are set to 0. The threshold value is visually selected from the histogram distribution of weights. As the network weights are normally distributed with non-zero mean, $c_{ij} \sim N(0.15, 0.02)$, true weights will tend to have regressed value much greater than zero. In contrast, noise effects will yield to low valued spurious weights. This results in a bimodal distribution of weights. Hence, the threshold value can be selected as a value that distinguishes between the two modes.

The performance of the regression algorithm weights was found to be relatively robust against noise. For low

noise levels $\xi \leq 0.04$, the effects of noise appear follow a linear relationship (see Figure 14). The increase in weight errors ϵ_C slow for higher levels of noise. In the latter case, the weight filtration was found to be very effective in removing spurious edge weights allowing a further decrease in weight errors. However, we found that higher values of noise with $\xi \geq 0.15$ resulted in instability of the regression algorithm and divergence in the weight errors.

The effect of noise was found to significantly affect the quality of the local dynamics models (see Figure 15). For small noise levels $\xi \leq 0.01$, the approximations $\hat{f}$ of the local dynamics was relatively stable. However, there was a large difference in prediction performance between the noiseless and noisy case. Noise effects are more detrimental for $\xi > 0.01$ where the long run predictions of the resulting models no longer conform to the attractor of the original Lorenz system. In several instances, it was found that the models slowly drifted and converged to a fixed point. For $\xi \geq 0.07$, the observational noise results in a local model that is worse than the initial mean field estimate.

The poor learning of the local model can be attributed to the combined effects of noise changing the temporal structure of the time-series and the relatively low sampling rates of the time-series. An integration step of $dt = 0.02$ corresponds to approximately 35 points per oscillation. Each oscillation consists of a short transient extrema. Additive noise result in over or underestimation of these peaks, which can have a detrimental effect on calculated vector field. The noise tests were repeated at a lower integration step of $dt = 0.002$, which was smoothed and downsampled to $dt = 0.02$. This configuration allowed a better treatment of the additive noise and yielded more robust results.

We find that the performance of this method under noise is comparable to other methods used to analysed dynamical networks. For local dynamics, the attractor of the regressed local model dynamics began to collapse at $\xi = 0.05$ corresponding to an SNR of 26dB. This is comparable to the test results conducted by [11] where the methods SINDy, ARNI and Gao's method exhibit large decreases in performance at approximately 55dB, 40dB and 25dB respectively.

5 METHOD LIMITATIONS

There are several data and system requirements needed for the application of the backpropagation regression algorithm. Firstly, the backpropagation regression method requires networks that are not fully synchronised. This is because differences in the trajectories of coupled nodes are used to decouple observed node signals to infer links. Networks operating in the fully synchronised regime will have fully coincident trajectories and cannot be used to infer weights.

Secondly, reliance of a mean field approximation during the initialisation stage assumes that node trajectories do not deviate too much from the attractor of the local dynamics of the uncoupled single node case. Small deviations will inevitably occur, and is accounted for during the decoupling and retraining stage of each iteration. However, deviations that are too large may cause convergence problems.

Thirdly, there must be sufficient full node state observations with trajectories that fully explore the state space for the regression to converge and produce a good local model.

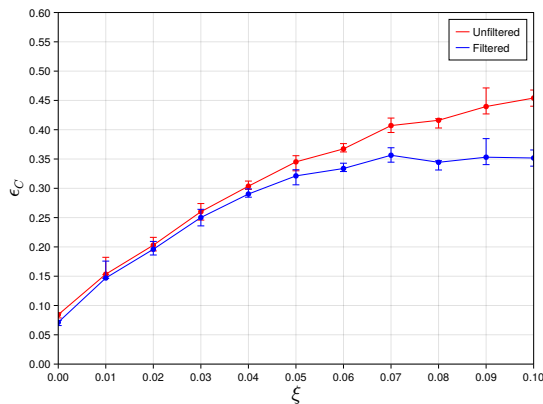


Fig. 14. Weight errors ϵ_C after 40 iterations for backpropagation regression applied to 16 node Lorenz dynamical networks with varying levels of additive Gaussian noise. Each noise configuration was repeated with 8 randomly chosen networks and initial conditions. Error bars correspond to interquartile range of weight errors. Weights with a magnitude less than a threshold 0.04 are truncated and set to zero. The threshold value selected from the histogram distribution of all weights.

