Neural Granger Causality for Nonlinear Time Series

Alex Tank
Department of Statistics
University of Washington

Ian Covert, Nicholas J. Foti
Department of Computer Science
University of Washington

Ali Shojaie
Department of Biostatistics
University of Washington

Emily B. Fox
Department of Computer Science
Department of Statistics
University of Washington

February 19, 2018

Abstract

While most classical approaches to Granger causality detection assume linear dynamics, many interactions in applied domains, like neuroscience and genomics, are inherently nonlinear. In these cases, using linear models may lead to inconsistent estimation of Granger causal interactions. We propose a class of nonlinear methods by applying structured multilayer perceptrons (MLPs) or recurrent neural networks (RNNs) combined with sparsity-inducing penalties on the weights. By encouraging specific sets of weights to be zero—in particular through the use of convex group-lasso penalties—we can extract the Granger causal structure. To further contrast with traditional approaches, our framework naturally enables us to efficiently capture long-range dependencies between series either via our RNNs or through an automatic lag selection in the MLP. We show that our neural Granger causality methods outperform state-of-the-art nonlinear Granger causality methods on the DREAM3 challenge data. This data consists of nonlinear gene expression and regulation time courses with only a limited number of time points. The successes we show in this challenging dataset provide a powerful example of how deep learning can be useful in cases that go beyond prediction on large datasets. We likewise demonstrate our methods in detecting nonlinear interactions in a human motion capture dataset.

1 Introduction

Granger causality quantifies the extent to which the past activity of one time series is predictive of another time series. When an entire system of time series is studied, networks of interactions may be uncovered [1]. Classically, most methods for estimating Granger causality assume linear time series dynamics and utilize the popular vector autoregressive (VAR) model [2, 3]. However, many real world systems exhibit *nonlinear* dependence between series so that using linear models may lead to inconsistent estimation of Granger causal interactions [4, 5, 6]. Common nonlinear approaches to detecting interactions in time series use additive models, where the past of each series may have an additive nonlinear effect that decouples across series [4, 7, 8]. However, additive models may miss important nonlinear interactions between predictors so they may also fail to detect important Granger causal connections.

To tackle these challenges we present a framework for interpretable nonlinear Granger causality discovery by augmenting neural networks with sparsity inducing penalties on the weights. Neural network models for time series analysis are traditionally used only for prediction and forecasting —

not for interpretation. This is due to the fact that the effects of inputs are difficult to quantify exactly due to the tangled web of interacting nodes in the hidden layers. We mitigate this difficulty through two steps. First, we consider *component-wise* architectures, one using a multilayer perceptron (MLP) and the other using a long-short term memory (LSTM) recurrent network [9], that disentangle the effects of lagged inputs on individual output series. Second, we place sparsity-inducing penalties on particular groupings of the weights that relate the histories of individual series to the output series of interest, thus allowing us to precisely select for time series that have no nonlinear Granger effects. We term these *sparse component-wise* models, e.g. cMLP and cLSTM.

In particular, we select for Granger causality by adding *group sparsity* penalties [10] on the outgoing weights of the inputs. For the MLP, we utilize a hierarchical group lasso penalty [11] that automatically detects both nonlinear Granger causality and also the lag of each inferred interaction. For the LSTM, we use a single group lasso penalty across all outgoing weights for each series input. When the true network of nonlinear interactions is sparse, these approaches will select a subset of the time series that Granger cause the output series. To our knowledge, these approaches represent the first set of nonlinear Granger causality methods where precise lag specification is not necessary: with the cMLP, the hierarchical penalty performs series-specific lag selection; with the cLSTM, the recurrent architecture efficiently models long range dependencies [9].

We first validate our approach and associated penalties via simulations on both linear VAR and nonlinear Lorenz-96 data [12], showing that our nonparametric approach accurately selects the Granger causality graph in both linear and nonlinear settings. Second, we compare our cMLP and cLSTM models to existing Granger causality approaches [13, 14] on the difficult DREAM3 gene expression network recovery datasets [15] and find that our methods outperform a wide set of competitors across all five datasets in the challenge. Finally, we use our cLSTM method to explore causal interactions between body parts during natural motion with a highly nonlinear and complex dataset of human motion capture [16, 17]

Traditionally, the success stories of neural networks have been on prediction tasks in large datasets. In contrast, here our performance metrics relate to our ability to produce interpretable structures of interaction amongst the observed time series. Furthermore, these successes are achieved in limited data scenarios. Our ability to produce interpretable structures and train neural network models with limited data can be attributed to our use of sparsity-inducing penalties and the regularization such penalties provide, respectively. We note that sparsity inducing penalties have been used for architecture selection in neural networks [18, 19]. However, the focus of the architecture selection was on improving predictive performance rather than on returning interpretable structures of interaction amongst observed quantities. In concurrent work, a similar notion of sparse-input neural networks were developed for high-dimensional regression and classification tasks for independent data [20].

2   Background and Problem Formulation

Let $x_t \in \mathbb{R}^p$ denote a p-dimensional stationary time series and assume we have observed the process at T time points, $(x_1, \ldots, x_T)$. Granger causality in time series analysis is typically studied using the vector autoregressive model (VAR) [2]. In this model, the time series at time t, x_t, is assumed to be a linear combination of the past K lags of the series

$$x_t = \sum_{k=1}^{K} A^{(k)} x_{t-k} + e_t, \tag{1}$$

where $A^{(k)}$ is a $p \times p$ matrix that specifies how lag k affects the future evolution of the series and e_t is mean zero noise. In this model, time series j does not Granger cause time series i iff $\forall k, A_{ij}^{(k)} = 0$.

