Scalable Causal Graph Learning through a Deep Neural Network

C. Xu, S. Yoo

Submitted to the CIKM 2019 Conference
to be held at Beijing China
November 03 - 07, 2019

Computational Science Initiative

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC), Advanced Scientific Computing Research (SC-21)

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Scalable Causal Graph Learning through a Deep Neural Network

Chenxiao Xu*
chenxiao.xu@stonybrook.edu
Stony Brook University

Hao Huang*
haohuangcssbu@gmail.com
GE Global Research

Shinjae Yoo
shinjae.yoo@stonybrook.edu
Stony Brook University

ABSTRACT

Learning the causal graph in a complex system is crucial for knowledge discovery and decision making, yet it remains a challenging problem because of the unknown nonlinear interaction among system components. Most of the existing methods either rely on predefined kernel or data distribution, or they focus simply on the causality between a single target and the remaining system. This work presents a deep neural network for scalable causal graph learning (SCGL) through low-rank approximation. The SCGL model can explore nonlinearity on both temporal and intervariable relationships without any predefined kernel or distribution assumptions. Through low-rank approximation, the noise influence is reduced, and better accuracy and high scalability are achieved. Experiments using synthetic and real-world datasets show that our SCGL algorithm outperforms existing state-of-the-art methods for causal graph learning.

CCS CONCEPTS

• **Computing methodologies → Causal reasoning and diagnostics**.

KEYWORDS

Causal Graph; Deep Neural Network; Scalability

ACM Reference Format:
Chenxiao Xu, Hao Huang, and Shinjae Yoo. 2019. Scalable Causal Graph Learning through a Deep Neural Network. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357864

1 INTRODUCTION

Causal discovery on time series data is an important analysis task used to obtain insight into the underlying mechanisms of a whole system. It helps to interpret data, formulate and test hypotheses, and build or improve the theories of modeling. Causal discovery also is crucial for the rapidly evolving field of Explainable Artificial Intelligence [13], which aims to construct interpretable and transparent algorithms that also can explain how they model the data.

$$X(3,t_3) = \boxed{X(1,t_1)^2} + \boxed{\log\big(X(2,t_1)\big)} + \boxed{X(1,t_2)^{X(3,t_1)}} + \boxed{\boxed{X(1,t_1)^3}\,\boxed{\cos\big(X(3,t_2)\big)}}$$

Figure 1: Example of data nonlinearity differentiating: the blue bounded parts are temporal nonlinearity defined on the univariate level, while intervariable nonlinearity defined on the multivariate level is surrounded in red. $X(i,t_j)$ denotes the value of the i-th variable at time t_j.

Specifically, we are interested in learning the causal graph, where each node represents a variable or component of the targeted system. The edges are directed and each describes the causality relationship between the two connected nodes. In many applications, this graph structure is unknown or partially known, and a first issue is to infer the unknown causal relationship between nodes from the data. However, it is an especially challenging task because of 1) the unknown and complex (usually nonlinear) relationship existing inside the system, 2) noise in the dataset, and 3) scalability problem stemming from the large number of nodes.

Traditional causal graph learning methods are based either on linear systems (VAR Granger analysis, Generalized Additive Models) [22] or certain pre-assumed regression models [29]. However, many interactions are nonlinear and with unknown distribution [9, 23]. Selecting the appropriate kernel or distribution model for each time series requires a deeper understanding of domain knowledge and, in many cases, is not even possible. Furthermore, the data noise, system diversity, and scales (i.e., number of nodes) all challenge their ability to derive a reliable causal graph [15, 28, 33].

Deep learning techniques have become more and more popular in industrial and scientific applications. Yet, there is scant research about how deep learning can contribute to learning causal graphs on time series data. One problem is how to learn causal graphs given complex nonlinearity in the temporal data. In this research, we differentiate data nonlinearity into two types: temporal nonlinearity on the univariate level and intervariable nonlinearity on the multivariate level. Figure 1 shows an example, where blue rounded parts are temporal nonlinearity on the univariate level, while the red rounded ones represent intervariable nonlinearity involving more than two variables. We assume that any nonlinear causality can be represented by the combination of these two types of nonlinearity.

In our research, these two types of nonlinearity are learned in a deep neural network called *Scalable Causal Graph Learning*, or ***SCGL***. Figure 2 shows the proposed model design. *SCGL* can discover nonlinear causality between any pair of nodes (input variables) without any knowledge of generalization rules (e.g., predefined kernel or distribution assumption). We approximate the causal graph through low-rank decomposition. Through such low-rank

*Both authors contributed equally to this research.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357864

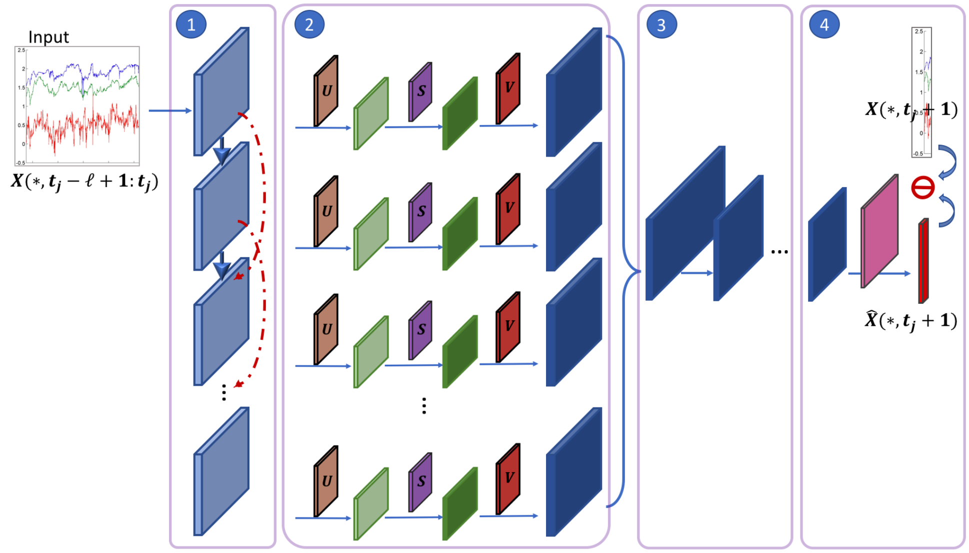


Figure 2: Our model learns a causal graph through a regression setting. It consists of four modules. Module 1 learns the temporal nonlinearity. Module 2 discovers the underlying causal graph through low-rank approximation, while Module 3 prospects intervariable nonlinearity. Module 4 is the prediction module.

learning, our *SCGL* model is less sensitive to data noise and has higher scalability in time and space.

Contributions of this work include:

(1) Our *SCGL* model not only learns the causality associated to a single targeted variable, but the whole causal graph with relationships between each pair of variables.
(2) By differentiating nonlinearity into two types, our *SCGL* model can discover complex causality without any predefined kernel or distribution assumption.
(3) We propose a multi-chain framework to explore the relevance of different-level nonlinearity to the underlying causal graph. Relevance is designed to be learnable and adaptive to different datasets.
(4) The underlying causal graph is learned in a low-rank setting, which provides model robustness against data noise and is scalable in time and space.

2 PRELIMINARIES

This section mathematically describes the problem setting, beginning with formally defining our notation.