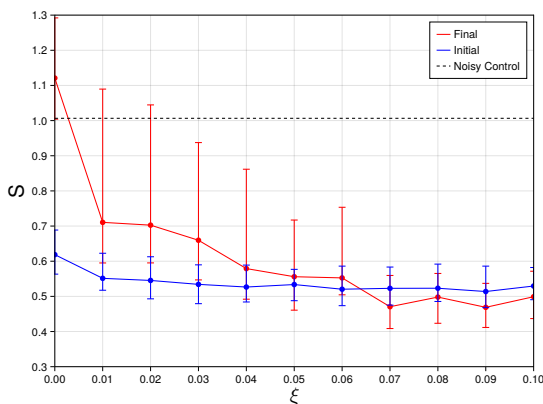


Fig. 15. Mutual information of the local models for configurations with increasing observational noise ξ . Model scores are compared at the beginning (blue) and end (red) of the 40 refit iterations. Mutual information is compared against the noisy control case where the exact model known with perturbed initial conditions.

The method was found to be convergent with time-series lengths as low as 5000 time steps with $dt = 0.02$, excluding an initial washout period of 2000 time steps to account for dynamical transients.

6 COMPUTATION TIME AND SCALABILITY

The backpropagation regression algorithm inherits similar computational difficulties as the BPTT algorithm. The most expensive operation is the calculation of error gradients in each regression step, which has a complexity of approximately $\mathcal{O}(t_{in}N^4)$. The original BPTT algorithm allows a speedup of gradient calculations by predefining expressions for the derivatives of the activation function. In contrast, the backpropagation regression algorithm for dynamical networks is defined with respect to an arbitrary local model as the activation function, which may not have a closed form and derivatives must be calculated using finite differences. We note that if a feedforward neural

network is used as the local model with well-defined activation functions in each layer, the calculation of partial derivatives with respect to each input may be sped up significantly. However, this has not been implemented in the current algorithm. There are also additional potential gains in computation speed that can be achieved via parallelisation of some parts of the algorithm. A copy of the algorithm is available at: <https://github.com/eugenetkj98/BackpropagationRegressionPublic>

Computation benchmarks were done on system with a pair of 20 Core Intel Xeon Gold 6242R 3.1 GHz CPUs with hyperthreading enabled and 512 GB RAM. Neural network computation was completed with an NVIDIA Quadro RTX6000 GPU. Code was written in Julia 1.7.3 and the machine learning package, Flux. The 16 node dynamical networks with 10 step free run prediction for regression and 40 refit iterations required a total of 20 minutes to complete. The 64 node dynamical network with the same regression settings required a total of 42 hours to complete.

7 CONCLUSION

Dynamical networks are an interesting and useful framework to analyse complex systems with multiple interacting components. However, application of this framework is usually difficult due to lack of information pertaining to features such as the connectivity structure and local dynamics. Therefore, there is a need for a method to extract these features purely from observed node states.

In this paper, we propose an adapted backpropagation method for inferring the local model and connectivity structure of dynamical networks purely from node time-series. The result is an estimate of the inferred network structure and a local dynamics model that is disentangled from the coupling interactions. This method draws inspiration from the backpropagation through time (BPTT) algorithm commonly used to train RNNs and adapts it for the purpose of regressing couplings weights in a dynamical network. Regressed values for the network coupling are then used to decouple the observed node time-series and improve the construction of the local model without coupling effects. The two steps of backpropagation weight regression and decoupling alternate until sufficient convergence is achieved yielding a final local model and regressed coupling weights.

