Generating Fine-Grained Causality in Climate Time Series Data for Forecasting and Anomaly Detection

Dongqi Fu [1] **Yada Zhu** [2 3] **Hanghang Tong** [1] **Kommy Weldemariam** [4] **Onkar Bhardwaj** [3] **Jingrui He** [1]

Abstract

Understanding the causal interaction of time series variables can contribute to time series data analysis for many real-world applications, such as climate forecasting and extreme weather alerts. However, causal relationships are difficult to be fully observed in real-world complex settings, such as spatial-temporal data from deployed sensor networks. Therefore, to capture fine-grained causal relations among spatial-temporal variables for further a more accurate and reliable time series analysis, we first design a conceptual fine-grained causal model named *TBN Granger Causality*, which adds time-respecting Bayesian Networks to the previous time-lagged Neural Granger Causality to offset the instantaneous effects. Second, we propose an end-to-end deep generative model, named *TacSas*, which discovers TBN Granger Causality in a generative manner to help forecast time series data and detect possible anomalies during the forecast. For evaluations, besides the causality discovery benchmark Lorenz-96, we also test TacSas on climate benchmark ERA5 for climate forecasting and the extreme weather benchmark of NOAA for extreme weather alerts.

1. Introduction

"*Climate science investigates the structure and dynamics of earth's climate system and seeks to understand how global, regional, and local climates are maintained as well as the processes by which they change over time*",[1] where the corresponding data are usually stored in the format of time series, recording the climate features, geo-locations, time attributes, etc.

In time series data, variables often exhibit high-dimensional characteristics, and correlation between variables tends to be intricate, hard to obtain, and encompassing aspects such as non-linearity and time dependency. Taking the climate time series data as an example, multiple variables such as temperature, wind gust, atmospheric water content, and solar radiation co-appear on the time axis. Although we can access their tabular representations, their interactions are typically complex (e.g., non-linear, time-dependent), making it difficult to understand and capture the time series evolution trend and latent distribution of values. As a result, this complexity may lead to sub-optimal performance in time series analysis, such as time series forecasting and anomaly detection.

Structure learning has recently gained much attention, such as (Li et al., 2018; Wu et al., 2020; Zhao et al., 2020; Cao et al., 2020; Shang et al., 2021; Deng & Hooi, 2021; Marcinkevics & Vogt, 2021; Geffner et al., 2022; Tank et al., 2022; Spadon et al., 2022; Fu & He, 2022; Zhou et al., 2022; Gong et al., 2023; Fu et al., 2023; Li et al., 2023b; Fu et al., 2024). Among others, causal graphs as a directed acrylic graph structure provide more explicit and interpretable correlations between variables, thus enabling a better understanding of the underlying physical mechanisms and dynamic systems for time series (Kofinas et al., 2023; 2021). As a widely applied causal structure in time series understanding and explanation, Granger Causality (Granger, 1969; Arnold et al., 2007) discovers causal relations among variables in an autoregressive (or time-lagged) manner. The discovered Granger causal structures can help many time series analysis tasks, like building parsimonious prediction models such as Earth System (Runge et al., 2019). Moreover, real-world time series data can have many variables, and their causal relations can be even more complex, i.e., non-linear and instantaneous, which require complex causality discovery beyond the classic Granger model. Although some nascent non-linear (or neural) Granger models have been proposed (Nauta et al., 2019; Xu et al., 2019; Tank et al., 2022; Khanna & Tan, 2020; Huang et al., 2020; Pamfil et al., 2020; Marcinke-

[1]University of Illinois Urbana-Champaign, USA [2]IBM Research, USA [3]MIT-IBM Watson AI Lab, USA [4]Amazon Sustainability Science and Innovation, USA. Correspondence to: Jingrui He <jingrui@illinois.edu>.

ICML 2024 AI4Science Workshop, Vienna, Austria.

[1]https://plato.stanford.edu/entries/climate-science/

vics & Vogt, 2021; Geffner et al., 2022), how to effectively integrate instantaneous causal effects with neural Granger models has the great research potential (Moneta et al., 2013; Wild et al., 2010; Dahlhaus & Eichler, 2003; Malinsky & Spirtes, 2018; Assaad et al., 2022) and remains largely underexplored (Pamfil et al., 2020; Gong et al., 2023).

Motivated by the above analysis, in this paper, we start from the tensor time series data as shown in Figure 1(a), in which the 3D structure contains higher dimensions than typical 2D multivariate time series data. For example, tensor time series can represent multivariate climate time series data (e.g., time plus temperature, wind, and atmospheric water content) with corresponding spatial information (e.g., longitude, latitude, and geocode). After that, we aim to build a comprehensive causality model for this tensor time series, which could not only capture non-linear and time-lagged causality (like the Granger model (Granger, 1969; Tank et al., 2022)) but also offset the ignored instantaneous causal effects at each timestamp, as shown in Figure 1(b). Our **ultimate goal** is to leverage the discovered comprehensive causality to understand the trend and latent distribution of the historical tensor time series and finally contribute to the analysis tasks like tensor time series forecasting and anomaly detection.

To this end, we first propose a comprehensive causal model named Time-Respecting Bayesian Network Augmented Neural Granger Causality, i.e., TBN Granger Causality. Theoretically, discovering TBN Granger Causality relies on a bi-level optimization. The inner optimization discovers a sequence of Bayesian Networks at each timestamp t respectively for representing the instantaneous causal effects among variables (i.e., which causality is responsible for the instantaneous feature generation). Then, the outer optimization realizes integrating time-respecting Bayesian Networks with time-lagged neural Granger causality in an autoregressive manner.

Second, to embed TBN Granger Causality into guiding the tensor time series analysis tasks like forecasting and anomaly detection, we propose an end-to-end deep generative model, called **T**ime-**A**ugmented **C**ausal Time **S**eries **A**nalysi**S** Model, i.e., TacSas. TacSas fits more real-world application scenarios (e.g., climate or transportation) by investigating how to capture good causal structures *without* the ground-truth structures guidance. Furthermore, TacSas is end-to-end, meaning that it can not only discover TBN Granger Causality from the observed time series but also seamlessly use the discovery to forecast future time series and detect possible anomalies.

To evaluate TacSas, we first use the synthetic benchmark, Lorenz-96 (Lorenz, 1996), to verify that TacSas can indeed discover ground-truth causal structures with high accuracy.

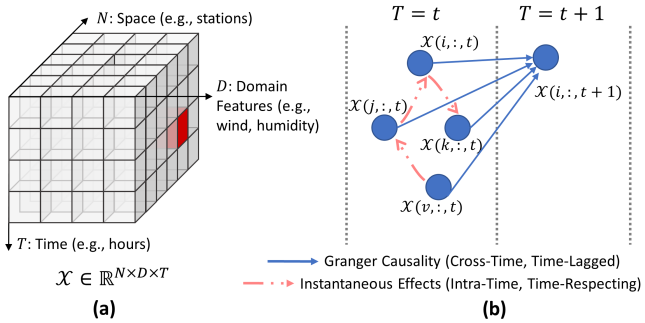


Figure 1: (a) Tensor Time-Series Data: The Red Cell Means the Possible Anomaly. (b) Visualization of (Neural) Granger Causality's Time-Lagged Property without Instantaneous Effects.

Then, we extend to the real-world setting and test if TacSas can utilize the discovered causality to conduct tensor time series forecasting and identify anomalies. We use four tensor time series datasets from the hourly climate benchmark database ERA5 (Hersbach et al., 2018) and align them with the extreme weather database of NOAA[2] based on geoinformation and contribute a new benchmark for climate science. The results show that TacSas outperforms both state-of-the-art forecasting and detection baselines.

2. Preliminary

Tensor Time Series. As shown in Figure 1(a), we have tensor time series data stored in $\mathcal{X} \in \mathbb{R}^{N \times D \times T}$. Note that a slice of $\mathcal{X}$, i.e., $\mathcal{X}(i,:,:) \in \mathbb{R}^{D \times T}$, $i \in \{1 \dots, N\}$, is typically denoted as the common multivariate time series data (Su et al., 2019; Zhao et al., 2020). In this way, tensor time series can be understood as multiple multivariate time series data. Such tensor time series data can usually be found in the real world. For example, in each element $\mathcal{X}(i,d,t)$ of the nationwide weather data $\mathcal{X}$, $i \in \{1 \dots, N\}$ can be the spatial locations (e.g., counties), $d \in \{1 \dots, D\}$ can be the weather features (e.g., temperature and humidity), and $t \in \{1 \dots, T\}$ can be the time dimension (e.g., hours). Throughout the paper, we use the calligraphic letter to denote a 3D tensor (e.g., $\mathcal{X}$) and the bold capital letter to denote a 2D matrix (e.g., $\boldsymbol{X}$).