A Granger causal analysis in a VAR model thus reduces to determining which values in $A^{(k)}$ are zero over all lags. In higher dimensional settings, this may be determined by solving a group lasso regression problem [21]

$$\min_{A^{(1)},\ldots,A^{(K)}} \sum_{t=K}^{T} \left(x_t - \sum_{k=1}^{K} A^{(k)} x_{t-k} \right)^2 + \lambda \sum_{ij} \|(A_{ij}^{(1)},\ldots,A_{ij}^{(K)})\|_2,$$

where $\|.\|_2$ denotes the the L_2 norm which acts as a group penalty jointly shrinking all values of $(A_{ij}^{(1)},\ldots,A_{ij}^{(K)})$ to zero [10] and $\lambda > 0$ is a hyperparameter that controls the level of group sparsity. More recently, [11] proposed an optimization scheme that replaces the group penalty with a structured hierarchical penalty [22, 23] which for a fixed λ automatically selects the lag of each interaction.

2.1 Nonlinear Autoregressive Models and Granger Causality

A *nonlinear* autoregressive model allows x_t to evolve according to more general nonlinear dynamics

$$x_t = g(x_{<t1},\ldots,x_{<tp}) + e_t, \tag{2}$$

where $x_{<ti} = (\ldots,x_{(t-2)i},x_{(t-1)i})$ denotes the past of series i. The nonlinear autoregressive function g may be written componentwise:

$$x_{ti} = g_i(x_{<t1},\ldots,x_{<tp}) + e_{ti}$$

where g_i is a function that specifies how the past K lags influence series i. In this context, Granger non-causality between two series j and i means that the function g_i does not depend on $x_{<tj}$, the past lags of series j. More formally,

Definition 1 *Time series j is* Granger non-causal *for time series i if for all $(x_{<t1},\ldots,x_{<tp})$ and all $x'_{<tj} \neq x_{<tj}$,*

$$g_i(x_{<t1},\ldots,x_{<tj},\ldots,x_{<tp}) = g_i(x_{<t1},\ldots,x'_{<tj},\ldots x_{<tp});$$

that is, g_i is invariant to $x_{<tj}$.

Our goal is to flexibly estimate nonlinear Granger causal and non-causal relationships using a penalized optimization approach similar to (2).

3 Sparse Input MLPs for Time Series

Our first approach models the nonlinear dynamics with a multilayer perceptron (MLP). In a forecasting setting, it is common to model the full set of outputs x_t using an MLP where the inputs are $x_{<t} = x_{(t-1):(t-K)}$, for some lag K. There are two problems with applying this approach to inferring Granger causality. First, due to sharing of hidden layers, it is difficult to specify sufficient conditions on the weights that simultaneously allows series j to influence series i but not influence series i' for $i \neq i'$. Sufficient conditions for Granger causality are needed because we wish to add selection penalties during estimation time. Second, a joint MLP requires all g_i functions to depend on the same lags. However, in practice each g_i may have different lag order dependencies.

To tackle these challenges we model each component g_i with a separate MLP, so that we can easily disentangle the effects from inputs to outputs. We refer to this approach as a *componentwise* MLP (cMLP). Assume that for each i, g_i takes the form of an MLP with L layers and let the vector h_t^ℓ denote the values of the ℓth hidden layer at time t. Let $\mathbf{W} = \{W^1,\ldots,W^L\}$ denote the weights
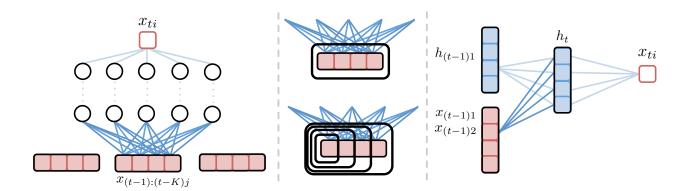
Figure 1: (left) Schematic for modeling Granger causality using cMLPs. If the outgoing weights for series j, shown in dark blue, are penalized to zero, then series j does not influence series i. (middle) The group lasso penalty jointly penalizes the full set of outgoing weights while the hierarchical version penalizes the nested set of outgoing weights, penalizing higher lags more. (right) Schematic for modelling Granger causality using a cLSTM. If the dark blue outgoing weights to the hidden units from an input $x_{(t-1)j}$ are zero, then series j does not influence series i.

at each layer and let the first layer weights be written as $W^1 = \{W^{11}, \ldots, W^{1K}\}$. The first layer hidden values at time t are given by

$$h_t^1 = \sigma \left(\sum_{k=1}^{K} W^{1k} x_{t-k} + b^1 \right), \tag{3}$$

where σ is an activation function and b^1 is the bias at layer 1. The subsequent layers are given by fully connected units with σ activation functions. The output x_{ti} is given by

$$x_{ti} = g_i(x_{<t}) + e_{ti} = w_O^T h_t^L + e_{ti} \tag{4}$$

where w_O^T is the linear output decoder and h_t^L is the final hidden output from the final Lth layer.

In Eq. (3), if the jth column of the first layer weight matrix, $W_{:j}^{1k}$ contains zeros for all k, then time series j does not Granger cause series i. Thus, analogously to the VAR case, one may select for Granger causality by applying a group lasso penalty to the columns of the W^{1k} matrices for each g_i as in Eq. (2),

$$\min_{\mathbf{W}} \sum_{t=K}^{T} \left(x_{it} - g_i(x_{(t-1):(t-K)}) \right)^2 + \lambda \sum_{j=1}^{p} \|(W_{:j}^{11}, \ldots, W_{:j}^{1K})\|_F. \tag{5}$$

For large enough λ, the solutions to Eq. (5) will lead to many zero columns in each W^{1k} matrix, implying only a small number of estimated Granger causal connections.