The method was tested on dynamical networks of nodes of identical chaotic oscillators with successful results. Two types of chaotic oscillators, Lorenz and single-scrolled Chua, were tested. The backpropagation regression method was able to accurately identify the coupling weights down to a margin of 7%. The regressed models were also compared to a control case where the true local model is known, but whose initial states are perturbed by a small error. It was found that the trained local model was comparable to the control case. Combined, the regressed weights and constructed local model were able to achieve respectable freerun prediction horizons for the 16 node dynamical network compared to a control case. Due to the chaotic nature of the network dynamics, long-term predictions are typically unfeasible. However, the ability to forecast accurate short-term predictions can be useful for inferring potential

cascading failures or imminent undesirable behaviour. This may be used to inform preemptive intervention measures.

Despite its poor scalability, the method was also able to accurately reconstruct the local model and coupling weights of a larger 64 node network with similar results to the smaller 16 node counterparts. Additional testing was also done on dynamical networks with negative and asymmetric coupling with similar performance.

The robustness of the method was also tested against noise and network heterogeneity (see Appendix). The algorithm was found to be convergent for small to moderate amounts of observational noise. However, larger noise levels resulted in poor local models and potential instability. In contrast, network heterogeneity did not significantly affect the performance of the backpropagation algorithm.

We acknowledge that the method proposed in this paper is a proof of concept that, whilst functional, has multiple areas of further research and potential refinement. Firstly, there remains further work on the refinement of the speed and complexity of the algorithm to be more scalable. Secondly, there remains a need for further testing the on the effect of hyperparameters and initialisation on algorithm performance. Thirdly, it would be beneficial to investigate the performance of this method on different dynamical network structures. There also remains the possibility of investigating how other machine learning methods used in recurrent neural networks may be applied to the backpropagation regression algorithm.

ACKNOWLEDGMENTS

E.T. is supported by a Robert and Maude Gledden Postgraduate Research Scholarship and Australian Government RTP Scholarship at The University of Western Australia. M.S. and D.C.C. acknowledge the support of the Australian Research Council through the Centre for Transforming Maintenance through Data Science (grant number IC180100030), funded by the Australian Government. M.S. is also supported by the ARC Discovery Grant DP200102961, funded by the Australian Government.