2.1 Problem Definition

Our *SCGL* model input is the historical data of m time series with time length n, i.e., $X = \{X(1,*), X(2,*), ..., X(m,*)\} \in \mathbb{R}^{m\times n}$, where each $X(i,*) \in \mathbb{R}^{1\times n}$ is a single time series associated to the i-th variable. We denote $X(i, t_j)$ as the value of the i-th time series at time t_j and $X(i, t_j - \ell + 1 : t_j) \in \mathbb{R}^{1\times\ell}$ as the values of the i-th time series from $t_j - \ell + 1$ to t_j. $X(*, t_j) = \{X(1, t_j), X(2, t_j), ..., X(m, t_j)\} \in \mathbb{R}^{m\times 1}$ are the values of all time series at time t_j, and $X(*, t_j - \ell + 1 : t_j) = \{X(1, t_j - \ell + 1 : t_j), X(2, t_j - \ell + 1 : t_j), ..., X(m, t_j - \ell + 1 : t_j)\} \in \mathbb{R}^{m\times\ell}$ are the values of all time series from time $t_j - \ell + 1$ to t_j.

Our goal is to learn the underlying causal graph $A \in \mathbb{R}^{m\times m}$, which is a directed and non-negative matrix. When the value of $A(i, j)$ is much larger than zero, it denotes that the i-th time series is among the factors (causes) of the j-th time series in time. We use $A(*, j) \in \mathbb{R}^{m\times 1}$ to represent the j-th column of A, which tells the influencing factors of the j-th time series.

2.2 Multivariate Granger Causality

Multivariate Granger causality analysis usually is performed by fitting a vector autoregressive model (VAR) to the time series input [22]. In particular, it is formulated as the following regression problem, which tries to predict $X(*, t_j + 1)$ with $X(*, t_j - \ell + 1 : t_j)$:

$$X(*, t_j + 1) = \sum_{r=1}^{\ell} A_r X(*, t_j + 1 - r) + \epsilon(*, t_j + 1), \qquad (1)$$

where $\epsilon(*, t_j + 1) \in \mathbb{R}^{m\times 1}$ is a white Gaussian random vector at time $t_j + 1$, ℓ is the time order (involving time lags), and A_r is the causal graph for each time lag r. Time series $X(i,*)$ is called a Granger cause of time series $X(j,*)$ if at least one of the elements $A_r(i, j)$ for $r = 1, 2, ..., \ell$ is significantly larger than zero (in absolute value).

However, Equation (1) is purely linear and does not capture any nonlinear causal relationships [11]. In our research, we want to discover a more general and complex relationship by exploring the capability of deep neural networks, which do not involve any kernel assumption as found in the linear autoregression model for Equation (1). Furthermore, this research involving our *SCGL* model focuses solely on learning one global causal graph A.

3 THE PROPOSED SCGL MODEL

We deconstruct data nonlinearity into two types: temporal and intervariable nonlinearity. Temporal nonlinearity is defined on the univariate level, which describes the nonlinearity between time

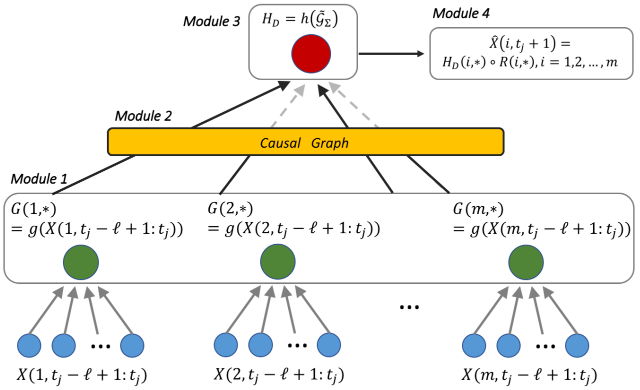

Module 3
$H_D = h(\tilde{\mathcal{G}}_\Sigma)$
Module 4
$\hat{X}(i, t_j + 1) =$
$H_D(i,*) \circ R(i,*), i = 1,2, \dots, m$
Module 2
Causal Graph
Module 1
$G(1,*)$
$= g(X(1, t_j - \ell + 1 : t_j))$
$G(2,*)$
$= g(X(2, t_j - \ell + 1 : t_j))$
$G(m,*)$
$= g(X(m, t_j - \ell + 1 : t_j))$
$X(1, t_j - \ell + 1 : t_j)$
$X(2, t_j - \ell + 1 : t_j)$
$X(m, t_j - \ell + 1 : t_j)$

Figure 3: Relationship of the four modules in Figure 2. The bottom is raw input. Function $g(*)$ in Module 1 provides the set of univariate nonlinearity among different time lags in each time series. Module 2 learns the causal graph and selects those relevant univariate nonlinearity $\tilde{\mathcal{G}}_\Sigma$. Function $h(*)$ in Module 3 learns multivariate nonlinearity H_D. Finally, Module 4 approximates the regression target.

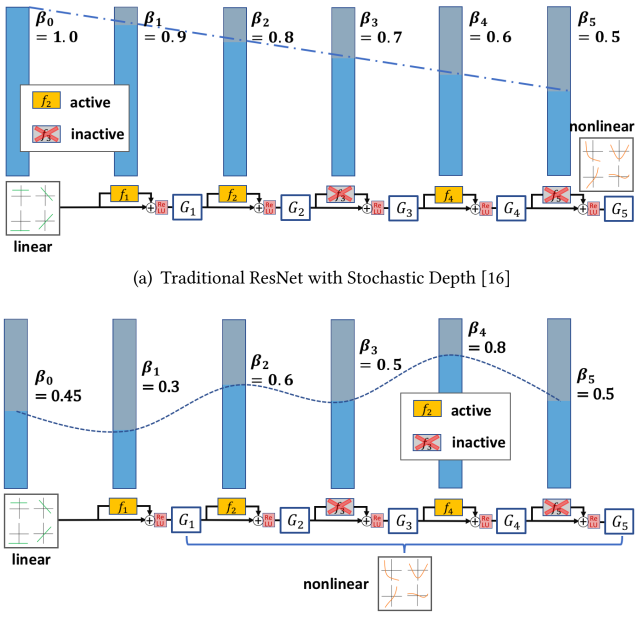
lags in each time series. Intervariable nonlinearity is defined on multivariate level, which describes the nonlinearity between time series.

By differentiating and integrating these two types of nonlinearity in the *SCGL* model, we approach the causal graph A through a regression setting: to predict $X(*, t_j + 1)$ with $X(*, t_j - \ell + 1 : t_j)$. Figure 3 shows how to integrate the two types of nonlinearity along with the causal graph learning and regression module. The bottom blue layer denotes the raw input from each time series. The $g(*)$ function in Module 1 is a nonlinear function that represents all of the useful temporal nonlinearity on each time series (univariate level). In other words, $G(i, *)$ learns the nonlinear transformation involving different time lags in $X(i, *)$ that contribute to the final regression. In Module 2, the nonlinear variables relevant to regressing $X(i, *)$ are selected by a learned causal graph. Module 3 rolls all of the selected time series $\tilde{\mathcal{G}}_\Sigma$ in a nonlinear way, which amounts to intervariable nonlinearity set H_D between time series. The final nonlinearity set is used in Module 4 to approximate the regression target.