Problem Definition. In this paper, we aim to discover and utilize comprehensive causal structures for tensor time-series analysis tasks, including forecasting and anomaly detection. To be more specific, given the tabular data $\mathcal{X} \in \mathbb{R}^{N \times D \times T}$ as shown in Figure 1, we aim to forecast the future data $\mathcal{X}' \in \mathbb{R}^{N \times D \times \tau}$, where τ is a forecasting window. Additionally, with the forecasted $\mathcal{X}'$, we also aim to detect if $\mathcal{X}'$ contains abnormal values.

[2] https://www.ncdc.noaa.gov/stormevents/ftp.jsp

3. TacSas: Discovering TBN Granger Causality via Generative Learning

In this section, we introduce how TacSas discovers TBN Granger Causality in the historical tensor time series and utilizes it to guide tensor time series forecasting and anomaly detection. The overall framework of TacSas is shown in Figure 2.

The upper component of Figure 2 represents the data preprocessing part (i.e., converting raw input $\mathcal{X}$ to latent representation $\mathcal{H}$) of TacSas through a pre-trained autoencoder. The goal of this component is leveraging comprehensive causality (e.g., TBN Granger Causality) to achieve seamless forecasting and anomaly detection. The theoretical reasoning and necessity are introduced in Sec.3.3, and the empirical validation is demonstrated in Appendix B.2.

The lower component of Figure 2 shows how TacSas discovers TBN Granger Causality in the historical tensor time series (in the form of $\mathcal{H}$ other than $\mathcal{X}$) and generates future tensor time series. In brief, the optimization of TacSas is bi-level. First, the inner optimization captures instantaneous effects among variables at each timestamp, respectively, which describes the inner-time feature generation. These causal structures are then stored in the form of a sequence of Bayesian Networks. The details are introduced in Sec.3.1. Second, the outer optimization discovers the Neural Granger Causality among variables in a time window with the support of a sequence of Bayesian Networks (i.e., TBN Granger Causality). After introducing details in Sec.3.2, we derive the formal equation of TBN Granger Causality, Eq. 7.

3.1. Inner Optimization of TacSas for Identifying Instantaneous Causal Relations in Time Series

Generally speaking, the inner optimization produces a sequence of Bayesian Networks for each observed timestamp. At time t, the instantaneous causality is discovered based on input features $\mathcal{H}(:,:,t) = \boldsymbol{H}^{(t)} \in \mathbb{R}^{N\times H}$, and is represented by a directed acyclic graph $\mathcal{G}^{(t)} = (\boldsymbol{A}^{(t)} \in \mathbb{R}^{N\times N}, \boldsymbol{H}^{(t)} \in \mathbb{R}^{N\times H})$. To be specific, $\boldsymbol{A}^{(t)}$ is a weighted adjacency matrix of the Bayesian Network at time t, and each cell represents the coefficient of causal effects between variables u and $v \in \{1,\dots,N\}$. The features (e.g., $\mathcal{H}(v,:,t)$) are transformed from the input raw features (e.g., $\mathcal{X}(v,:,t)$). The transformation is causality-agnostic but necessary for downstream time series analysis tasks, with details introduced in Sec.3.3.3.

The reasoning for discovering the instantaneous causal effects in the form of the Bayesian Network originates from a widely adopted assumption of causal graph learning (Zheng et al., 2018; Yu et al., 2019; Guo et al., 2021; Geffner et al., 2022; Gong et al., 2023): there exists a ground-truth causal graph $\mathbf{S}^{(t)}$ that specifies instantaneous parents of variables to recover their value generating process. Therefore, in our inner optimization, the goal is to discover the causal structure $\mathbf{S}^{(t)}$ at each time t by recovering the generation of input features $\boldsymbol{H}^{(t)}$. Specifically, given the observed $\boldsymbol{H}^{(t)}$, we aim to estimate a structure $\boldsymbol{A}^{(t)}$, through which a certain distribution $\boldsymbol{Z}^{(t)}$ could generate $\boldsymbol{H}^{(t)}$ for $t \in \{1,\dots,T\}$. In this way, the instantaneous causal effects are discovered, and the corresponding structures are encoded in $\boldsymbol{A}^{(t)}$. The generation function is expressed as follows.

$$\sum_t log\mathcal{P}(\boldsymbol{H}^{(t)}) = \sum_t log \int \mathcal{P}(\boldsymbol{H}^{(t)}|\boldsymbol{Z}^{(t)})\mathcal{P}(\boldsymbol{Z}^{(t)})d\boldsymbol{Z}^{(t)} \tag{1}$$

where the generation likelihood $\mathcal{P}(\boldsymbol{H}^{(t)}|\boldsymbol{Z}^{(t)})$ also takes $\boldsymbol{A}^{(t)}$ as input. The complete formula is shown in Eq. 3.

For Eq 1, on the one hand, it is hard to get the prior distribution $\mathcal{P}(\boldsymbol{Z}^{(t)})$, which is highly related to the distribution of ground-truth causal graph distribution $\mathcal{P}(\mathbf{S}^{(t)})$ at time t (Geffner et al., 2022). On the other hand, for the generation likelihood $\mathcal{P}(\boldsymbol{H}^{(t)}|\boldsymbol{Z}^{(t)})$, the actual posterior $\mathcal{P}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})$ is also intractable. Thus, we resort to the variational autoencoder (VAE) (Kingma & Welling, 2014). In this way, the actual posterior $\mathcal{P}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})$ can be replaced by the variational posterior $\mathcal{Q}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})$, and the prior distribution $\mathcal{P}(\boldsymbol{Z}^{(t)})$ is approximated by a Gaussian distribution. Furthermore, the inside encoder and decoder modules should take the structure $\boldsymbol{A}^{(t)}$ as the input. This design can be realized by various off-the-shelf variational graph autoencoders such as VGAE (Kipf & Welling, 2016), etc. However, the inner optimization is coupled with the outer optimization, i.e., the instantaneous causality will be integrated with cross-time Granger causality to make inferences. The inner complex neural architectures and parameters may render the outer optimization module hard to train, especially when the outer module itself needs to be complex. Therefore, we extend the widely-adopted linear Structural Equation Model (SEM) (Zheng et al., 2018; Yu et al., 2019; Geffner et al., 2022; Gong et al., 2023) to the time-respecting setting as follows.

For $\mathcal{Q}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})$, the encoder equation is expressed as

$$\boldsymbol{Z}^{(t)} = (\boldsymbol{I} - {\boldsymbol{A}^{(t)}}^{\top}) f_{\theta^{(t)}_{enc}}(\boldsymbol{H}^{(t)}) \tag{2}$$

For $\mathcal{P}(\boldsymbol{H}^{(t)}|\boldsymbol{Z}^{(t)})$, the decoder equation is expressed as

$$\boldsymbol{H}^{(t)} = f_{\theta^{(t)}_{dec}}((\boldsymbol{I} - {\boldsymbol{A}^{(t)}}^{\top})^{-1}\boldsymbol{Z}^{(t)}) \tag{3}$$

As analyzed above[3], $f_{\theta^{(t)}_{enc}}$ and $f_{\theta^{(t)}_{dec}}$ do not need complicated neural architectures. Therefore, we can use two-layer

[3]The complete forms of $\mathcal{Q}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})$ and $\mathcal{P}(\boldsymbol{H}^{(t)}|\boldsymbol{Z}^{(t)})$ are $\mathcal{Q}_{A^{(t)}}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})$ and $\mathcal{P}_{A^{(t)}}(\boldsymbol{H}^{(t)}|\boldsymbol{Z}^{(t)})$, we omit the subscript $A^{(t)}$ for brevity.