The zero outgoing weights are a sufficient but not necessary condition to represent Granger non-causality. Indeed, series i could be Granger non-causal of series j through a complex configuration of the weights that exactly cancel each other. However, because we wish to *interpret* the outgoing weights of the inputs as a measure of dependence, it is important that these weights reflect the true relationship between inputs and outputs. Our penalization scheme acts as a prior that biases the network to represent Granger non-causal relationships with zeros in the outgoing weights of the inputs, rather than through other configurations. Our simulation results in Section 5 validate this intuition.

3.1 Simultaneous Granger Causality and Lag Selection

We may simultaneously select for both Granger causality and the lag order of the interaction by replacing the group lasso penalty in Eq. (5) with a *hierarchical* group lasso penalty [11] in the MLP optimization problem,

$$\min_{\mathbf{W}} \sum_{i=1}^{T} \left(x_{it} - g_i(x_{(t-1):(t-K)}) \right)^2 + \lambda \sum_{j=1}^{p} \sum_{k=1}^{K} \|(W_{:j}^{1k}, \ldots, W_{:j}^{1K})\|_F. \tag{6}$$

The hierarchical penalty leads to solutions such that for each j there exists a lag k such that all $W_{:j}^{1k'} = 0$ for $k' > k$ and all $W_{:j}^{1k'} \neq 0$ for $k' \leq k$. Thus, this penalty effectively selects the lag of each interaction. The hierarchical penalty also sets many columns of W^{1k} to be zero across all k, effectively selecting for Granger causality. In practice, the hierarchical penalty allows us to fix K to a large value, ensuring that no Granger causal connections at higher lags are missed.

Optimizing the Penalized cMLP Objective   We optimize the nonconvex objectives of Eq. (5) and (6) using proximal gradient descent with line search. Line search is preferred because it aids in convergence to a local optimum. Proximal optimization is important in our context because it leads to *exact zeros* in the columns of the input matrices, an important requirement for interpreting Granger non-causality in our framework. Since all datasets we study are relatively small, our gradients are with respect to the full data objective; for larger datasets one could use proximal stochastic gradient. The proximal step for the group lasso penalty is given by a group soft-thresholding operation on the input weights [24]. The proximal step for the hierarchical penalty is given by iteratively applying the group soft-thresholding operation on each nested group in the penalty, from the smallest group to the largest group [22].

4   Sparse Input Recurrent Neural Networks

Recurrent neural networks (RNNs) are particularly well suited to modeling time series, as they compress the past of a time series into a hidden state, allowing them to capture complicated nonlinear dependencies at longer time lags than traditional time series models. As with MLPs, time series forecasting with RNNs typically proceeds by jointly modeling the entire evolution of the multivariate series using a single recurrent network. However, as in the MLP case, it is difficult to disentangle how each series affects the evolution of another series. This problem is even more severe in complicated recurrent networks like LSTMs.

To model Granger causality with RNNs, we follow the same strategy as with MLPs and model each g_i function using a separate RNN. For simplicity, we assume a one-layer recurrent network, but our formulation may be easily generalized to accommodate more layers.

4.1   Componentwise Recurrent Networks

Let $h_{t-1} \in \mathbb{R}^m$ represent the m-dimensional hidden state at time t, that represents the historical context of the time series for predicting a component x_{ti}. The hidden state at time $t+1$ is updated recursively

$$h_t = f(x_t, h_{t-1}), \tag{7}$$

where f is some nonlinear function that depends on the particular recurrent architecture. For simplicity, we model the output $g_i(x_{<t})$ as a linear function of the hidden states at time t:

$$x_{it} = g_i(x_{<t}) + e_{ti} = w_O^T h_t + e_{ti}, \tag{8}$$

where the dependence of g_i on the full past sequence $x_{<t}$ is due to recursive updates of the hidden state h_t.

Due to their effectiveness at modeling complex time dependencies, we choose to model the recurrent function f using an LSTM. The LSTM model introduces a second hidden state variable c_t, referred to as the cell state, giving the full set of hidden parameter as (c_t, h_t). The standard LSTM model takes the form

$$
\begin{aligned}
&f_t = \sigma\left(W^f x_t + U^f h_{(t-1)}\right), \ \ i_t = \sigma\left(W^{in} x_t + U^{in} h_{(t-1)}\right) \\
&o_t = \sigma\left(W^o x_t + U^o h_{(t-1)i}\right) \\
&c_t = f_t \odot c_{t-1} + i_t \odot \sigma\left(W^c x_t + U^c h_{(t-1)}\right) \\
&h_t = o_t \odot \sigma(c_t)
\end{aligned} \tag{9}
$$

where $\odot$ denotes componentwise multiplication and i_t, f_t, and o_t represent input, forget and output gates, respectively, that control how each component of the state cell, c_t, is updated and then transferred to the hidden state used for prediction, h_t.