Figure 2 shows the design of our deep neural network. It consists of four modules with the functionality shown in Figure 3. The following subsections describe them individually. Section 3.1 focuses on Module 1 that explores the temporal nonlinearity. Section 3.2 introduces Module 2 that aims to discover the causal graph, while Section 3.3 explains Module 3 that prospects intervariable nonlinearity between time series. Section 3.4 details Module 4 and the model's loss function.

3.1 Module 1: Learning Temporal Nonlinearity

In the first module, we apply a residual neural network (ResNet) [14] with learnable relevance on each layer as shown in Figure 4(b). Through this design, we will:



(a) Traditional ResNet with Stochastic Depth [16]

(b) Our ResNet with Learnable Relevance

Figure 4: Figure 4(b) shows the Module 1 design. Different from traditional ResNet with stochastic depth (Figure 4(a)), we design a ResNet with Learnable Layer Relevance to the underlying causal graph (Figure 4(b)). At the end, all the layer outputs (not just the last layer) are sent to Module 2.

(1) Learn the temporal nonlinearity on univariate level with deep yet easily trainable architecture while avoiding vanishing gradient.
(2) Discover the relevance of different levels of temporal nonlinearity to the underlying causal graph.

By setting $G_0 = X(*, t_j - \ell + 1 : t_j)$, a typical ResNet proposed in [14] is defined as:

$$G_q = ReLU(G_{q-1}B_q + id(G_{q-1})), \quad q = 1, 2, \dots, Q \tag{2}$$

where $G_q \in \mathbb{R}^{m \times p}$ denotes the output of the q-th residual block, $B_q \in \mathbb{R}^{p \times p}$ notes the weight matrix in the q-th block[1] , *id* is an identity mapping, and Q is the total number of residual blocks. The core idea of ResNet is to introduce an identity shortcut connection that skips one or more layers if they have no contribution to the final target. Therefore, the *SCGL* model goes deeper as it will not produce a training error greater than its shallower counterparts. In other words, if identity mappings are optimal, the solvers may simply drive the weights in B_q toward zero to approach identity mappings [14]. To accelerate this learning speed, Huang et al. [16] proposed ResNet with stochastic depth:

$$G_q = ReLU(b_q(G_{q-1}B_q) + id(G_{q-1})), \quad q = 1, 2, \dots, Q \tag{3}$$

where b_q is a Bernoulli random variable that can be only 1 or 0 (indicating if the q-th block is active). The block becomes a normal residual block when $b_q = 1$ and an identity layer otherwise. The

[1] The first weight $B_1 \in \mathbb{R}^{\ell \times p}$

authors in [16] drew an intuition that the earlier layers extract low-level variables used by later layers and, therefore, should be more reliably present, i.e., b_q should more possibly be zero than b_{q-1} and so forth. In practice, they introduced:

$$G_q = ReLU(\beta_q(G_{q-1}B_q) + id(G_{q-1})), \tag{4}$$

where β_q is the survival probability of layer q, which is a hyperparameter that can be predefined by:

$$\beta_q = 1 - \frac{q}{Q}(1 - \beta_Q), \tag{5}$$

where β_Q is set as 0.5.

Normally as the ResNet goes deeper, we have more opportunities to discover complicated nonlinearity on the temporal level. However, in practice it is difficult to know which layer (level of nonlinearity) is more relevant to the underlying causal graph. In fact, their relevance to the underlying causal graph varies on different datasets. Here, we design their relevance to be learnable variables. It remains the same as Equation (4), but all $\beta_q, q = 1, 2, ..., Q$ become variables. Figure 5 shows the learned relevance of different layers across epochs for two datasets (described in the Experiment section). It indicates that during training, the relevance converge fast to a stable status, and the underlying causal graph has different relevance to different level of temporal nonlinearity on different datasets. This demonstrates the rationality behind our design of Module 1.

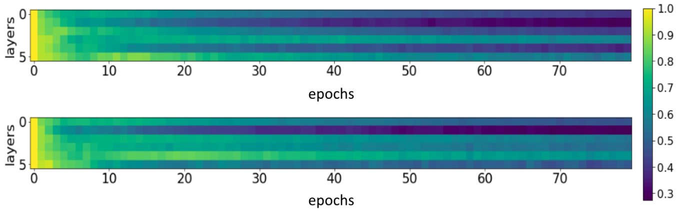


Figure 5: The learned relevance of different layers to the underlying causal graph of two datasets (described in the Experiment section) across epochs. Y axis is the layer index, and X axis is the epoch index. It is evident that: 1) the relevance vector converges to a stable status, and 2) different datasets may assign distinct relevance to each layer, which motivates us to make the relevance learnable.

Specifically, the input layer of Module 1 is $X(*, t_j - \ell + 1 : t_j) \in \mathbb{R}^{m \times \ell}$, and the output of the q-th residual layer is $G_q \in \mathbb{R}^{m \times p}$. Each G_q represents a certain level of temporal nonlinearity. Given Q is the total number of residual blocks, $G_{1:Q}(i, *)$ can approximate $G(i, *)$ in Figure 3. So we use the output of all the blocks, i.e., $G_{1:Q}(*, *)$, as the input to the next module.

Before we describe the rest of the *SCGL* model, please note:

(1) Each column in B_1 can be treated as a one-dimension convolution filter with size ℓ. Each filter performs on all of the m time series, respectively.

(2) In this module, we focus only on the temporal nonlinearity on univariate level without rolling information between time series. The reason is that such rolling should only involve the factor variables of each time series, which should be controlled by the causal graph learned in Module 2. If we involve it too early, it will disrupt the causal graph learning.

(3) Apart from traditional ResNet that only makes use of the last layer output, we feed the output from all layers to the causal learning module. The motivation is the same as for introducing layer relevance: causality also may exist on the shallow level of temporal nonlinearity. Therefore, in our model, we feed different levels of temporal nonlinearity into the next module to explore their effects on the underlying causal graph.

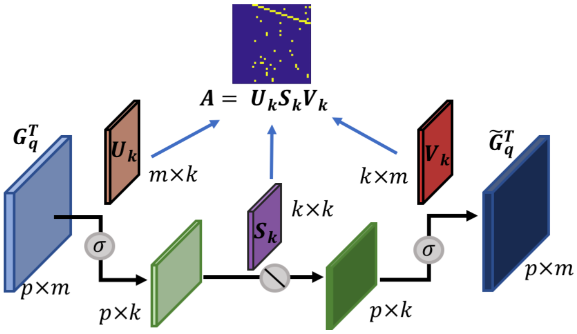


Figure 6: Module 2 learns the causal graph using low-rank approximation.

3.2 Module 2: Learning Causal Graph

The second module is the *SCGL* model's key component, which is to learn the causal graph. For each $G_q \in G_{1:Q}(i, *)$ from Module 1, we want to select those variables, that only contribute to predict each $X(i, t_j + 1), i = 1, 2, ..., m$. Given a causal graph A_q, this can be done by

$$\tilde{\mathcal{G}}_q^T(*, i) = G_q^T A_q(*, i), \quad i = 1, 2, ..., m \tag{6}$$

Such an operation can be extended to the whole variable space. However, when the size of variables (m) is quite large, learning the full size of $A \in \mathbb{R}^{m \times m}$ would be unscalable.

In practice, the number of factor variables usually is small, and the relationship between variables is low-rank in hidden space [10, 36, 37]. In our model, we approximate A through a k-rank matrix decomposition with $k < m$.