REFERENCES

- [1] I. Brugere, B. Gallagher, and T. Y. Berger-Wolf, "Network structure inference, a survey: Motivations, methods, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–39, 2018.
- [2] D. M. Abrams and S. H. Strogatz, "Chimera states for coupled oscillators," *Physical Review Letters*, vol. 93, no. 17, p. 174102, 2004.
- [3] A. Andreev, N. Frolov, A. Pisarchik, and A. Hramov, "Chimera state in complex networks of bistable Hodgkin-Huxley neurons," *Physical Review E*, vol. 100, no. 2, p. 022224, 2019.
- [4] Y. Zhu, Z. Zheng, and J. Yang, "Chimera states on complex networks," *Physical Review E*, vol. 89, no. 2, p. 022914, 2014.
- [5] S. Majhi, B. K. Bera, D. Ghosh, and M. Perc, "Chimera states in neuronal networks: a review," *Physics of Life Reviews*, vol. 28, pp. 100–121, 2019.
- [6] E. M. Izhikevich, *Dynamical Systems in Neuroscience*. MIT press, 2007.
- [7] B. Wu, D. Zhou, F. Fu, Q. Luo, L. Wang, and A. Traulsen, "Evolution of cooperation on stochastic dynamical networks," *PLoS One*, vol. 5, no. 6, p. e11187, 2010.
- [8] T. Liu, J. Zhao, and D. J. Hill, "Exponential synchronization of complex delayed dynamical networks with switching topology," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 11, pp. 2967–2980, 2010.
- [9] A. R. Hota, T. Sneh, and K. Gupta, "Impacts of game-theoretic activation on epidemic spread over dynamical networks," *SIAM Journal on Control and Optimization*, no. 0, pp. S92–S118, 2021.
- [10] M. Rosenblum, M. Frühwirth, M. Moser, and A. Pikovsky, "Dynamical disentanglement in an analysis of oscillatory systems: an application to respiratory sinus arrhythmia," *Philosophical Transactions of the Royal Society A*, vol. 377, no. 2160, p. 20190045, 2019.
- [11] T.-T. Gao and G. Yan, "Autonomous inference of complex network dynamics from incomplete and noisy data," *Nature Computational Science*, vol. 2, no. 3, pp. 160–168, 2022.
- [12] M. Breakspear, "Dynamic models of large-scale brain activity," *Nature Neuroscience*, vol. 20, no. 3, pp. 340–352, 2017.
- [13] T.-T. Gao and G. Yan, "Data-driven inference of complex system dynamics: A mini-review," *Europhysics Letters*, 2023.
- [14] S. G. Shandilya and M. Timme, "Inferring network topology from complex dynamics," *New Journal of Physics*, vol. 13, no. 1, p. 013004, 2011.
- [15] G. Stepaniants, B. W. Brunton, and J. N. Kutz, "Inferring causal networks of dynamical systems through transient dynamics and perturbation," *Physical Review E*, vol. 102, no. 4, p. 042309, 2020.
- [16] A. Banerjee, J. Pathak, R. Roy, J. G. Restrepo, and E. Ott, "Using machine learning to assess short term causal dependence and infer network links," *Chaos*, vol. 29, no. 12, p. 121104, 2019.
- [17] A. Banerjee, J. D. Hart, R. Roy, and E. Ott, "Machine learning link inference of noisy delay-coupled networks with optoelectronic experimental tests," *Physical Review X*, vol. 11, no. 3, p. 031014, 2021.
- [18] D. Napoletani and T. D. Sauer, "Reconstructing the topology of sparsely connected dynamical networks," *Physical Review E*, vol. 77, no. 2, p. 026103, 2008.
- [19] C. Weistuch, L. Agozzino, L. R. Mujica-Parodi, and K. A. Dill, "Inferring a network from dynamical signals at its nodes," *PLoS Computational Biology*, vol. 16, no. 11, 2020.
- [20] M. Paluš, "Coupling in complex systems as information transfer across time scales," *Philosophical Transactions of the Royal Society A*, vol. 377, no. 2160, p. 20190094, 2019.
- [21] M. Kornilov, I. Sysoev, D. Astakhova, D. Kulminsky, B. Bezruchko, and V. Ponomarenko, "Reconstruction of the Coupling Architecture in the Ensembles of Radio-Engineering Oscillators by Their Signals Using the Methods of Granger Causality and Partial Directed Coherence," *Radiophysics and Quantum Electronics*, vol. 63, no. 7, pp. 542–556, 2020.
- [22] X. Wu, W. Wang, and W. X. Zheng, "Inferring topologies of complex networks with hidden variables," *Physical Review E*, vol. 86, no. 4, p. 046106, 2012.
- [23] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*. Springer, 1981, pp. 366–381.
- [24] J.-S. Zhang and X.-C. Xiao, "Predicting chaotic time series using recurrent neural network," *Chinese Physics Letters*, vol. 17, no. 2, p. 88, 2000.
- [25] M. Small and C. K. Tse, "Minimum description length neural networks for time series prediction," *Physical Review E*, vol. 66, no. 6, p. 066701, 2002.
- [26] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD) Technical Report*, vol. 148, p. 13, 2001.
- [27] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," *Chaos*, vol. 27, no. 12, p. 121102, 2017.
- [28] Z. Lu, B. R. Hunt, and E. Ott, "Attractor reconstruction by machine learning," *Chaos*, vol. 28, no. 6, p. 061104, 2018.
- [29] A. Haluszczynski and C. Räth, "Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing," *Chaos*, vol. 29, no. 10, p. 103143, 2019.
- [30] E. Tan, D. Corrêa, T. Stemler, and M. Small, "Grading your models: Assessing dynamics learning of models using persistent homology," *Chaos*, vol. 31, no. 12, p. 123109, 2021.
- [31] M. J. Panaggio, M.-V. Ciocanel, L. Lazarus, C. M. Topaz, and B. Xu, "Model reconstruction from temporal data for coupled oscillator networks," *Chaos*, vol. 29, no. 10, p. 103116, 2019.
- [32] W.-X. Wang, Y.-C. Lai, and C. Grebogi, "Data based identification and prediction of nonlinear and complex dynamical systems," *Physics Reports*, vol. 644, pp. 1–76, 2016.