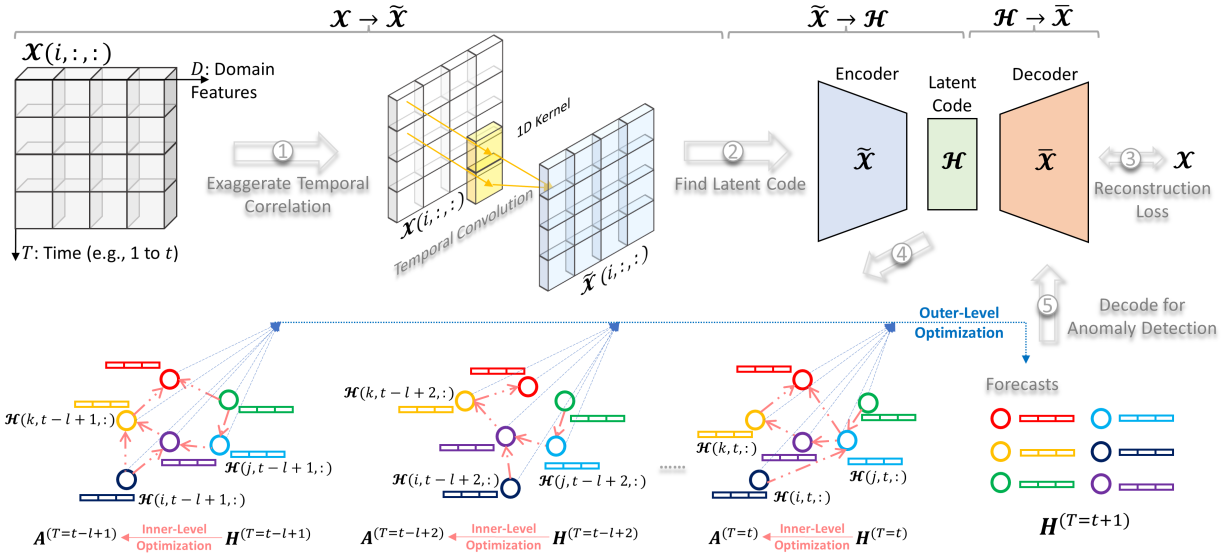


Figure 2: Working Flow of TacSas: Discovering and Utilizing the TBN Granger Causality through a Bi-Level Optimization for Tensor Time Series Forecasting and Anomaly Detection.

MLPs for them. Then, the objective function $\mathcal{L}_{DAG}^{(t)}$ for discovering the instantaneous causality at time t is expressed as follows, which corresponds to the inner optimization.

$$\begin{aligned}\min_{\theta_{enc}^{(t)},\theta_{dec}^{(t)},\boldsymbol{A}^{(t)}} \mathcal{L}_{DAG}^{(t)} = \; & D_{KL}(\mathcal{Q}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})\|\mathcal{P}(\boldsymbol{Z}^{(t)})) \\ & -\mathbb{E}_{\mathcal{Q}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})}[\log\mathcal{P}(\boldsymbol{H}^{(t)}|\boldsymbol{Z}^{(t)})] \\ \text{s.t.} \sum_t \text{Tr}[(\boldsymbol{I}+\boldsymbol{A}^{(t)}\circ\boldsymbol{A}^{(t)})^N] - N = 0, \; & \text{for } t \in \{1,\dots,T\}\end{aligned} \tag{4}$$

where the first term in $\mathcal{L}_{DAG}^{(t)}$ is the KL-divergence measuring the distance between the distribution of generated $\boldsymbol{Z}^{(t)}$ and the pre-defined Gaussian, and the second term is the reconstruction loss between the generated $\boldsymbol{Z}^{(t)}$ with the original input $\boldsymbol{H}^{(t)}$. Note that there is an important constraint, i.e., $\text{Tr}[(\boldsymbol{I}+\boldsymbol{A}^{(t)}\circ\boldsymbol{A}^{(t)})^N]-N=0$, on $\boldsymbol{A}^{(t)} \in \mathbb{R}^{N\times N}$. $\text{Tr}(\cdot)$ is the trace of a matrix, and $\circ$ denotes the Hadamard product. The meaning of the constraint is explained as follows. The constraint in Eq. 4, i.e., $\text{Tr}[(\boldsymbol{I}+\boldsymbol{A}^{(t)}\circ\boldsymbol{A}^{(t)})^N]-N=0$ regularizes the acyclicity of $\boldsymbol{A}^{(t)}$ during the optimization process, i.e., the learned $\boldsymbol{A}^{(t)}$ should not have any possible closed-loops at any length.

Lemma 3.1. *Let $\boldsymbol{A}^{(t)}$ be a weighted adjacency matrix (negative weights allowed). $\boldsymbol{A}^{(t)}$ has no N-length loops, if Tr$[(\boldsymbol{I}+\boldsymbol{A}^{(t)}\circ\boldsymbol{A}^{(t)})^N]-N=0$.*

The intuition is that there will be no k-length path from node u to node v on a binary adjacency matrix $\}(u,v)=0$. Compared with original acyclicity constraints in (Yu et al., 2019), our Lemma 3.1 gets rid of the λ condition. Then we can denote $\alpha(A^{(t)}) = \text{Tr}[(\boldsymbol{I}+\boldsymbol{A}^{(t)}\circ\boldsymbol{A}^{(t)})^N]-N$ and use Lagrangian optimization for Eq. 4 as follows.

$$\begin{aligned}\min_{\theta_{enc}^{(t)},\theta_{dec}^{(t)},\boldsymbol{A}^{(t)}} \mathcal{L}_{DAG}^{(t)} = \; & D_{KL}(\mathcal{Q}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})\|\mathcal{P}(\boldsymbol{Z}^{(t)})) \\ & -\mathbb{E}_{\mathcal{Q}(\boldsymbol{Z}^{(t)}|\boldsymbol{H}^{(t)})}[\log\mathcal{P}(\boldsymbol{H}^{(t)}|\boldsymbol{Z}^{(t)})] \\ & +\lambda\,\alpha(A^{(t)}) + \frac{c}{2}|\alpha(A^{(t)})|^2, \;\; \text{for } t \in \{1,\dots,T\}\end{aligned} \tag{5}$$

where λ and c are two hyperparameters, and larger λ and c enforce $\alpha(\boldsymbol{A}^{(t)})$ to be smaller.

Theorem 3.2. *If the ground-truth instantaneous causal graph $\mathbf{S}^{(t)}$ at time t generates the features of variables following the normal distribution, then the inner optimization (i.e., Eq. 4) can identify $\mathbf{S}^{(t)}$ under the standard causal discovery assumptions (Geffner et al., 2022).*

3.2. Outer Optimization of TacSas for Integrating Instantaneous Causality with Neural Granger

Given the inner optimization, Bayesian Networks can be obtained at each timestamp t, which means that multiple instantaneous causalities are discovered. Thus, in the outer optimization, we integrate these evolving Bayesian Networks into Granger Causality discovery. First, the classic Granger Causality (Granger, 1969) is discovered in the form of the variable-wise coefficients across different timestamps (i.e., a time window) through the autoregressive prediction process. The prediction based on the linear Granger Causality (Granger, 1969) is expressed as follows.

$$\boldsymbol{H}^{(t)} = \sum_{l=1}^{L} \boldsymbol{W}^{(l)}\boldsymbol{H}^{(t-l)} + \boldsymbol{e}^{(t)} \tag{6}$$

where $\boldsymbol{H}^{(t)} \in \mathbb{R}^{N\times D}$ denotes the features of N variables at time t, $\boldsymbol{e}^{(t)}$ is the noise, and L is the pre-defined time lag indicating how many past timestamps can affect the values of $\boldsymbol{H}^{(t)}$. Weight matrix $\boldsymbol{W}^{(l)} \in \mathbb{R}^{N\times N}$ stores the cross-time coefficients captured by Granger Causality, i.e., matrix $\boldsymbol{W}^{(l)}$ aligns the variables at time $t-l$ with the variables at time t. To compute those weights, several linear methods are proposed, e.g., vector autoregressive model (Arnold et al., 2007).

Facing non-linear causal relationships, neural Granger Causality discovery (Tank et al., 2022) is recently proposed to explore the nonlinear Granger Causality effects. The general principle is to represent causal weights $\boldsymbol{W}$ by deep neural networks. To integrate instantaneous effects with neural Granger Causality discovery, our TBN Granger Causality is expressed as follows.

$$\hat{H}(i,:)^{(t)} = f_{\Theta_i}[(\boldsymbol{A}^{(t-1)}, \boldsymbol{H}^{(t-1)}), \ldots, (\boldsymbol{A}^{(t-L)}, \boldsymbol{H}^{(t-L)})] \tag{7}$$

where L is the lag (or window size) in the Granger Causality, and i is the index of the i-th variable. f_{Θ_i} is a neural computation unit with all parameters denoted as Θ_i, whose input is an L-length time-ordered sequence of $(\boldsymbol{A}, \boldsymbol{H})$. And f_{Θ_i} is responsible for discovering the TBN Granger Causality for variable i at time t from all variables that occurred in the past time lag l. The choice of neural unit f_{Θ_i} is flexible, such as MLP and LSTM (Tank et al., 2022). Different neural unit choices correspond to different causality interpretations. In our proposed TacSas model, we use graph recurrent neural networks (Wu et al., 2021), and the causality interpretations are introduced in Sec 3.3.