Figure 6 illustrates the Module 2 design. For each of the nonlinear temporal embedding G_q from Module 1, Module 2 projects them from the original m space into k-rank embeddings (latent factor space). This is done via nonlinear mapping with a weight matrix $U_{q,k} \in \mathbb{R}^{m \times k}$. Then, a scaling matrix (diagonal matrix $S_{q,k} \in \mathbb{R}^{k \times k}$) will be learned to scale each low dimension. Finally, the k-rank embeddings will be projected back to original m space with a weight matrix $V_{q,k} \in \mathbb{R}^{k \times m}$. The following equation describes this process:

$$\tilde{\mathcal{G}}_q^T = \sigma((\sigma(G_q^T U_{q,k})S_{q,k})V_{q,k}), \tag{7}$$

where $U_{q,k}$ contains factor mapping information, $V_{q,k}$ contains information from effect mapping, and $S_{q,k}$ are scaling factors. The causal graph A_q can be approximated by

$$A_q \approx U_{q,k} S_{q,k} V_{q,k}. \tag{8}$$

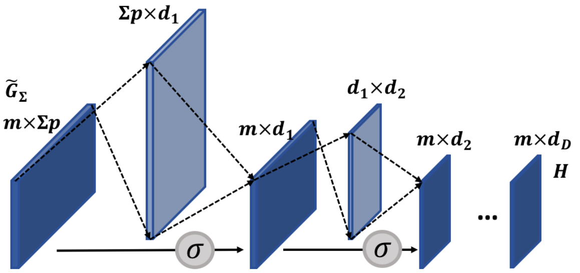


Figure 7: Module 3 is used to learn intervariable nonlinearity via a simple FC chain.

The number of parameters needed to learn through Equation (8) is $2mk + k^2$, which is much smaller than m^2 when $k \ll m$. By experiment, we set the first and second activation functions as *tanh* and *Relu*, which give the best performance. Notably, we did not add any activation function on Equation (8) because our causal graph construction is not based on the input G_q^T but instead on the relative weights of factor, effector and scalar matrix, i.e., $U_{q,k}S_{q,k}V_{q,k}$. Our empirical studies also confirm our choice.

Similar to traditional SVD, we force the $U_{q,k}$ and $V_{q,k}$ to be orthonormal, which is done by applying weight penalty $|U_{q,k}^T U_{q,k} - I|$ and $|V_{q,k}^T V_{q,k} - I|$ to loss function. Orthogonality regularization is discussed and popularly used [4, 6, 34]. By doing so, the weight matrix is maintained as a unitary matrix that preserves a vector's length. The constructed embedding also has theoretical connection to Stiefel manifolds and spectral theorem [30].

For each G_q from Module 1, we perform the preceding procedure and get A_q, respectively. The global causal graph A is obtained by

$$A \approx \sum_{q=1}^{Q} \beta_q A_q, \tag{9}$$

where β_q is the learned relevance of G_q from Module 1.

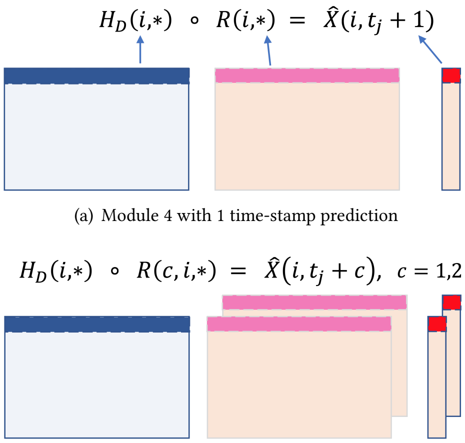
3.3 Module 3: Learning Intervariable Nonlinearity

To learn the nonlinearity among the time series, we designed Module 3. The input is all $\tilde{\mathcal{G}}_q \in \mathbb{R}^{m \times p}, q = 1, 2, \ldots Q$ from Module 2 (Equation (7)). We concatenate them column-wisely and denote the whole matrix as: $\tilde{\mathcal{G}}_\Sigma \in \mathbb{R}^{m \times \Sigma p}$. Module 3 consists of a series of fully connected layers, and the j-th layer output is defined as:

$$H_j = \sigma(H_{j-1} W_j + b_j), \quad j = 1, 2, \ldots, D \tag{10}$$

where $H_{j-1} \in \mathbb{R}^{m \times d_{j-1}}$ is the output of layer $j-1$, $W_j \in \mathbb{R}^{d_{j-1} \times d_j}$ is a weight matrix, and b_j is a bias vector. The activation function here is *tanh*. We set $H_0 = \tilde{\mathcal{G}}_\Sigma$. The final output of this module is $H_D \in \mathbb{R}^{m \times d_D}$ where D is the total number of layers in Module 3.

Intuitively, $\tilde{\mathcal{G}}_\Sigma(i, *)$ contains linear combinations of all the temporal nonlinearity from the factor variables of time series $X(i, *)$. Module 3 rolls them in a nonlinear way. In other words, our model learns intervariable nonlinearity using Module 3.


$H_D(i,*) \circ R(i,*) = \hat{X}(i, t_j + 1)$
$H_D(i,*) \circ R(c,i,*) = \hat{X}(i, t_j + c), \; c = 1,2$

(a) Module 4 with 1 time-stamp prediction

(b) Module 4 with 2 time-stamp prediction

Figure 8: Module 4 predicts the regression target.

3.4 Module 4: Regression

The final module predicts the regression target $\hat{X}(*, t_i + 1) \in \mathbb{R}^{m \times 1}$ using the output H_D from Module 3. Here, the model learns a weight matrix $R \in \mathbb{R}^{m \times d_D}$. The regression is performed by a row-wise dot product (Figure 8(a)):

$$\hat{X}(i, t_i + 1) = H_D(i, *) \circ R(i, *), \quad i = 1, 2, \ldots, m \tag{11}$$

The residual between the predicted $\hat{X}(*, t_i + 1)$ and the actual $X(*, t_i + 1)$ is calculated by mean squared error (MSE) with regularization terms. Our loss function is defined as:

$$\begin{aligned} loss = & \frac{1}{mn} \sum \left(X(*, t_i + 1) - \hat{X}(*, t_i + 1) \right)^2 \\ & + \lambda_1 \sum_q \left(\left\| U_{q,k} \right\|_2 + \left\| V_{q,k} \right\|_2 \right) \\ & + \lambda_2 \sum_q \left(\left\| U_{q,k}^T U_{q,k} - I \right\|_2 + \left\| V_{q,k}^T V_{q,k} - I \right\|_2 \right), \end{aligned} \tag{12}$$

where n is the total number of samples and m is the number of variables. We pose two regularization terms. The first aims to avoid over-fitting on the causal graph, while the second one is used for imposing orthonormality to U and V (as described in Section 3.2). Then, the calculated error would be backpropagated through the model to update its parameters.

Intuitively, the i-row of H_D contains all of the **possible contributing nonlinear forms** from the factors of $X(i, t_j + 1)$, where $i = 1, 2, \ldots, m$. Module 4 constructs their optimal combination to approach $X(i, t_j + 1)$. In practice, we predict more than one future timestamp. Figure 8(b) shows that we predict $X(*, t_j + 1 : t_j + 2)$ by learning two layers of R respectively. The motivation here is that by involving more timestamps as regression targets, we may be able to learn a more accurate causal graph A because we have more information in every training step. We will verify this intuition further in the Experiment Section.