4.2 Granger Causality Selection in LSTMs

In Eq. (9) the set of input matrices,

$$W = \left((W^f)^T, (W^{in})^T, (W^o)^T, (W^c)^T\right)^T, \tag{10}$$

controls how the past time series, x_t, influences the forget gates, input gates, output gates, and cell updates, and, consequently, the update of the hidden representation. Like in the MLP case, for this componentwise LSTM model (cLSTM) a sufficient condition for Granger non-causality of an input series j on an output i is that all elements of the jth column of W are zero, $W_{:j} = 0$. Thus we may select for which series Granger cause series i during estimation using a group lasso penalty across columns of W

$$\min_{W,U,w^O} \sum_{t=2}^{T} (x_{it} - g_i(x_{<t}))^2 + \lambda \sum_{j=1}^{p} \|W_{:j}\|_2, \tag{11}$$

where $U = \left((U^f)^T, (U^{in})^T, (U^o)^T, (U^c)^T\right)^T$. For a large enough λ, many columns of W will be zero, leading to a sparse set of Granger causal connections.

Optimizing the Penalized cLSTM Objective Similar to the cMLP, we optimize Eq. (11) using proximal gradient descent with line search. When the data consists of many replicates of short time series, like in the DREAM data in Section 6, we perform a full backpropagation through time to compute the gradients. However, for longer series we truncate the backpropagation through time by unlinking the hidden sequences. In practice, we do this by splitting the data set up into equal sized batches, and treating each batch as an independent realization. Under this approach, the gradients used to optimize Eq. (11) are only approximations of the gradients of the full component-wise LSTM model[1].

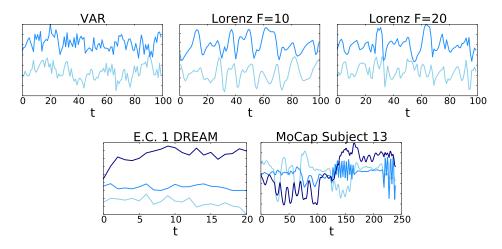[1]Code for all models will be released upon publication.

Figure 2: Example multivariate linear (VAR) and nonlinear (Lorenz, DREAM, and MoCap) series that we analyze using both cMLP and cLSTM models. Note as the forcing constant, F, in the Lorenz model increases, the data become more chaotic.

4.3 Comparing cMLP and cLSTM Models for Granger Causality

Both cMLP and cLSTM frameworks model each component function g_i using independent networks for each i. For the cMLP model, one needs to specify a maximum possible model lag K. However, our lag selection strategy (Eq. 6) allows one to set that to a large value and the weights for higher lags are automatically removed from the model. On the other hand, the cLSTM model requires no maximum lag specification, and instead automatically learns the memory of each interaction. As a consequence, the cMLP and cLSTM differ in the amount of data used for training, as noted by a comparison of the t index in Eq. (11) and (6). For a length T series, the cMLP and cLSTM models use $T - K$ and $T - 1$ data points, respectively. While insignificant for large T, when the data consists of independent replicates of short series, as in the DREAM3 data in Sec. 6, the difference may be important. This ability to simultaneously model longer range depedencies while harnessing the full training set may explain the impressive performance of the cLSTM on the DREAM3 data in Section 6.

5 Simulation Experiments

5.1 cMLP and cLSTM Simulation Comparison

To compare and analyze the performance of our two approaches, cMLP and cLSTM, we apply both methods to detecting Granger causality networks in simulated VAR data and simulated Lorenz-96 data [12], a nonlinear model of climate dynamics. Overall, our results show that our methods can accurately reconstruct the underlying Granger causality graph in both linear and nonlinear settings. We describe the results from the Lorenz experiment in detail and present the similar VAR results in the Supplementary Material.

The continuous dynamics in a p-dimensional Lorenz model are

$$\frac{dx_{ti}}{dt} = (x_{i+1} - x_{i-2}) x_{i-1} - x_i + F, \tag{12}$$

where $x_{-1} = x_{p-1}$, $x_0 = x_p$, and $x_{p+1} = x_1$ and F is a forcing constant which determines the level of nonlinearity and chaos in the series. Example series for two settings of F are displayed in Figure

7

F	10	10	40	40
T	500	1000	500	1000
cMLP	**95.7**	**99.1**	86.5	92.0
cLSTM	76.1	90.0	**87.5**	**96.25**

Table 1: Mean AUROC comparisons between cMLP and cLSTM Granger causality selection across five simulated Lorenz datasets, as a function of the forcing constant F, and the length of the time series T.

2. We numerically simulate the Lorenz-96 model with a sampling rate of $\Delta_t = 0.05$, which results in a multivariate, nonlinear time series with sparse Granger causal connections.

Average area under the ROC curve (AUROC) results across five simulation seeds are shown in Table 1 for both the cMLP and cLSTM models for two different data set lengths, $T \in (500, 1000)$, and forcing constants, $F \in (10, 40)$. We use $m = 10$ hidden units for both methods. While more layers may prove beneficial, for all experiments we fix the number of hidden layers, L, to one and leave the effects of additional hidden layers to future work. For the cMLP, we use the hierarchical penalty with model lag of $K = 5$; see Section 5.2 for a performance comparison of several possible penalties across model input lags.

The AUROC curves are computed by sweeping λ across a range of values; discarded edges (inferred Granger non-causality) for a particular λ setting are those whose associated L_2 norm of the input weights of the neural network is exactly zero. Note that our proximal gradient algorithm sets many of these groups to be exactly zero.

First, as expected, the results indicate that the cMLP and cLSTM performance improves as the data set size T increases. Furthermore, the cMLP outperforms the cLSTM in the less chaotic regime of $F = 10$, but underperforms in the more chaotic regime when $F = 40$.