In the outer optimization, to evaluate the prediction under the TBN Granger Causality, we use the mean absolute error (MAE) loss on the prediction and the ground truth, which is effective and widely applied to time-series forecasting tasks (Li et al., 2018; Shang et al., 2021).

$$\min_{\Theta_i, \boldsymbol{A}^{(t-1)}, \ldots, \boldsymbol{A}^{(t-l)}} \mathcal{L}_{pred} = \sum_i \sum_t |H(i,:)^{(t)} - \hat{H}(i,:)^{(t)}| \tag{8}$$

where $\Theta_i, \boldsymbol{A}^{(t-1)}, \ldots, \boldsymbol{A}^{(t-l)}$ are all the parameters for the prediction $\hat{H}(i,:)^{(t)}$ of variable i at time t. The composition and update rules are expressed below.

For updating f_{Θ_i}, we employ the recurrent neural structure to fit the input sequence. Moreover, the sequential inputs also contain the structured data $\boldsymbol{A}$. Therefore, we use the graph recurrent neural architecture (Li et al., 2018) because it is designed for directed graphs, whose core is a gated recurrent unit (Chung et al., 2014).

$$\begin{aligned}
\boldsymbol{R}^{(t)} &= \text{sigmoid}(\boldsymbol{W}_{\boldsymbol{R}*\boldsymbol{A}^{(t)}}[\boldsymbol{H}^{(t)} \oplus \boldsymbol{S}^{(t-1)}] + \boldsymbol{b}_R) \\
\boldsymbol{C}^{(t)} &= \tanh(\boldsymbol{W}_{\boldsymbol{C}*\boldsymbol{A}^{(t)}}[\boldsymbol{H}^{(t)} \oplus (\boldsymbol{R}^{(t)} \odot \boldsymbol{S}^{(t-1)})] + \boldsymbol{b}_C) \\
\boldsymbol{U}^{(t)} &= \text{sigmoid}(\boldsymbol{W}_{\boldsymbol{U}*\boldsymbol{A}^{(t)}}[\boldsymbol{H}^{(t)} \oplus \boldsymbol{S}^{(t-1)}] + \boldsymbol{b}_U) \\
\boldsymbol{S}^{(t)} &= \boldsymbol{U}^{(t)} \odot \boldsymbol{S}^{(t-1)} + (\boldsymbol{I} - \boldsymbol{U}^{(t)}) \odot \boldsymbol{C}^{(t)}
\end{aligned} \tag{9}$$

where $\boldsymbol{R}^{(t)}$, $\boldsymbol{C}^{(t)}$, and $\boldsymbol{U}^{(t)}$ are three parameterized gates, with corresponding weights $\boldsymbol{W}$ and bias $\boldsymbol{b}$. $\boldsymbol{H}^{(t)}$ is the input, and $\boldsymbol{S}^{(t)}$ is the hidden state. Gates $\boldsymbol{R}^{(t)}$, $\boldsymbol{C}^{(t)}$, and $\boldsymbol{U}^{(t)}$ share the similar structures. For example, in $\boldsymbol{R}^{(t)}$, the graph convolution operation for computing the weight $\boldsymbol{W}_{\boldsymbol{R}*\boldsymbol{A}^{(t)}}$ is defined as follows, and the same computation applies to gates $\boldsymbol{U}^{(t)}$ and $\boldsymbol{C}^{(t)}$.

$$\boldsymbol{W}_{\boldsymbol{R}*\boldsymbol{A}^{(t)}} = \sum_{k=0}^{K} \theta_{k,1}^R ({\boldsymbol{D}_{out}^{(t)}}^{-1} \boldsymbol{A}^{(t)})^k + \theta_{k,2}^R ({\boldsymbol{D}_{in}^{(t)}}^{-1} {\boldsymbol{A}^{(t)}}^\top)^k \tag{10}$$

where $\theta_{k,1}^R$, $\theta_{k,2}^R$ are learnable weight parameters; scalar k is the order for the stochastic diffusion operation (i.e., similar to steps of random walks); ${\boldsymbol{D}_{out}^{(t)}}^{-1} \boldsymbol{A}^{(t)}$ and ${\boldsymbol{D}_{in}^{(t)}}^{-1} {\boldsymbol{A}^{(t)}}^\top$ serve as the transition matrices with the in-degree matrix $\boldsymbol{D}_{in}^{(t)}$ and the out-degree matrix $\boldsymbol{D}_{out}^{(t)}$; -1 and $\top$ are inverse and transpose operations.

For updating each of $\{\boldsymbol{A}^{(t-1)}, \ldots, \boldsymbol{A}^{(t-l)}\}$, we take $\boldsymbol{A}^{(t-l)}$ as an example to illustrate. The optimal $\boldsymbol{A}^{(t-l)}$ stays in the space of $\{0,1\}^{N\times N}$. To be specific, each edge $A^{(t-l)}(i,j)$ can be parameterized as $\theta_{i,j}^{(t-l)}$ following the Bernoulli distribution. However, $N^2 l$ is hard to scale, and the discrete variables are not differentiable. Therefore, we adopt the Gumbel reparameterization from (Jang et al., 2017; Maddison et al., 2017). It provides a continuous approximation for the discrete distribution, which has been widely used in the graph structure learning (Kipf et al., 2018; Shang et al., 2021). The general reparameterization form can be written as $A^{(t-l)}(i,j) = softmax(FC((H(i,:)^{(t-l)} || H(j,:)^{(t-l)}) + g)/\xi)$, where FC is a feedforward neural network, g is a scalar drawn from a Gumbel$(0,1)$ distribution, and ξ is a scaling hyperparameter. Different from (Kipf et al., 2018; Shang et al., 2021), in our setting, the initial structure input is constrained by the causality discovery, which originates from the inner optimization step. Hence, the structure learning in the outer optimization takes the adjacency matrix from the inner optimization as the initial input, which is

$$A_{outer}^{(t-l)}(i,j) = softmax(A_{inner}^{(t-l)}(i,j) + \boldsymbol{g})/\xi) \tag{11}$$

where $A_{inner}^{(t)}(i,j)$ is the structure learned by our inner optimization through Eq. 4, $A_{outer}^{(t)}(i,j)$ is the updated structure, and $\boldsymbol{g}$ is a vector of i.i.d samples drawn from

a Gumbel$(0, 1)$ distribution. In outer optimization, Eq. 8 fine-tunes the evolving Bayesian Networks to make the intra-time causality fit the cross-time causality well. Note that, the outer optimization w.r.t. $\boldsymbol{A}^{(t)}$ may break the acyclicity, and another round of inner optimization may be necessary.

3.3. Deployment of TacSas for Time Series Forecasting and Anomaly Detection

In this section, we introduce how TacSas achieves tensor time series forecasting and anomaly detection in threefold: data preprocessing, neural architecture selection, and training procedure.

First (data preprocessing), in addition to forecasting, TacSas is also for anomaly detection. Thus, we design the hidden feature $\mathcal{H}$ extraction in TacSas motivated by the Extreme Value Theory (Beirlant et al., 2004) or so-called Extreme Value Distribution in stream (Siffer et al., 2017).

Remark 3.3. According to the Extreme Value Distribution (Fisher & Tippett, 1928), under the limiting forms of frequency distributions, extreme values have the same kind of distribution, regardless of original distributions.

An example (Siffer et al., 2017) can help interpret and understand the Extreme Value Distribution theory. Maximum temperatures or tide heights have more or less the same distribution even though the distributions of temperatures and tide heights are not likely to be the same. As rare events have a lower probability, there are only a few possible shapes for a general distribution to fit. Inspired by this observation, we can design a simple but effective module in TacSas to achieve anomaly detection, i.e., a pre-trained autoencoder model that tries to explore the distribution of normal features in $\mathcal{X}$ as shown in Figure 2. As long as this autoencoder model can capture the latent distribution for normal events, then the generation probability of a piece of time series data can be utilized as the condition for detecting anomaly patterns. This is because the extreme values are identified with a remarkably low generation probability. To be specific, after the forecast $\boldsymbol{H}^{(t)}$ is output, the generation probability of $\boldsymbol{H}^{(t)}$ into $\boldsymbol{X}^{(t)}$ through the pre-trained autoencoder can be used to detect the anomalies at t.