4 DISCUSSION

In this section, we examine our proposed SCGL method, including its connection with other causal graph learning theories and methods. We also justify the utility of our *SCGL* model by briefly discussing the connection and distinction with a few other existing methods.

Granger Causality. In [12], Granger causality detects the temporal causal time-lag relationship of two time series. The proposed method in [5] first establishes a VAR model on two variables then performs F-statistics tests on the residuals of one variable with and without considering the other. This method can be performed m^2 times to learn the full causal graph. Among all the limitations of this method, the most notable one is its high false alarm rate because it only models two variables at a time without considering the effect from other variables. Later works such as [17, 35] are built on pairwise conditional Granger causality where all m variables are involved in VAR. However, such methods have three limitations. First, it requires performing F-tests to detect any causal relationship between two variables. It assumes normal distribution on the dataset, which is not always true in real applications. Second, the causal graphs are still constructed by a pairwise relationship, even though we can include other conditional variables. Therefore, any more than three joint relationships cannot be effectively reflected in the final causal graph. Third, the VAR model is still inherently linear. In contrast, our proposed approach does not rely on any distribution assumption or pairwise causal graph construction, and can detect nonlinear causality.

Transfer Entropy. It was firstly proposed in [27] for learning pairwise causality without distribution assumption and linear setting. One variable X is considered a cause of another variable Y if the past values of X significantly decrease the uncertainty, measured by Shannon entropy, in the future value of Y given its past. Such method is extended in the work [20] to be more robust and computationally efficient. However, this type of methods only detect pairwise causality and thus share similar limitations of Granger Causality in this aspect.

Graph Learning. Research such as [1, 2, 7] learn causality by adding lasso or ridge regularization to VAR. The causality between any two variables can be learned by detecting whether or not the sum of weights of these two variables across all timestamps is close to zero. These methods generate causal graph in the form of graph adjacency matrix in one shot. However, it can only learn linear temporal relationship. On the contrary, our *SCGL* method is designed to capture complicated nonlinear uni- and inter-variate relationships.

Kernel-based Methods. Work in [19, 24, 25, 28] proposed a series of conditional Granger causality analysis using kernel-based VAR model. Several types of kernels are used to capture the nonlinear temporal relationship. However, such methods come with a heavy computational cost. For example in [28], it is claimed that their computational complexity is $O(m^3 + n^3)$, where m is the number of variable and n is the number of total timestamps. On the other hand, our model is more scalable. Please refer to Section 5.7 for scalability comparison.

Supervised Learning Methods. Works such as [8, 21] formulate the causal inference problem as a cause-effect classification problem. Concretely, the classification model request relationship labels: 1 (X causes Y), -1 (Y causes X), and 0 (no causal relationship) between certain number of variable pairs. Technically, this is achieved by mapping the conditional distribution of X with and without a previous state of Y to a point by kernel mean embeddings, then calculating the distance as a metric of causality classification in the reproducing kernel Hilbert space. However, such methods request labeled relationship for training, which is quite difficult to collect in many cases. Comparatively, our *SCGL* learns a full causal graph through time series regression without any label.

MLP and LSTM. Work in [31, 32] proposed multi-layer perception (MLP) and a long short-term memory (LSTM) deep learning framework to learn causal graph. They all try to learn causal graph through time series regression. The MLP-based method is considered as an extension of the VAR method with more hidden layers to capture nonlinear relationships, while LSTM is widely used in sequential data. Both methods learn causality by detecting if the sum of the weights in the first neural network layer is close to zero or not. However, this kind of method has worse performance than our *SCGL* method because they do not have a systematical way to learn different type of nonlinearity (please refer to Experiment Section).

Graphical Neural Network. Work in [18] use graph neural network to learn latent interaction in multi-variate dynamics system. It is a type of message passing neural network where node-edge message passing operations are defined through multi-layer perceptron (MLP) for encoder and LSTM or MLP as decoder. However, 1) the designed model is for a general relationship learning, not causal relationship learning, and 2) there is no low-rank approximation to learn causal relationship. Therefore, these models suffer from overfitting and are not scalable to large number of variables. On the other hand, our *SCGL* is designed for more general types of interaction graph, and have good scalability both in time and in memory.

5 EXPERIMENT

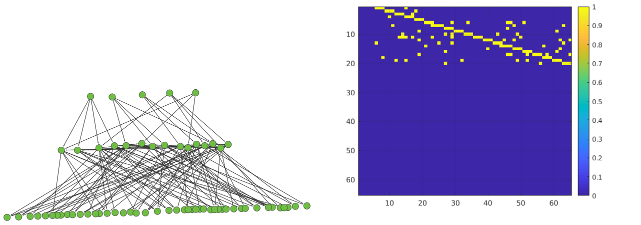
This section demonstrates the superior capability of our *SCGL* method via a thorough comparison with several popular baselines on both synthetic and real-world datasets [2].

5.1 Experiment Setup

Construction of Synthetic Dataset

For simulation test, we constructed two datasets. We first construct the causal graphs to build the datasets and later use them as groundtruth to evaluate our result. Based on the constructed causal graph, we then generate the time series data with certain nonlinear relationships. Specifically, the underlying causal graph is generated by initially designing a three-layer hierarchical structure. Nodes in the top layer serve as the cause of nodes in the second layer. The second layer nodes then are the cause of the bottom layer nodes. To increase the complexity, we add random directed edges between nodes within the second layer. We call the nodes in the top layers as *masters*, and the remaining nodes as *effectors*. This causal structure is very common in biology science (e.g. gene regulatory network).

[2] The code and simulated datasets are available in https://drive.google.com/open?id=1Py6Sr108ccfO-mxR2CxaoSGTMsTxBQqS



(a) Hierarchical causal structure with 5-15-45 nodes distribution

(b) The corresponding causal graph. $A(i, j) = 1$ means node i is the factor of node j.

Figure 9: Causal graph for *Synthetic Dataset A*.

We assign 5-15-45 nodes in the three layers of *Synthetic Dataset A*, while *Synthetic Dataset B* has a hierarchical structure of 10-30-90 nodes. Figure 9 illustrates the causal structure and causal graph of *Synthetics Dataset A*. Based on the causal graph, we design the temporal causal relationship as follows:

Synthetic Dataset A. For *masters*, we pose a self-regulating relationship with time-lag 2:

$$X(i, t_j) = \cos(\sqrt{3}d(i) \times X(i, t_j - 1) \times d(i)^2 \times X(i, t_j - 2)) + \epsilon(i, t_j), \tag{13}$$

where $d(i)$ is the decay factor of node i that randomly selected from uniform distribution $\mathcal{U}(0.95, 1)$, and $\epsilon(i, t_j)$ is the random noise sampled from normal distribution $\mathcal{N}(0, 1)$. The expression value of *effectors* is generated by

$$X(k, t_j) = \sum_{l=1}^{m} \log((r(l \Rightarrow k) \times X(l, t_j - p(l \Rightarrow k)))^2) + \epsilon(k, t_j), \tag{14}$$

with $r(l \Rightarrow k)$ following $\mathcal{U}(-1, 1)$ if node l is among the factors of node k and 0 otherwise. Term $p(l \Rightarrow k)$ represents the time lag of the temporal causal relationship from l to k and is randomly selected from $\{1, 2, 3\}$. The first three timestamp values of all nodes are randomly generated from $\mathcal{N}(0, 1)$. Thus, *Synthetic Dataset A* is designed to have the model order 3.