5.2   Quantitative Analysis of the Hierarchical Penalty

We next quantitatively compare three possible structured penalties for Granger causality selection in the cMLP model. In Section 3 we introduced the full group lasso (GROUP) penalty over all lags (Eq. (5)), and the hierarchical (HIER.) lag selection penalty (Eq. (6)). An additional mixed group sparse group lasso (MIXED), a generalization of the sparse group lasso [25] to our case, is considered as well. More information on this penalty is availible in the Supplementary Material. We compare these approaches across various choices of the cMLP's model lag, $K \in (5, 10, 20)$ with $m = 10$ hidden units, for data simulated from the nonlinear Lorenz model with $F = 20$, $p = 20$, and $T = 750$. As in Section 5.1, we compute the mean AUROC over five simulation runs and display the results in Table 2. Importantly, the hierarchical penalty outperforms both group and mixed penalties across all model input lags K. Furthermore, performance significantly declines as K increases in both group and mixed settings while the performance of the hierarchical penalty stays roughly *constant* as K increases. This result suggests that performance of the hierarchical penalty for nonlinear Granger causality selection is robust to the input lag, implying that precise lag specification is unnecessary. In practice, this allows one to set the model lag to a high value without worrying that nonlinear Granger causality detection will be compromised.
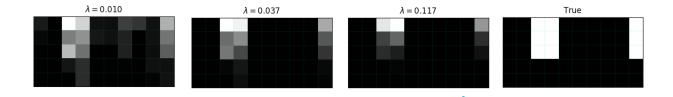
Figure 3: Qualitative results of the cMLP automatic lag selection using a hierarchical group lasso penalty and maximal lag of $K = 5$. The true data are from a VAR(3) model. The images display results for a single cMLP (one output series) using various penalty strengths λ. The columns of each image correspond to different input series while the rows correspond to the lag, with $k = 1$ at the top and $k = 5$ at the bottom. The magnitude of each entry is the L_2 norm of the associated input weights of the neural network after training. The true lag interactions are shown in the rightmost image.

5.3 Qualitative Analysis of the Hierarchical Penalty

To qualitatively validate the performance of the hierarchical group lasso penalty for automatic lag selection, we apply our penalized cMLP framework to data generated from a sparse VAR model with longer interactions. Specifically, we generate data from a $p = 10$, VAR(3) model as in Eq. (1). To generate sparse dependencies for each time series i, we create self dependencies and randomly select two more dependencies among the other $p - 1$ time series. Where series i depends on series j, we set $A_{ij}^k = .096$ for $k = 1, 2, 3$. All other entries of A are set to zero. This implies that the Granger causal connections that do exist are all of true lag 3. We run the cMLP with the hierarchical group lasso penalty and a maximal lag order of $K = 5$.

We visually display the selection results for one cMLP (i.e., one output series) across a variety of λ settings in Figure 3. For the lower $\lambda = .01$ setting, the cMLP both (i) overestimates the lag order for a few input series and (ii) allows some false positive Granger causal connections. For the higher $\lambda = .037$, lag selection performs almost perfectly, in addition to correct estimation of the Granger causality graph. Higher λ values lead to larger penalization on longer lags, resulting in weaker long-lag connections.

K	5	10	20
GROUP	89.2	85.6	84.2
MIXED	90.2	88.5	87.4
HIER.	**94.5**	**93.5**	**94.3**

Table 2: AUROC comparisons between different cMLP Granger causality selection penalties on simulated Lorenz data as a function of the input model lag, K.

6 DREAM Challenge

We next apply our methodology to determine Granger causality networks in a realistically simulated time course gene expression data set. The data are from the DREAM3 challenge [15] and provide a difficult, nonlinear data set for rigorously comparing methods for Granger causality detection [13, 14]. The data is simulated using continuous gene expression and regulation dynamics, with multiple hidden factors that are not observed. The challenge contains five different simulated data sets, each with different ground truth Granger causality graphs: two E. Coli (E.C.) data sets and three Yeast

(Y.) data sets. Each data set contains $p = 100$ different time series, each with 46 replicates sampled at 21 time points for a total of 966 time points. This represents a very limited data scenario relative to the dimensionality of the networks and complexity of the underlying dynamics of interaction. Three time series components from a single replicate of the E. Coli 1 data set are shown in Figure 2.

We apply both the cMLP and cLSTM to all five data sets. Due to the short length of the series replicates, we choose the maximum lag in the cMLP to be $K = 2$ and use 5 and 10 hidden units for the cMLP and LSTM, respectively. For our performance metric, we consider the DREAM3 challenge metrics of area under the ROC curve (AUROC) and area under the precision recall curve (AUPR). Both curves are computed by sweeping λ over a range of values, as described in Section 5.

In Figure 4, we compare the AUROC and AUPR of our cMLP and cLSTM to previously published AUROC and AUPR results on the DREAM3 data [13]. These comparisons include both linear and nonlinear approaches: (i) a linear VAR model with a lasso penalty (LASSO) [3], (ii) a dynamic Bayesian network using first-order conditional dependencies (G1DBN) [14], and (iii) a state-of-the-art multi-output kernel regression method (OKVAR) [13]. The latter is the most mature of a sequence of nonlinear kernel Granger causality detection methods [8, 26]. In terms of AUROC, our cLSTM outperforms all methods across all five datasets. Furthermore, the cMLP method outperforms previous methods on two datasets, Y.1 and Y.3, ties G1DBN on Y.2, and slightly under performs OKVAR in E.C.1 and E.C.2. In terms of AUPR, both cLSTM and cMLP methods do much better than all previous approaches, with the cLSTM outperforming the cMLP in three datasets. The PR and ROC curves are displayed in the Supplementary Material.
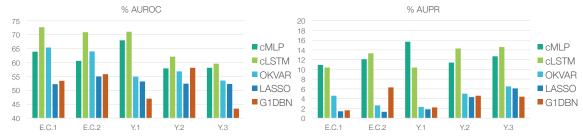

Figure 4: (Top) AUROC and (bottom) AUPR (given in %) results for our proposed regularized cMLP and cLSTM models and the set of methods—OKVAR, LASSO, and G1DBN—presented in [13]. These results are for the DREAM3 size-100 networks using the original DREAM3 data sets.