Second (neural architecture selection), we encode f_{Θ_i} into a sequence-to-sequence model (Sutskever et al., 2014). That is, given a time window (or time lag), TacSas could forecast the corresponding features for the next time window. Moreover, with $\boldsymbol{W}^{(l)}$ in Eq. 6 and f_{Θ_i} in Eq. 7, we can observe that the classical linear Granger Causality $\boldsymbol{W}^{(l)}$ can be discovered for each time lag. In other words, each time lag has its own discovered coefficients, but f_{Θ_i} is shared by all time lags. This sharing manner is designed for scalability and is called Summary Causal Graph (Marcinkevics & Vogt, 2021; Assaad et al., 2022). The underlying intuition is that the causal effects mainly depend on the near timestamps. Further, for the neural Granger Causality interpretation in f_{Θ_i}, we follow the rule (Tank et al., 2022) that if the j-th row of ($\boldsymbol{W}_{\boldsymbol{R}*\boldsymbol{A}^{(t)}}$, $\boldsymbol{W}_{\boldsymbol{C}*\boldsymbol{A}^{(t)}}$, and $\boldsymbol{W}_{\boldsymbol{U}*\boldsymbol{A}^{(t)}}$) are zeros, then variable j is not the Granger-cause for variable i in this time window.

Third (training procedure), as shown in Figure 2, the autoencoder can be pre-trained with reconstruction loss (e.g., MSE) ahead of the inner and outer optimization, to obtain $\mathcal{H}$ for the feature latent distribution representation. By utilizing all input $\mathcal{H}$, the inner optimization learns the sequential Bayesian Networks, and the outer optimization aligns Bayesian Networks with the neural Granger Causality to produce all the forecast $\mathcal{H}'$. The inner and outer optimization can be trained interchangeably.

4. Experiments

The ground-truth causality discovery experiments in the synthetic benchmark, Lorenz 96 System (Lorenz, 1996), are shown in Appendix B.1, where our TacSas can capture the true causality with the competitive high accuracy. Then, in this section, we test TacSas on utilizing its discovery for time series forecasting and anomaly detection.

4.1. Experiment Setup

Datasets. Our forecasting data (i.e., hourly tensor time series data) originates from climate domain benchmark ERA5 (Hersbach et al., 2018)[4]. To be specific, we select four datasets covering 45 weather features (i.e., wind gusts, rain, etc.) from 238[5] counties in the United States of America during 2017–2020. Moreover, we choose thunderstorms as the anomaly pattern to be detected after forecasting. The thunderstorm record is identified in NOAA database[6] hourly and nationwide, i.e., 1 means a thunderstorm happens in the corresponding hour at a certain location, and 0 means no thunderstorm happens. We processed the geocode to align weather features in ERA5 with anomaly patterns in NOAA. The geographic distribution and anomaly pattern frequency distribution are shown in Appendix D.

Baselines. Besides the causality discovery baseline in Appendix B, the **first** category is for tensor time series forecasting: (1) GRU (Chung et al., 2014) is a classical sequence to sequence generation model. (2) DCRNN (Li

[4] https://cds.climate.copernicus.eu/cdsapp#!/home

[5] 100 of 238 counties are top-ranked counties for the thunderstorm (anomaly label) frequency, and the rest are randomly selected.

[6] https://www.ncdc.noaa.gov/stormevents/ftp.jsp

Table 1: Forecasting Error (MAE, 10^{-2})

	ERA5-2017 (↓)	ERA5-2018 (↓)	ERA5-2019 (↓)	ERA5-2020 (↓)
GRU	1.8834 ± 0.0126	1.9764 ± 0.1466	1.6194 ± 0.2645	1.7859 ± 0.2324
DCRNN	0.0819 ± 0.0025	0.0797 ± 0.0049	0.0799 ± 0.0035	0.0826 ± 0.0033
GTS	0.0777 ± 0.0054	0.0766 ± 0.0029	0.0760 ± 0.0031	0.0742 ± 0.0021
TacSas	0.0496 ± 0.0017	0.0499 ± 0.0017	0.0502 ± 0.0016	0.0488 ± 0.0019
ST-SSL	0.0345 ± 0.0051	0.0330 ± 0.0018	0.0361 ± 0.0021	0.0348 ± 0.0020
TacSas++	$\mathbf{0.0271 \pm 0.0004}$	$\mathbf{0.0276 \pm 0.0004}$	$\mathbf{0.0282 \pm 0.0003}$	$\mathbf{0.0265 \pm 0.0004}$

et al., 2018) is a graph convolutional recurrent neural network, of which the input graph structure is given, not causal, and static (i.e., shared by all timestamps). In this viewpoint, we let each node randomly distribute its unit weights to others. (3) GTS (Shang et al., 2021) is also a graph convolutional recurrent neural network that does not need the input graph but learns the structure based on the node features, but the learned structure is also shared by all timestamps and is not causal. To compare the performance of DCGNN (Li et al., 2018) and GTS (Shang et al., 2021) with TacSas, causality is the control variable since we make all the rest (e.g., neural network type, number of layers, etc.) identical for them. The **second** category is for anomaly detection on tensor time series: (1) DeepSAD (Ruff et al., 2020), (2) DeepSVDD (Ruff et al., 2018), and (3) DROCC (Goyal et al., 2020). Since these three have no forecast abilities, we let them use the ground-truth observations, and our TacSas utilizes the forecast features during anomaly detection experiments. Also, these three baselines are designed for multi-variate time-series data, not tensor time-series. Thus, we flatten our tensor time series along the spatial dimension and report the average performance for these three baselines over all locations.

Next, we introduce forecasting and anomaly detection performance. Details about split and hyperparameters are in Appendix C. More ablation studies can be found in Appendix B.3.

4.2. Forecasting Performance

In Table 1, we present the forecasting performance in terms of mean absolute error (MAE) on the testing data of three algorithms, namely DCGNN (Li et al., 2018), GTS (Shang et al., 2021), ST-SSL (Ji et al., 2023), our TacSas, and TacSas++ (i.e., TacSas with persistence forecast constraints). Here, we set the time window as 24, meaning that we use the past 24 hours tensor time series to forecast the future 24 hours in an autoregressive manner. Moreover, for baselines and TacSas, we set f_{Θ_i} in Eq.7 shared by all weather variables to ensure the scalability, such that we do not need to train N recurrent graph neural networks for a single prediction. In Table 1, we can observe a general pattern that our TacSas outperforms the baselines with GTS performing better than DCGNN. For example, with 2017 as the testing data, our TacSas performs 39.44% and 36.16% better than DCRNN and GTS. An explanation is that the temporally fine-grained causal relationships can contribute more to the forecasting accuracy than non-causal directed graphs, since DCGNN, GTS, and our TacSas all share the graph recurrent manner. TacSas however, discovers causalities at different timestamps, while DCGNN and GTS use feature similarity based connections. Moreover, ST-SSL achieves competitive forecasting performance via contrastive learning on time series. Motivated by contrastive manner, TacSas++ is proposed by persistence forecast constraints. That is, the current forecast of TacSas is further calibrated by its nearest time window (i.e., the last 24 hours in our setting). The detailed implementation is provided in Appendix C.

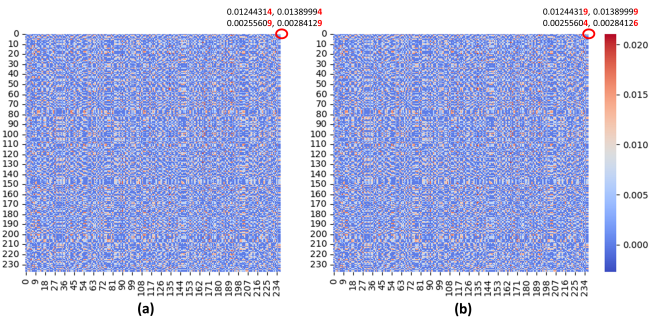


Figure 3: Time-Respecting Bayesian Networks of at the Same Hour of Two Consecutive Days.