Synthetic Dataset B. For *masters*, we constitute the following self-regulating relationship with time-lag 5:

$$X(i, t_j) = \arctan(\sqrt{3}d(i) \times X(i, t_j - 1) + d(i)^2 \times X(i, t_j - 5)^2) + \epsilon(i, t_j). \tag{15}$$

The expression value of *effectors* is simulated in a more complicated way. Here we introduce an intermediate variable $Z_l(k, t_j)$ as

$$Z_l(k, t_j) = r(l \Rightarrow k) \times X(l, t_j - p(l \Rightarrow k)), \tag{16}$$

where $r(*)$ and $p(*)$ are defined in the same manner as *Synthetic Dataset A*. $Z_l(k, t_j)$ then represents the contributing value of node l to node k at timestamp t_j. We then let

$$X(k, t_j) = \sum_{l=1}^{m} (Z_l^2(k, t_j) + \tanh(Z_l(k, t_j) \times X(k, t_j - p(l \Rightarrow k)))) \tag{17}$$

$$+ \sum_{s=1}^{m} \sum_{l=1}^{m} (\sin(Z_s(k, t_j) \times Z_l(k, t_j))) + \epsilon(k, t_j).$$

As such, the entire time series dataset is designed to have model order 5. Here, the value of node k at timestamp t_j is affected by the interaction of its previous value with its factor node l. It is also affected by the interaction of previous values of node l and node s if they both are node k's factors.

Evaluation Metrics and Baseline Methods

To evaluate the learned causal graph against ground truth, we use two metrics: area under precision and recall curve (**AUPR**) and area under receiver operating characteristics curve (**AUROC**). We compare our *SCGL* method with the following five popular baselines:

1. *VAR* [1]. It is one of the most well-known methods, which applies the vector autoregressive model with ridge regularization to learn causal graph.
2. *PCKGC* [25]. It stems from pairwise conditional kernel-based Granger causality analysis [24] and only considers the partial conditioning to a pre-selected variable subset with the highest mutual information scores to save the heavy computational cost in original methods.
3. *Copula* [3]. It transforms the marginal distribution of each variable to Gaussian domain and then apply VAR graph learning methods. Certain levels of high order temporal relationship can be captured. We used ridge regularization instead of original lasso for better performance in our experiments.
4. *cLSTM* [32]. It applies LSTM to past values of all variables to regress future value with group lasso regularization on the model parameters of the first neural network layer. The causal graph is then generated from the absolute sum of all the corresponding parameters.
5. *pTE* [20]. It calculates the phase transfer entropy by transforming temporal signals of each variable to discrete phases with certain histogram-based probability functions.

5.2 Comparison on Synthetic Datasets

For experiments on *Synthetic Dataset A*, we use the following hyperparameter setting: the input time series window is set to 3, and the prediction window size (pre-win) is set to 2. We set the number of temporal layers $Q = 5$ in Module 1, each with dimension $p = 20$ and low-rank embedding $k = 30$ in Module 2. We use a single fully connected layer with dimension $d_1 = 50$ in Module 3. For *Synthetic Dataset B*, we set the input time series window to be 5 and the prediction window size as 3. We set the number of temporal layers $Q = 6$ with dimension $p = 20$ and low-rank embedding $k = 55$. In Module 3, we use a single fully connected layer with dimension $d_1 = 50$. For both experiments, we use 80% of the data for training and the remaining data for validation. For more details please refer to our uploaded code. For the other baselines, we tune their hyperparameters to the best of our knowledge.

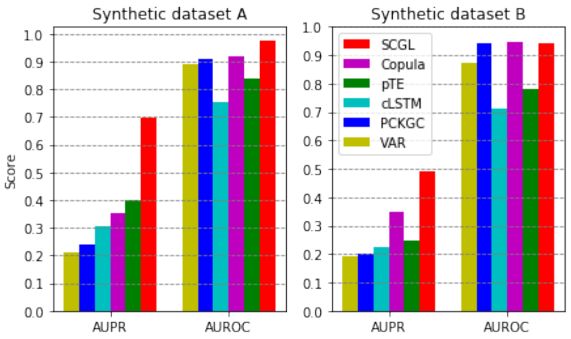


Figure 10: Comparison of AUPR and AUROC of all methods.

Figure 10 shows the comparison result on the two synthetic datasets. Apparently, our *SCGL* is superior to all the baselines, especially as the ground truth causal graph is quite sparse and AUPR is the more appropriate metric [26]. The AUPR by our *SCGL* is significantly larger than all the baselines (40%+ improvement at least). The AUROC of our *SCGL* is also the best. The *pTE* and *Copula* are the second and third best methods, since they do not require any statistical assumptions and can capture certain level of nonlinearity. Comparatively, *cLSTM* gives a little bit lower performance. It only uses parameters of the first layer to detect causality, which tends to provide a relatively lower recall rate with a given precision rate. *PCKGC* and *VAR* have the worst performance. It is because that *PCKGC* only detects the conditional Granger causality which is conditioned to a pre-selected subset of variables, while *VAR* suffers from its linear regression setting.

To show the convergence of our *SCGL* method, we plot the training loss, validation loss, and AUPR across training epochs in Figure 11. First, it shows that both the training loss and validation loss converge well, proving that our *SCGL* model is reasonably stable. Second, it is evident that *AUPR* rises simultaneously as loss converges, which illustrates that the quality of the learned causal graph improves as the prediction error converges.

5.3 Comparison on a Real-world Dataset

This section details the causal learning test on real-world engine operating data. The dataset contains six months data collected from 50 different engines. The number of total training samples exceeds $180k$. For privacy reason, we didn't include sensitive descriptions. In this test, we use the following setting in *SCGL*: input time series window = 10, pre-win= 2, $Q = 3$, $p = 20$, $k = 10$ and $d_1 = 40$.

Figure 12 shows the comparison of the learned causal graphs. Due to space limitation, we only show 14 key variables by the *SCGL* and the best result from baselines (by qualitative study on this dataset). Based on knowledge from domain experts, our *SCGL* model successfully captures the following causal relationships:

(1) Corrected Altitude (BrCrtAlt) measures altitude based on a function of total pressure (TtlPrs) and total air temperature (TtlAirTmp).
(2) Computed Airspeed (CmptAirSp) is subject to air-density changes, which is relevant to BrCrtAlt.

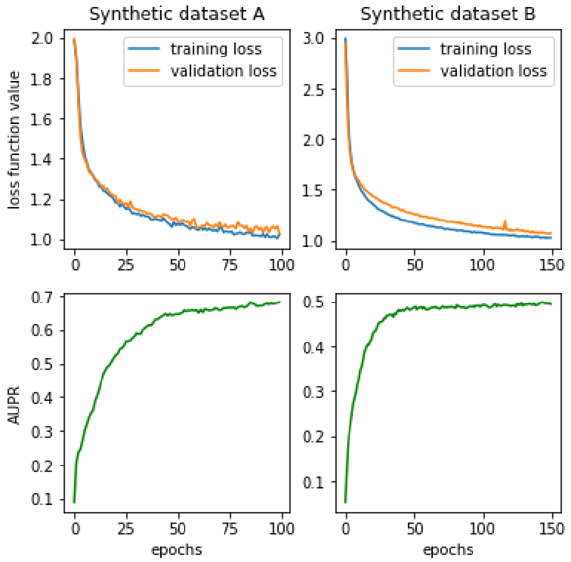


Figure 11: Training/Validation loss and AUPR across epochs.