- [33] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [34] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and M. A. F. Harrison, "Time-series-based prediction of complex oscillator networks via compressive sensing," *EPL (Europhysics Letters)*, vol. 94, no. 4, p. 48006, 2011.
- [35] J. Casadiego, M. Nitzan, S. Hallerberg, and M. Timme, "Model-free inference of direct network interactions from nonlinear collective dynamics," *Nature Communications*, vol. 8, no. 1, pp. 1–10, 2017.
- [36] D. Eroglu, M. Tanzi, S. van Strien, and T. Pereira, "Revealing dynamics, communities, and criticality from data," *Physical Review X*, vol. 10, no. 2, p. 021047, 2020.
- [37] I. Topal and D. Eroglu, "Reconstructing network dynamics of coupled discrete chaotic units from data," *Physical Review Letters*, vol. 130, no. 11, p. 117401, 2023.
- [38] M. Novaes, E. R. dos Santos, and T. Pereira, "Recovering sparse networks: Basis adaptation and stability under extensions," *Physica D: Nonlinear Phenomena*, vol. 424, p. 132895, 2021.
- [39] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2688–2697.
- [40] M. Cranmer, A. Sanchez Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, "Discovering symbolic models from deep learning with inductive biases," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 429–17 442, 2020.
- [41] Y. Zhang, Y. Guo, Z. Zhang, M. Chen, S. Wang, and J. Zhang, "Universal framework for reconstructing complex networks and node dynamics from discrete or continuous dynamics data," *Physical Review E*, vol. 106, no. 3, p. 034315, 2022.
- [42] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [43] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.
- [44] J. Kengne, "On the dynamics of Chua's oscillator with a smooth cubic nonlinearity: Occurrence of multiple attractors," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 363–375, 2017.
- [45] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos*. Academic Press, 2012.
- [46] S. Raubitzek and T. Neubauer, "Taming the chaos in neural network time series predictions," *Entropy*, vol. 23, no. 11, p. 1424, 2021.
- [47] P. Dubois, T. Gomez, L. Planckaert, and L. Perret, "Data-driven predictions of the Lorenz system," *Physica D: Nonlinear Phenomena*, vol. 408, p. 132495, 2020.
- [48] M. Sangiorgio and F. Dercole, "Robustness of LSTM neural networks for multi-step forecasting of chaotic time series," *Chaos, Solitons and Fractals*, vol. 139, p. 110045, 2020.
- [49] P. J. Green and B. W. Silverman, *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press, 1993.
- [50] C. H. Reinsch, "Smoothing by spline functions," *Numerische Mathematik*, vol. 10, no. 3, pp. 177–183, 1967.

Michael Small is a professor and CSIRO-UWA Chair in Complex Engineering Systems at the University of Western Australia. He completed his Ph.D. in applied mathematics focusing on nonlinear dynamical systems and time-series analysis at The University of Western Australia.

Eugene Tan is a Robert and Maude Gledden scholar and PhD candidate at the University of Western Australia. He completed his B.Sc and MPE in mechanical engineering at the University of Western Australia.

Débora Corrêa completed her Ph.D. degree in Applied Physics with a focus on Computational Physics at the University of Sao Paulo (USP). She is currently a lecturer at the University of Western Australia

Thomas Stemler completed his Ph.D. at the Technische Universität Darmstadt. He is currently a senior lecturer at the University of Western Australia.