These results clearly demonstrate the importance of taking a nonlinear approach to Granger causality detection in a (simulated) real-world scenario. Among the nonlinear approaches, the neural network methods are extremely powerful. Furthermore, the cLSTM's ability to efficiently capture long memory (without relying on long-lag specifications) appears to be particularly useful. This result validates many findings in the literature where LSTMs outperform MLPs. An interesting facet of these results, however, is that the impressive performance gains are achieved in a limited data scenario and on a task where the goal is recovery of interpretable structure. This is in contrast to the standard story of prediction on large datasets. For this, the regularization and induced sparsity of our penalties is critical.

7   Dependencies in Human Motion Capture Data

We next apply our methodology to detect complex, nonlinear dependencies in human motion capture (MoCap) recordings. In contrast to the DREAM3 challenge results, this analysis allows us to more easily visualize and interpret the learned network. Human motion has been previously modeled

(a) $\lambda = .06$ (b) $\lambda = .18$ (c) $\lambda = .36$
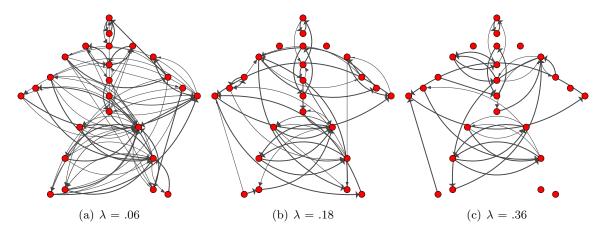
Figure 5: Nonlinear Granger causality graphs inferred from the human MoCap data set using the regularized cLSTM model. Results are displayed for a range of λ values. Each node corresponds to one location on the body.

using both linear dynamical systems [27], switching linear dynamical systems [28, 17] and also nonlinear dynamical models using Gaussian processes [29]. While the focus of previous work has been on motion classification [27] and segmentation [17], our analysis delves into the potentially long-range, nonlinear dependencies between different regions of the body during natural motion behavior. We consider a data set from the CMU MoCap database [16] previously studied in [17]. The data set consists of $p = 54$ joint angle and body position recordings across two different subjects for a total of $T = 2024$ time points. In total there are recordings from 24 unique regions because some regions, like the thorax, contain multiple angles of motion corresponding to the degrees of freedom of that part of the body.

We apply the cLSTM model with $m = 8$ hidden units to this data set. For computational speed ups, we break the original series into length 20 segments and fit the regularized cLSTM model from Eq. (11) over a range of λ values. To develop a weighted graph for visualization, we let the edge weight e_{ij} between components be the norm of the outgoing cLSTM weights from input series j to output component series i, standardized by the maximum such edge weight associated with the cLSTM for series i. Finally, edges associated with more than one degrees of freedom (angle directions) for the same body part are averaged together.

The resulting estimated graphs are displayed in Figure 5 for multiple values of the regularization parameter, λ. To interpret the presented skeleton plots, it is useful to understand the full set of motion behaviors exhibited in this data set. These behaviors are depicted in the Supplementary Material, and include instances of *jumping jacks*, *side twists*, *arm circles*, *knee raises*, *squats*, *punching*, various forms of *toe touches*, and *running in place*. Due to the extremely limited data for any individual behavior, we chose to learn interactions from data aggregated over the entire collection of behaviors. In Figure 5, we see many intuitive learned interactions. For example, even in the more sparse graph (largest λ) we learn a directed edge from right knee to left knee and a separate edge from left knee to right. This makes sense as most human motion, including the motions in this dataset involving lower body movement, entail the right knee leading the left and then vice versa. We also see directed interactions leading down each arm, and between the hands and toes for toe touches.

8   Discussion

We have presented a framework for nonlinear Granger causality selection using regularized neural network models of time series. To easily disentangle the effects of the past of an input series on the future of an output, we model each output series using a separate neural network. We then apply both cMLP and cLSTM architectures, with associated sparsity promoting penalties on incoming weights to the network, and select for Granger causality. Overall, our results show that these methods outperform existing Granger causality approaches on the challenging DREAM3 data set and furthermore discover interpretable and insightful structure on a highly nonlinear MoCap data set.

Our work opens the door to multiple exciting avenues for future work. First, while we are the first to use a hierarchical lasso penalty in a neural network, it would be interesting to also explore other types of structured penalties, such as tree structured penalties [30].

Furthermore, while we have presented two relatively simple approaches, based off single layer MLPs and LSTMs, our general framework of penalized input weights easily accommodates more powerful architectures. Exploring the effects of multiple hidden layers, powerful recurrent and convolutional architectures, like clockwork RNNs [31] and dilated causal convolutions [32], opens up a wide swath of research directions and has the potential to detect very long range and complex dependencies.

Finally, while we consider sparse input models, a different *sparse output* architecture would use a network, like an RNN, to learn hidden representations of each individual input series, and then model each output component as a sparse combination across the hidden states of all time series, allowing a shared hidden representation across component tasks.

References

[1] Sumanta Basu, Ali Shojaie, and George Michailidis. Network Granger causality with inherent grouping structure. *The Journal of Machine Learning Research*, 2015.