To evaluate our explanation, we visualize causal connections at different times in Figure 3. Specifically, we show the Bayesian Network of 238 counties at the same hour on two consecutive days in the training data (i.e., May 1st and May 2nd, 2018). Interestingly, we can observe that two patterns in Figure 3 are almost identical at first glance. That could be the reason why DCRNN and GTS can perform well using the static structure. However, upon closer inspection, we find that these two are quite different to some extent if we zoom in, such as, in the upper right corner. Although the values have a tiny divergence, their volume is quite large. In two matrices of Figure 3, the number of different cells is 28,509, and the corresponding percentage is $\frac{28509}{238\times238} \approx 0.5033$. We suppose that discovering those value-tiny but volume-big differences makes TacSas outperform, to a large extent.

Table 2: Anomaly Detection Performance (AUC-ROC)

	NOAA-2017 (↑)	NOAA-2018 (↑)	NOAA-2019 (↑)	NOAA-2020 (↑)
DeepSAD	0.5305 ± 0.0481	0.5267 ± 0.0406	0.5563 ± 0.0460	0.6420 ± 0.0054
DeepSVDD	0.5201 ± 0.0045	0.5603 ± 0.0111	$\mathbf{0.6784 \pm 0.0112}$	0.5820 ± 0.0205
DROCC	0.5319 ± 0.0661	0.5103 ± 0.0147	0.6236 ± 0.0992	0.5630 ± 0.1082
TacSas	$\mathbf{0.5556 \pm 0.0010}$	$\mathbf{0.5685 \pm 0.0011}$	0.6298 ± 0.0184	$\mathbf{0.6745 \pm 0.0185}$

4.3. Anomaly Detection

After forecasting, we can have the hourly forecast of weather features at certain locations, denoted as $\mathcal{X}'$. Then, we use the encoder-decoder model in Figure 2 to calculate the feature-wise generation probability using mean squared error (MSE) between $\mathcal{X}'$ and its generation $\bar{\mathcal{X}}'$. Thus, we can calculate the average of feature-wise generation probability as the condition of anomalies to identify if an anomaly weather pattern (e.g., a thunderstorm) happens in an hour in a particular location. In Table 2, we use the Area Under the ROC Curve (i.e., AUC-ROC) as the metric, repeat the experiments four times, and report the performance of TacSas with baselines.

From Table 2, we can observe that the detection module of TacSas achieves very competitive performance. An explanation is that, based on the anomalies distribution shown in Table 3, it can be observed that the anomalies are very rare events. Our generative manner could deal with the very rare scenario by learning the feature latent distributions instead of the (semi-)supervised learning manner. For example, the maximum frequency of occurrences of thunderstorms is 770 (i.e., Jun 2017), which is collected from 238 counties over $30 \times 24 = 720$ hours, and the corresponding percentage is $\frac{770}{238 \times 30 \times 24} \approx 0.45\%$. Recall Remark 3.3, facing such rare events, we possibly find a single distribution to fit various anomaly patterns.

5. Related Work

Noteworthy applications of graph learning techniques in time series forecasting span in climate domains, including but not limited to heatwave prediction (Li et al., 2023a), and frost forecasts (Lira et al., 2022). To improve the the time series analysis effectiveness, there has been a growing focus on structured learning in the context of tabular time series data (Li et al., 2018; Wu et al., 2020; Zhao et al., 2020; Cao et al., 2020; Shang et al., 2021; Deng & Hooi, 2021; Marcinkevics & Vogt, 2021; Geffner et al., 2022; Tank et al., 2022; Spadon et al., 2022; Gong et al., 2023), which learned structures contribute to various time series analysis tasks like forecasting, anomaly detection, imputation, etc. As a directed and interpretable structure, causal graphs attract much research attention in this research topic (Guo et al., 2021). Granger Causality is a classic tool for discovering the cross-time variable causality in time series (Granger, 1969; Arnold et al., 2007). Facing complex patterns in time series data, different upgraded Granger Causality discovery methods emerge in different directions. Also, neural Granger Causality tools are recently proposed (Tank et al., 2022; Nauta et al., 2019; Khanna & Tan, 2020; Marcinkevics & Vogt, 2021; Xu et al., 2019; Huang et al., 2020), which utilizes the deep neural network to discover the nonlinear Granger causal coefficients and serve for the time-series forecasting tasks better. For example, in (Tank et al., 2022), authors introduce how to use multi-layer perception (MLPs) and long short-term memory (LSTMs) to realize the Neural Granger Causality for the forecasting task and how to interpret the Granger causal coefficients from neurons in deep networks. However, Granger Causality or Neural Granger Causality focuses on cross-time variable causality discovery and overlooks the instantaneous (or intra-time) variable causality. Also, how to utilize the discovered comprehensive causality to contribute to the downstream time series analysis tasks is under-explored mainly, especially in a setting where the ground-truth causal structures are hardly available for evaluation.

6. Conclusion

In this paper, we first propose TBN Granger Causality to align the instantaneous causal effects with time-lagged Granger causality. Moreover, we design TacSas to use TBN Granger Causality on time series analysis tasks like forecasting and anomaly detection in the real-world tensor time-series data and perform extensive experiments, where the results show the effectiveness of TacSas.

Acknowledgement

This work is supported by National Science Foundation under Award No. IIS-2117902, MIT-IBM Watson AI Lab, and IBM-Illinois Discovery Accelerator Institute - a new model of an academic-industry partnership designed to increase access to technology education and skill development to spur breakthroughs in emerging areas of technology. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

Arnold, A., Liu, Y., and Abe, N. Temporal causal modeling with graphical granger methods. In Berkhin, P., Caruana, R., and Wu, X. (eds.), *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, pp. 66–75. ACM, 2007. doi: 10.1145/1281192.1281203. URL https://doi.org/10.1145/1281192.1281203.

Assaad, C. K., Devijver, E., and Gaussier, É. Discovery of extended summary graphs in time series. In Cussens, J. and Zhang, K. (eds.), *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands*, volume 180 of *Proceedings of Machine Learning Research*, pp. 96–106. PMLR, 2022. URL https://proceedings.mlr.press/v180/assaad22a.html.

Beirlant, J., Goegebeur, Y., Segers, J., and Teugels, J. L. *Statistics of extremes: theory and applications*, volume 558. John Wiley & Sons, 2004.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. Spectral temporal graph neural network for multivariate time-series forecasting. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/cdf6581cb7aca4b7e19ef136c6e601a5-Abstract.html.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL http://arxiv.org/abs/1412.3555.

Dahlhaus, R. and Eichler, M. Causality and graphical models in time series analysis. *Oxford Statistical Science Series*, pp. 115–137, 2003.

Deng, A. and Hooi, B. Graph neural network-based anomaly detection in multivariate time series. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 4027–4035. AAAI Press, 2021. doi: 10.1609/aaai.v35i5.16523. URL https://doi.org/10.1609/aaai.v35i5.16523.

Fisher, R. A. and Tippett, L. H. C. Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Mathematical proceedings of the Cambridge philosophical society*, volume 24, pp. 180–190. Cambridge University Press, 1928.

Fu, D. and He, J. Natural and artificial dynamics in graphs: Concept, progress, and future. *Frontiers Big Data*, 5, 2022. doi: 10.3389/FDATA.2022.1062637. URL https://doi.org/10.3389/fdata.2022.1062637.

Fu, D., Xu, Z., Tong, H., and He, J. Natural and artificial dynamics in gnns: A tutorial. In Chua, T., Lauw, H. W., Si, L., Terzi, E., and Tsaparas, P. (eds.), *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*, pp. 1252–1255. ACM, 2023. doi: 10.1145/3539597.3572726. URL https://doi.org/10.1145/3539597.3572726.

Fu, D., Hua, Z., Xie, Y., Fang, J., Zhang, S., Sancak, K., Wu, H., Malevich, A., He, J., and Long, B. Vcr-graphormer: A mini-batch graph transformer via virtual connections. *CoRR*, abs/2403.16030, 2024. doi: 10.48550/ARXIV.2403.16030. URL https://doi.org/10.48550/arXiv.2403.16030.

Geffner, T., Antorán, J., Foster, A., Gong, W., Ma, C., Kiciman, E., Sharma, A., Lamb, A., Kukla, M., Pawlowski, N., Allamanis, M., and Zhang, C. Deep end-to-end causal inference. *CoRR*, abs/2202.02195, 2022. URL https://arxiv.org/abs/2202.02195.

Gong, W., Jennings, J., Zhang, C., and Pawlowski, N. Rhino: Deep causal temporal relationship learning with history-dependent noise. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=i_1rbq8yFWC.