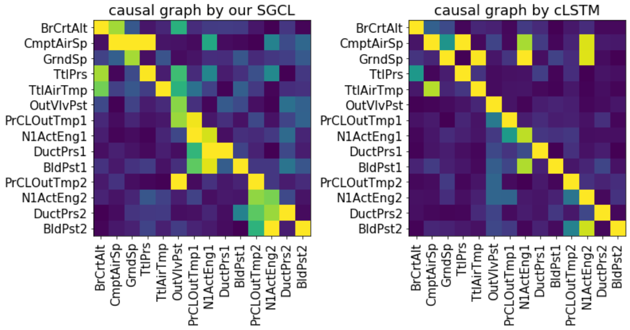


Figure 12: Learned causal graph of real-world data.

(3) Adjustment of fan air flow rate, which is relevant to fan speed (N1ActEng), provides a desired cooling outlet temperature (PrCLOutTmp).
(4) Fan speed (N1ActEng), which measures the rotational speeds of engine sections, is affected by Pressure (DuctPrs) and Bleed Position (BldPst).
(5) Outflow Valve is the actuator of the Cabin Pressure Regulating System. The change of Valve Position (OutVlvPst) is triggered by the change of BrCrtAlt and PrCLOutTmp.

Compared against our method, *cLSTM* and the other baselines fail to discover these physical relationships between sensors.

5.4 Robustness against Noise Levels

In Figure 13, we evaluate the robustness of all models against various noise levels. We injected three noise levels 20%, 35%, and 50% into data, e.g., noise 20% is added in the following way:

$$X_{20\%}(i, t_j) = X(i, t_j) \times \mathcal{U}(1 - 20\%, 1 + 20\%) \tag{18}$$

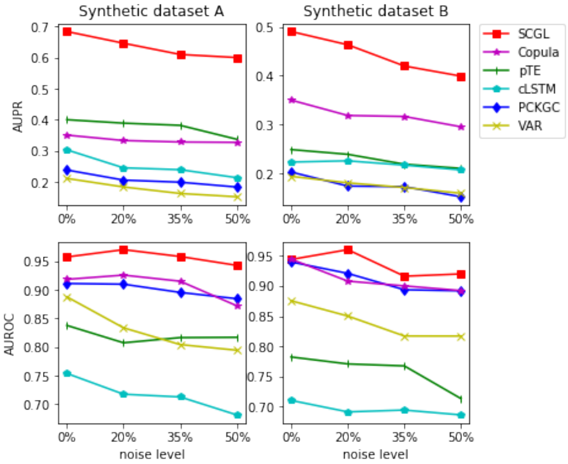


Figure 13: Comparison with various noise levels.

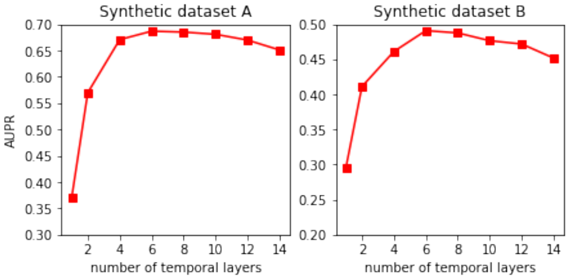


Figure 14: AUPR with various temporal layers in Module 1.

where $i = 1 \cdots m$ and $t_j = 1 \cdots T$, and $\mathcal{U}$ indicates uniform distribution. Generally, all methods' performance drop as noise increases. However, the *SCGL* method still maintains AUPR no less than 0.6 and 0.4 for the two datasets respectively even with 50% noise. This confirms *SCGL*'s robustness against noise by low-rank embedding, and our discussion in Section 3.2.

5.5 Effect of Number of Low-rank and Multiple Temporal Layers

Here, we discuss the effect of two important hyperparameters in *SCGL*. In Figure 14, we test the AUPR by our *SCGL* model with various numbers of temporal layers (Q in Module 1). It can be observed that by increasing Q, AUPR increases in the beginning but gradually decreases after reaching its peak. This tells that a certain number of layers are already enough to capture the nonlinear relationship on univariate level. But adding more layers may lead to overfitting.

In Figure 15, we test various values of k for k-rank embeddings in causal graph learning in Module 2. It shows that our model can learn a more accurate causal graph with reasonably large k. However, AUPR gradually decreases as k is getting too large. This shows that the dimension of low-rank should be large enough to learn complicated causality, but small enough to exclude noise influence from data and modeling process.

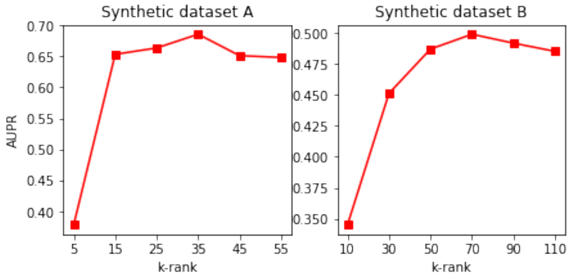


Figure 15: AUPR with various k-rank in Module 2.

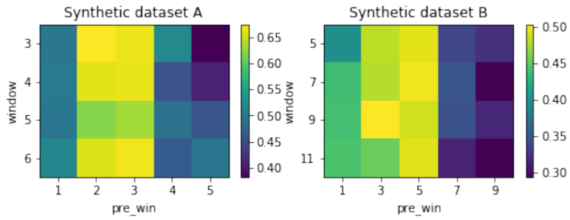


Figure 16: AUPR of various size of input window and predicting-window (pre-win) on two synthetic datasets

.

5.6 Robustness against Input Window and Pre-win Size

Figure 16 shows AUPR with different numbers of input time windows (denoted as *window*) and timestamps that we want to predict (noted as *pre-win*) on two synthetic datasets. We find that AUPR does not change much by increasing the time windows beyond the model order, which means that the temporal convolution in Module 1 correctly weights different time-lags during training. On the other hand, we found that for a given input window size, the AUPR decreases with too small or too large pre-win. One possible reason is that more than one predicting windows can involve more information that may contribute to learning better causal graph. But too many pre-wins will increase the difficulty in prediction, which causes more and more instability and leads to poor quality of the learned causal graph.

5.7 Scalability Comparison

In Figure 17, we evaluate the scalability of all methods with increasing number of variable (m). All of the experiments are performed with Intel Xeon E5-2670 processors with 128G memory. Figure 17 shows that the computing time of *VAR, Copula, PCKGC* and *pTE* are in the order of $O(m^3)$, while *SCGL* and *cLSTM* are in relatively lower order. However, *SCGL* has much better accuracy than *cLSTM* in learning causal graph. This confirms that our *SCGL* is both scalable and effective.

PCKGC consumes too much time when the number of variable reaches 10^3 or greater, whereupon we stopped measuring its computing time.