[2] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[3] Aurelie C Lozano, Naoki Abe, Yan Liu, and Saharon Rosset. Grouped graphical Granger modeling methods for temporal causal modeling. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[4] Timo Terasvirta, Dag Tjostheim, Clive WJ Granger, et al. Modelling nonlinear economic time series. *OUP Catalogue*, 2010.

[5] Howell Tong. Nonlinear time series analysis. In *International Encyclopedia of Statistical Science*. Springer, 2011.

[6] Bethany Lusch, Pedro D. Maia, and J. Nathan Kutz. Inferring connectivity in networked dynamical systems: Challenges using Granger causality. *Phys. Rev. E*, 2016.

[7] Trevor Hastie and Robert Tibshirani. *Generalized additive models*. Wiley Online Library, 1990.

[8] Vikas Sindhwani, Ha Quang Minh, and Aurélie C. Lozano. Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and Granger causality. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2013.

[9] Alex Graves. Supervised sequence labelling. In *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.

[10] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006.

[11] W. B. Nicholson, J. Bien, and D. S. Matteson. Hierarchical Vector Autoregression. *ArXiv e-prints*, 2014.

[12] A Karimi and Mark R Paul. Extensive chaos in the Lorenz-96 model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2010.

[13] Néhémy Lim, Florence dAlché Buc, Cédric Auliac, and George Michailidis. Operator-valued kernel-based vector autoregressive models for network inference. *Machine learning*, 2015.

[14] Sophie Lèbre. Inferring dynamic genetic networks with low order independencies. *Statistical applications in genetics and molecular biology*, 2009.

[15] Robert J Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K Sorger, Leonidas G Alexopoulos, Xiaowei Xue, Neil D Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: the DREAM3 challenges. *PloS one*, 2010.

[16] CMU. Carnegie Mellon University graphics lab motion capture database. Available at http://mocap.cs.cmu.edu/. 2009.

[17] Emily B Fox, Michael C Hughes, Erik B Sudderth, and Michael I Jordan. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *The Annals of Applied Statistics*, 2014.

[18] Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, 2016.

[19] C. Louizos, K. Ullrich, and M. Welling. Bayesian Compression for Deep Learning. *ArXiv e-prints*, 2017.

[20] J. Feng and N. Simon. Sparse-Input Neural Networks for High-dimensional Nonparametric Regression and Classification. *ArXiv e-prints*, 2017.

[21] Aurélie C Lozano, Naoki Abe, Yan Liu, and Saharon Rosset. Grouped graphical Granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 2009.

[22] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 2011.

[23] Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *Journal of Machine Learning Research*, 2011.

[24] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 2014.

[25] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 2013.

[26] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. Kernel-Granger causality and the analysis of dynamical networks. *Physical review E*, 2008.

[27] Eugene Hsu, Kari Pulli, and Jovan Popović. Style translation for human motion. In *ACM Transactions on Graphics (TOG)*. ACM, 2005.

[28] Vladimir Pavlovic, James M Rehg, and John MacCormick. Learning switching linear models of human motion. In *Advances in neural information processing systems*, 2001.

[29] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 2008.

[30] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. 2010.

[31] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. In *International Conference on Machine Learning*, 2014.

[32] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
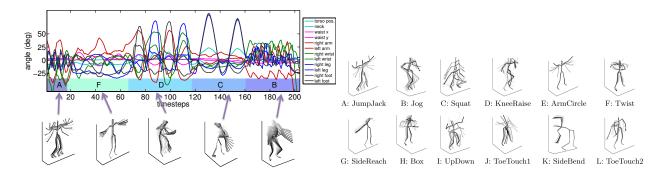
Figure 6: (top) Example time series from the MoCap data set paired with their particular motion behaviors. (bottom) Skeleton visualizations of 12 possible exercise behavior types observed across all sequences analyzed in the main text.

9 Supplementary Material

9.1 More information on the MoCap dataset

The MoCap dataset we analyze in the main paper has been extensively analyzed [17]. In particular, it contains motion behaviors from two different subjects repeatedly performing certain types of motions, like toe touches, jumping jacks, etc. A diagram of the motions obtained using an unsupervised segmentation algorithm [17], that we allude to in the main text, are shown in Figure 6.

9.2 Sparsity-encouraging penalties for cMLP

In Section 3 of the main text, we explain the rationale for encouraging group-wise sparsity among sets of parameters in the outgoing weights of the inputs. Because Granger non-causality is inferred when a group of weights is set to zero, there exists some flexibility in the choice of penalty. In general, the penalized objective function for g_i takes the form

$$\min_{\mathbf{W}} \sum_{i=1}^{T} \left(x_{it} - g_i(x_{(t-1):(t-K)}) \right)^2 + \lambda \Omega(W^1).$$

In Section 4.2 of the main text, we provide a quantitative comparison of three candidate penalties. We introduce those three penalties in this context below.

Group lasso The group lasso penalty is the simplest penalty that encourages specific groups of weights to go to zero. It imposes no additional structure, but shrinks all lags to zero as a group,

$$\Omega_{GL}(W^1) = \sum_{j=1}^{p} \|(W^{11}_{:j}, \ldots, W^{1K}_{:j})\|_F.$$

Mixed The mixed penalty is a combination of one group lasso over the entire lag history, and a separate set of group lassos applied independently on the neural network input weights for each lag. Like the sparse group lasso [25], it provides both sparsity of groups (a sparse set of Granger causal time series) and within groups (a subset of relevant lags),

$$\Omega_M(W^1) = \alpha \sum_{j=1}^{p} \|(W^{11}_{:j}, \ldots, W^{1K}_{:j})\|_F + (1-\alpha) \sum_{j=1}^{p} \sum_{k=1}^{K} \|(W^{1k}_{:j})\|_F.$$

15

T	500	1000
cMLP	**93.5**	**97.0**
cLSTM	63.6	87.5

Table 3: AUROC comparisons between cMLP and cLSTM Granger causality selection on a simulated VAR dataset, as a function of the length of the time series T.