Goyal, S., Raghunathan, A., Jain, M., Simhadri, H. V., and Jain, P. DROCC: deep robust one-class classification. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3711–3721. PMLR, 2020. URL http://proceedings.mlr.press/v119/goyal20c.html.

Granger, C. W. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.

Guo, R., Cheng, L., Li, J., Hahn, P. R., and Liu, H. A survey of learning causality with data: Problems and methods. *ACM Comput. Surv.*, 53(4):75:1–75:37, 2021.

doi: 10.1145/3397269. URL https://doi.org/10.1145/3397269.

Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., et al. Era5 hourly data on single levels from 1979 to present. *Copernicus climate change service (c3s) climate data store (cds)*, 10(10.24381), 2018.

Huang, H., Xu, C., Yoo, S., Yan, W., Wang, T., and Xue, F. Imbalanced time series classification for flight data analyzing with nonlinear granger causality learning. In d'Aquin, M., Dietze, S., Hauff, C., Curry, E., and Cudré-Mauroux, P. (eds.), *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pp. 2533–2540. ACM, 2020. doi: 10.1145/3340531.3412710. URL https://doi.org/10.1145/3340531.3412710.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Ji, J., Wang, J., Huang, C., Wu, J., Xu, B., Wu, Z., Zhang, J., and Zheng, Y. Spatio-temporal self-supervised learning for traffic flow prediction. In *AAAI 2023*, 2023.

Khanna, S. and Tan, V. Y. F. Economy statistical recurrent units for inferring nonlinear granger causality. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SyxV9ANFDH.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016. URL http://arxiv.org/abs/1611.07308.

Kipf, T. N., Fetaya, E., Wang, K., Welling, M., and Zemel, R. S. Neural relational inference for interacting systems. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2693–2702. PMLR, 2018. URL http://proceedings.mlr.press/v80/kipf18a.html.

Kofinas, M., Nagaraja, N. S., and Gavves, E. Roto-translated local coordinate frames for interacting dynamical systems. In *NeurIPS 2021*, 2021.

Kofinas, M., Bekkers, E. J., Nagaraja, N. S., and Gavves, E. Latent field discovery in interacting dynamical systems with neural fields. *CoRR*, abs/2310.20679, 2023. doi: 10.48550/ARXIV.2310.20679. URL https://doi.org/10.48550/arXiv.2310.20679.

Li, P., Yu, Y., Huang, D., Wang, Z.-H., and Sharma, A. Regional heatwave prediction using graph neural network and weather station data. *Geophysical Research Letters*, 50(7):e2023GL103405, 2023a.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

Li, Z., Fu, D., and He, J. Everything evolves in personalized pagerank. In Ding, Y., Tang, J., Sequeda, J. F., Aroyo, L., Castillo, C., and Houben, G. (eds.), *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pp. 3342–3352. ACM, 2023b. doi: 10.1145/3543507.3583474. URL https://doi.org/10.1145/3543507.3583474.

Lira, H., Martí, L., and Sanchez-Pi, N. A graph neural network with spatio-temporal attention for multi-sources time series data: An application to frost forecast. *Sensors*, 22(4):1486, 2022.

Lorenz, E. N. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1. Reading, 1996.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Malinsky, D. and Spirtes, P. Causal structure learning from multivariate time series in settings with unmeasured confounding. In Le, T. D., Zhang, K., Kiciman, E., Hyvärinen, A., and Liu, L. (eds.), *Proceedings of 2018 ACM SIGKDD Workshop on Causal Discovery, CD@KDD 2018, London, UK, 20 August 2018*, volume 92 of *Proceedings of Machine Learning Research*, pp. 23–47. PMLR, 2018.

URL http://proceedings.mlr.press/v92/malinsky18a.html.

Marcinkevics, R. and Vogt, J. E. Interpretable models for granger causality using self-explaining neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=DEa4JdMWRHp.

Moneta, A., Entner, D., Hoyer, P. O., and Coad, A. Causal inference by independent component analysis: Theory and applications. *Oxford Bulletin of Economics and Statistics*, 75(5):705–730, 2013.

Nauta, M., Bucur, D., and Seifert, C. Causal discovery with attention-based convolutional neural networks. *Mach. Learn. Knowl. Extr.*, 1(1):312–340, 2019. doi: 10.3390/make1010019. URL https://doi.org/10.3390/make1010019.

Pamfil, R., Sriwattanaworachai, N., Desai, S., Pilgerstorfer, P., Georgatzis, K., Beaumont, P., and Aragam, B. DYNOTEARS: structure learning from time-series data. In Chiappa, S. and Calandra, R. (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1595–1605. PMLR, 2020. URL http://proceedings.mlr.press/v108/pamfil20a.html.

Ruff, L., Görnitz, N., Deecke, L., Siddiqui, S. A., Vandermeulen, R. A., Binder, A., Müller, E., and Kloft, M. Deep one-class classification. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4390–4399. PMLR, 2018. URL http://proceedings.mlr.press/v80/ruff18a.html.

Ruff, L., Vandermeulen, R. A., Görnitz, N., Binder, A., Müller, E., Müller, K., and Kloft, M. Deep semi-supervised anomaly detection. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HkgH0TEYwH.

Runge, J., Nowack, P., Kretschmer, M., Flaxman, S., and Sejdinovic, D. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances*, 5(11):eaau4996, 2019.

Shang, C., Chen, J., and Bi, J. Discrete graph structure learning for forecasting multiple time series. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=WEHSlH5mOk.

Siffer, A., Fouque, P., Termier, A., and Largouët, C. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pp. 1067–1075. ACM, 2017. doi: 10.1145/3097983.3098144. URL https://doi.org/10.1145/3097983.3098144.

Spadon, G., Hong, S., Brandoli, B., Matwin, S., Jr., J. F. R., and Sun, J. Pay attention to evolution: Time series forecasting with deep graph-evolution learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5368–5384, 2022. doi: 10.1109/TPAMI.2021.3076155. URL https://doi.org/10.1109/TPAMI.2021.3076155.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., and Karypis, G. (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 2828–2837. ACM, 2019. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 3104–3112, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

Tank, A., Covert, I., Foti, N. J., Shojaie, A., and Fox, E. B. Neural granger causality. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):4267–4279, 2022. doi: 10.1109/TPAMI.2021.3065601. URL https://doi.org/10.1109/TPAMI.2021.3065601.

Wild, B., Eichler, M., Friederich, H.-C., Hartmann, M., Zipfel, S., and Herzog, W. A graphical vector autoregressive modelling approach to the analysis of electronic diary data. *BMC medical research methodology*, 10:1–13, 2010.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. Connecting the dots: Multivariate time se-

ries forecasting with graph neural networks. In Gupta, R., Liu, Y., Tang, J., and Prakash, B. A. (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 753–763. ACM, 2020. doi: 10.1145/3394486.3403118. URL https://doi.org/10.1145/3394486.3403118.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386. URL https://doi.org/10.1109/TNNLS.2020.2978386.

Xu, C., Huang, H., and Yoo, S. Scalable causal graph learning through a deep neural network. In Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E. A., Carmel, D., He, Q., and Yu, J. X. (eds.), *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pp. 1853–1862. ACM, 2019. doi: 10.1145/3357384.3357864. URL https://doi.org/10.1145/3357384.3357864.

Yu, Y., Chen, J., Gao, T., and Yu, M. DAG-GNN: DAG structure learning with graph neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7154–7163. PMLR, 2019. URL http://proceedings.mlr.press/v97/yu19a.html.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. Multivariate time-series anomaly detection via graph attention network. In Plant, C., Wang, H., Cuzzocrea, A., Zaniolo, C., and Wu, X. (eds.), *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*, pp. 841–850. IEEE, 2020. doi: 10.1109/ICDM50108.2020.00093. URL https://doi.org/10.1109/ICDM50108.2020.00093.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. Dags with NO TEARS: continuous optimization for structure learning. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 9492–9503, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e347c51419ffb23ca3fd5050202f9c3d-Abstract.html.

Zhou, D., Zheng, L., Fu, D., Han, J., and He, J. Mentorgnn: Deriving curriculum for pre-training gnns. In Hasan, M. A. and Xiong, L. (eds.), *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pp. 2721–2731. ACM, 2022. doi: 10.1145/3511808.3557393. URL https://doi.org/10.1145/3511808.3557393.