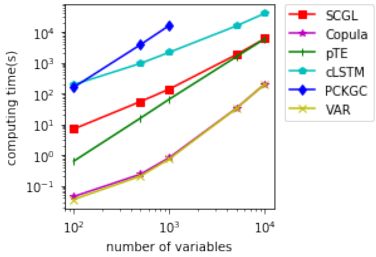


Figure 17: Scalability test of all models.

6 CONCLUSION

In this work, we present a scalable causal graph learning method, or SCGL, for multivariate time series data by deep neural network. This method can discover complex causality without any predefined kernel or distribution assumptions. Specifically, it learns the underlying causal graph through low-rank approximation, which makes the model more scalable and robust against noise. Using both synthetic and real-life datasets, we show that our proposed *SCGL* method outperforms other baselines by a significant margin.

ACKNOWLEDGMENTS

This work is supported in part by United State, Department of Energy, under Contract Number DE-SC0012704.

REFERENCES

[1] Andrew Arnold, Yan Liu, and Naoki Abe. 2007. Temporal Causal Modeling with Graphical Granger Methods. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*. ACM, New York, NY, USA, 66–75. https://doi.org/10.1145/1281192.1281203

[2] Mohammad Taha Bahadori and Yan Liu. [n. d.]. *Granger Causality Analysis in Irregular Time Series*. 660–671. https://doi.org/10.1137/1.9781611972825.57 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611972825.57

[3] Mohammad Taha Bahadori and Yan Liu. 2013. An examination of practical granger causality inference. In *Proceedings of the 2013 SIAM International Conference on data Mining*. SIAM, 467–475.

[4] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems*. 343–351.

[5] Steven L Bressler and Anil K Seth. 2011. Wiener–Granger causality: a well established methodology. *Neuroimage* 58, 2 (2011), 323–329.

[6] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. 2016. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093* (2016).

[7] Dehua Cheng, Mohammad Taha Bahadori, and Yan Liu. 2014. FBLG: A Simple and Effective Approach for Temporal Dependence Discovery from Time Series Data. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 382–391. https://doi.org/10.1145/2623330.2623709

[8] Yoichi Chikahara and Akinori Fujino. 2018. Causal Inference in Time Series via Supervised Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2018/282

[9] Belkacem Chikhaoui, Mauricio Chiazzaro, and Shengrui Wang. 2015. A new granger causal model for influence evolution in dynamic social networks: The case of dblp. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[10] Alessandro Chiuso and Gianluigi Pillonetto. 2012. A Bayesian approach to sparse dynamic network identification. *Automatica* 48, 8 (2012), 1553–1565.

[11] Cees Diks and Valentyn Panchenko. 2006. A new statistic and practical guidelines for nonparametric Granger causality testing. *Journal of Economic Dynamics and Control* 30, 9-10 (2006), 1647–1669.

[12] Clive Granger. 1969. Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Econometrica* 37, 3 (1969), 424–38. https://EconPapers.repec.org/RePEc:ecm:emetrp:v:37:y:1969:i:3:p:424-38

[13] David Gunning. 2017. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web* (2017).

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[15] Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. 2009. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*. 689–696.

[16] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. 2016. Deep networks with stochastic depth. In *European conference on computer vision*. Springer, 646–661.

[17] Dominik Janzing, David Balduzzi, Moritz Grosse-Wentrup, and Bernhard Scholkopf. 2013. Quantifying causal influences.

[18] Thomas N. Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard S. Zemel. 2018. Neural Relational Inference for Interacting Systems. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 2693–2702. http://proceedings.mlr.press/v80/kipf18a.html

[19] Néhémy Lim, Florence DâĂŹAlché-Buc, Cédric Auliac, and George Michailidis. 2015. Operator-valued kernel-based vector autoregressive models for network inference. *Machine learning* 99, 3 (2015), 489–513.

[20] Muriel Lobier, Felix SiebenhÃijhner, Satu Palva, and J. Matias Palva. 2014. Phase transfer entropy: A novel phase-based measure for directed connectivity in networks coupled by oscillatory interactions. *NeuroImage* 85 (2014). https://doi.org/10.1016/j.neuroimage.2013.08.056 New Horizons for Neural Oscillations.

[21] David Lopez-Paz, Krikamol Muandet, Bernhard SchÃűlkopf, and Iliya Tolstikhin. 2015. Towards a Learning Theory of Cause-Effect Inference. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 1452–1461. http://proceedings.mlr.press/v37/lopez-paz15.html

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

[23] Daniele Marinazzo, Wei Liao, Huafu Chen, and Sebastiano Stramaglia. 2011. Nonlinear connectivity by Granger causality. *Neuroimage* 58, 2 (2011), 330–338.

[24] D. Marinazzo, M. Pellicoro, and S. Stramaglia. 2008. Kernel-Granger causality and the analysis of dynamical networks. *Phys. Rev. E* 77 (May 2008), 056215. Issue 5. https://doi.org/10.1103/PhysRevE.77.056215

[25] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. 2012. Causal information approach to partial conditioning in multivariate data sets. *Computational and mathematical methods in medicine* 2012 (2012).

[26] Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one* 10, 3 (2015), e0118432.

[27] Thomas Schreiber. 2000. Measuring Information Transfer. *Phys. Rev. Lett.* 85 (Jul 2000), 461–464. Issue 2. https://doi.org/10.1103/PhysRevLett.85.461

[28] Vikas Sindhwani, Minh Ha Quang, and Aurélie C Lozano. 2012. Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and granger causality. *arXiv preprint arXiv:1210.4792* (2012).

[29] Linda Sommerlade, Marco Thiel, Bettina Platt, Andrea Plano, Gernot Riedel, Celso Grebogi, Jens Timmer, and Björn Schelter. 2012. Inference of Granger causal time-dependent influences in noisy multivariate time series. *Journal of neuroscience methods* 203, 1 (2012), 173–185.

[30] Petre Stoica and Randolph L Moses. 1997. *Introduction to spectral analysis*. Vol. 1. Prentice hall Upper Saddle River, NJ.

[31] Alex Tank, Ian Cover, Nicholas J Foti, Ali Shojaie, and Emily B Fox. 2017. An interpretable and sparse neural network model for nonlinear granger causality discovery. *arXiv preprint arXiv:1711.08160* (2017).

[32] Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily Fox. 2018. Neural granger causality for nonlinear time series. *arXiv preprint arXiv:1802.05842* (2018).

[33] Martin Vinck, Lisanne Huurdeman, Conrado A Bosman, Pascal Fries, Francesco P Battaglia, Cyriel MA Pennartz, and Paul H Tiesinga. 2015. How to detect the Granger-causal flow direction in the presence of additive noise? *Neuroimage* 108 (2015), 301–318.

[34] Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal. 2017. On orthogonality and learning recurrent networks with long term dependencies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 3570–3578.

[35] Shun Yao, Shinjae Yoo, and Dantong Yu. 2015. Prior knowledge driven Granger causality analysis on gene regulatory network discovery. *BMC Bioinformatics* 16, 1 (28 Aug 2015), 273. https://doi.org/10.1186/s12859-015-0710-1

[36] Mattia Zorzi and Alessandro Chiuso. 2015. A Bayesian approach to sparse plus low rank network identification. In *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 7386–7391.

[37] Mattia Zorzi and Alessandro Chiuso. 2017. Sparse plus low rank network identification: A nonparametric approach. *Automatica* 76 (2017), 355–366.