In our experiments, we set $\alpha = 0.5$.

Hierarchical The hierarchical penalty imposes additional structure by placing a higher cost on long-lag dependencies. This results in both entire sets of parameters being set to zero, and automatic lag selection: for each j there exists a lag k such that all $W_{:j}^{1k'} = 0$ for $k' > k$ and all $W_{:j}^{1k'} \neq 0$ for $k' \leq k$,

$$\Omega_H(W^1) = \sum_{j=1}^{p} \sum_{k=1}^{K} \|(W_{:j}^{1k}, \ldots, W_{:j}^{1K})\|_F.$$

9.3   cMLP and cLSTM comparison on VAR(2)

As in Section 5.1 in the main text, we performed a small simulation study to verify the performance of the cMLP and cLSTM on data generated from a VAR(2) model with $p = 20$. Data were generated as follows. To generate sparse dependencies for each time series i, we create self dependencies and randomly select three more dependencies among the other $p - 1$ time series. Where series i depends on series j, we set $A_{ij}^k = .112$ for $k = 1, 2$. All other entries of A are set to zero. We performed this analysis for a single random seed.

The results are displayed in Table 3 for $T \in (500, 1000)$. As expected, the performance of both models improves at larger T. Interestingly, the cMLP outperforms the cLSTM in both cases. We attribute this to the difficulty of our shallow cLSTM to memorize information from previous lags. By contrast, the cMLP can directly use information from previous lags.

9.4   ROC and PR plots

In Figures 9, 10, 7, and 8 we include the ROC and PR plots for both the cLSTM and cMLP on the five DREAM3 data sets. It should be noted that several PR curves associated with the LSTM are unsmooth, reflecting occasional difficulty in reaching convergence. However, poor convergence would only hurt our AUPR numbers, implying a better optimization scheme would only improve our results.
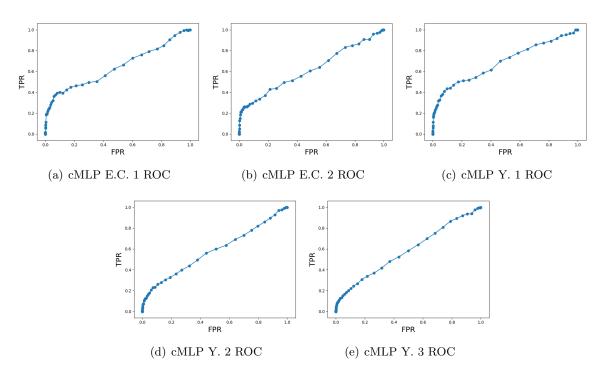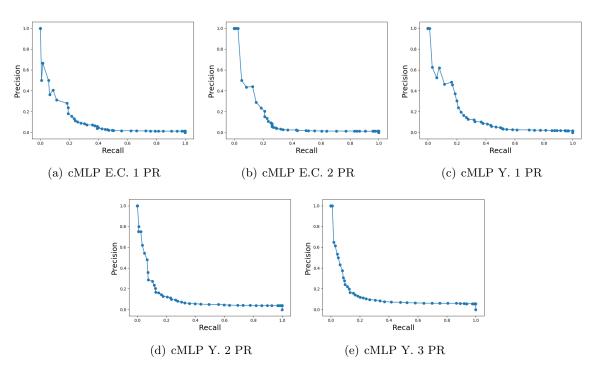
(a) cMLP E.C. 1 ROC

(b) cMLP E.C. 2 ROC

(c) cMLP Y. 1 ROC

(d) cMLP Y. 2 ROC

(e) cMLP Y. 3 ROC

Figure 7: ROC curves for the cMLP model

(a) cMLP E.C. 1 PR

(b) cMLP E.C. 2 PR

(c) cMLP Y. 1 PR

(d) cMLP Y. 2 PR

(e) cMLP Y. 3 PR

Figure 8: PR curves for the cMLP model

(a) cLSTM E.C. 1 ROC (b) cLSTM E.C. 2 ROC (c) cLSTM Y. 1 ROC

(d) cLSTM Y. 2 ROC (e) cLSTM Y. 3 ROC

Figure 9: ROC curves for the cLSTM model

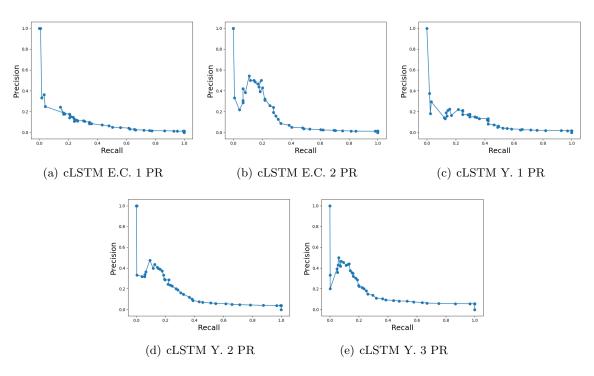(a) cLSTM E.C. 1 PR (b) cLSTM E.C. 2 PR (c) cLSTM Y. 1 PR

(d) cLSTM Y. 2 PR (e) cLSTM Y. 3 PR

Figure 10: PR curves for the cLSTM model