A. Theoretical Analysis

A.1. Proof of Lemma 3.1

Following (Yu et al., 2019), at each time t, we can extend $(\boldsymbol{I} + \boldsymbol{A}^{(t)} \circ \boldsymbol{A}^{(t)})^N$ by binomial expansion as follows.

$$(\boldsymbol{I} + \boldsymbol{A}^{(t)} \circ \boldsymbol{A}^{(t)})^N = \boldsymbol{I} + \sum_{k=1}^{N} \binom{N}{k} (\boldsymbol{A}^{(t)})^k \tag{12}$$

Since

$$\boldsymbol{I} \in \mathbb{R}^{N \times N} \tag{13}$$

then

$$\text{Tr}(\boldsymbol{I}) = N \tag{14}$$

Thus, if

$$(\boldsymbol{I} + \boldsymbol{A}^{(t)} \circ \boldsymbol{A}^{(t)})^N - N = 0 \tag{15}$$

then

$$(\boldsymbol{A}^{(t)})^k = 0, \text{ for any } k \tag{16}$$

Therefore, $\boldsymbol{A}^{(t)}$ is acyclic, i.e., no closed-loop exists in $\boldsymbol{A}^{(t)}$ at any possible length. Overall, the general idea of Lemma 3.1 is to ensure that the diagonal entries of the powered adjacency matrix have no 1s. There are also other forms for acyclicity constraints obeying the same idea but in different expressions, like exponential power form in (Zheng et al., 2018).

A.2. Sketch Proof of Theorem 3.2

According to Theorem 1 from (Geffner et al., 2022), the ELBO form as our Eq. 4 could identity the ground-truth causal structure $\mathbf{S}^{(t)}$ at each time t. The difference between our ELBO and the ELBO in (Geffner et al., 2022) is entries in the KL-divergence. Specifically, in (Geffner et al., 2022), the prior and variational posterior distributions are on the graph level. Usually, the prior distribution of graph structures is not easy to obtain (e.g., the non-IID and heterophyllous properties). Then, we transfer the graph structure distribution to the feature distribution that the Gaussian distribution can model. That's why our prior and variational posterior distributions in the KL-divergence are on the feature (generated by the graph) level.

B. Empirical Analysis

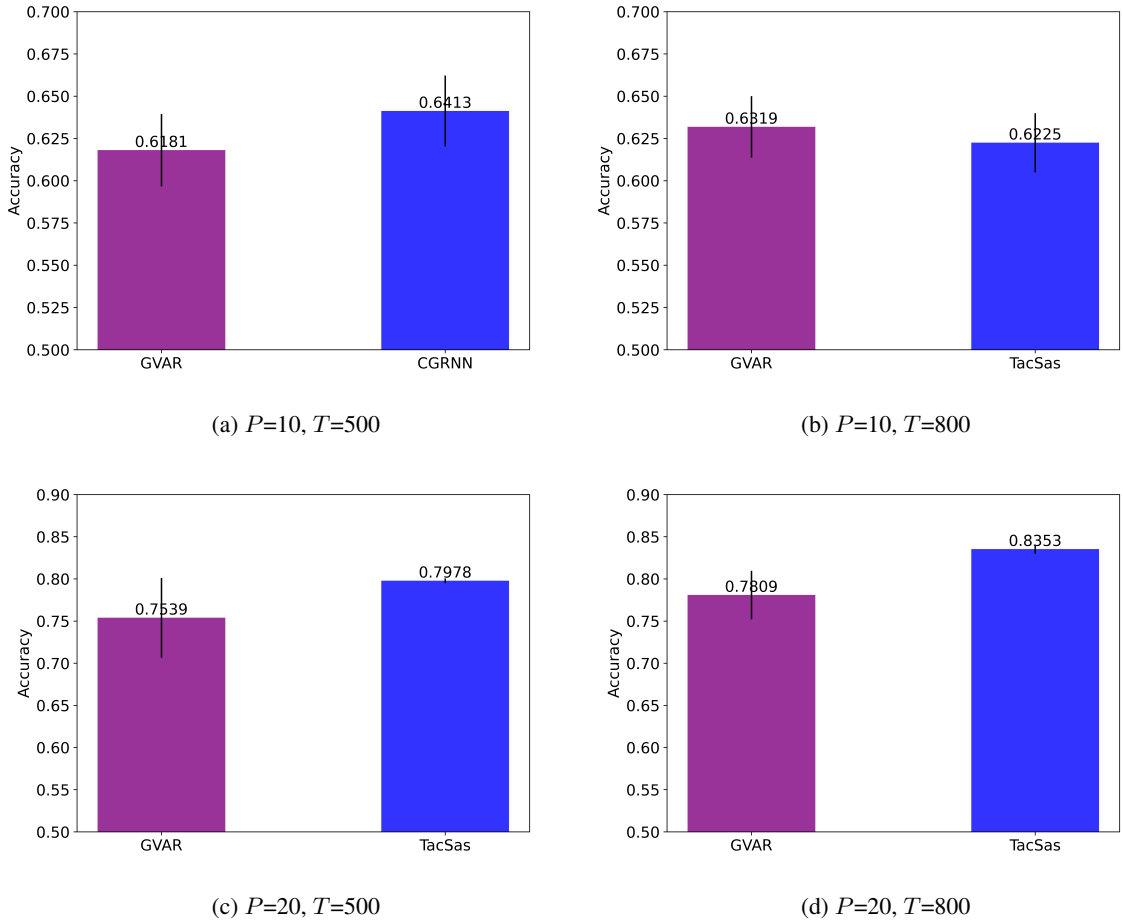
B.1. Ground-Truth Causality Discovery Ability of TacSas

Lorenz-96 model (Lorenz, 1996) is a famous synthetic system of multivariate time-series, e.g., $\boldsymbol{X} \in \mathbb{R}^{P \times T}$ is a P-dimensional time series whose dynamics can be modeled as follows.

$$\frac{d\boldsymbol{X}(i,t)}{dt} = (\boldsymbol{X}(i+1,t) - \boldsymbol{X}(i-2,t))\boldsymbol{X}(i-1,t) - \boldsymbol{X}(i,t) + F, \text{ for } i \in \{1, 2, \ldots, P\} \tag{17}$$

where $\boldsymbol{X}(0,t) = \boldsymbol{X}(P,t)$, $\boldsymbol{X}(-1,t) = \boldsymbol{X}(P-1,t)$, $\boldsymbol{X}(P+1,t) = \boldsymbol{X}(1,t)$, and F is the forcing constant determining the level of nonlinearity and chaos in the time series. With the above modeling, the corresponding ground-truth Granger causal structures can be simulated, involving multivariate, nonlinear, and sparse (Tank et al., 2022).

To generate the ground-truth causal structures, there are two parameters, i.e., the number of variables (i.e., P) and the number of timestamps (i.e., T). Therefore, we control these two parameters and report the accuracy of TacSas discovered causal structures against the ground-truth ones (i.e., 0/1 adjacency matrices), compared with the state-of-the-art causality discovery method GVAR (Marcinkevics & Vogt, 2021). The comparison is shown in Figure 4 after eight experiment trials with mean and variance computed, where we can observe our TacSas achieve the competitive accuracy of discovering the ground-truth causal structures. Also, by comparing Figure 4(a) and (b) (and Figure 4(c) and (d)), we can see that fixing the number of variables (i.e., P), increasing the time series length (i.e., T) may help discover the causality. And by comparing Figure 4(a) and (c) (and Figure 4(b) and (d)), we can see that fixing the time length (i.e., T), increasing the number of variables (i.e., P) may make the causality easier to be discovered.



(a) P=10, T=500

(b) P=10, T=800

(c) P=20, T=500

(d) P=20, T=800

Figure 4: Accuracy of Causality Discovery in Lorenz-96 with Varying Number of Variables (P) and Timestamps (T).

B.2. Validation of Anomaly Detection Ability of TacSas

Besides forecasting, another capability of TacSas is anomaly detection. Based on the analysis of Remark 3.3, the detection function of TacSas originates from the accurate expression of the feature distribution. Although our forecast features have better accuracy than selected baselines (e.g., DCGNN and GTS), we need to verify if the forecast features still have a negligible divergence from the ground-truth features in terms of distribution. If so, we can safely use the forecast features to detect anomalies. Therefore, we design the ablation study. We remove the forecasting part of TacSas i.e., we let the encoder and decoder in Figure 2 directly learn the distribution of ground-truth features (instead of forecast features) and then test reconstruction loss on ground-truth features. In Figure 5, we show the feature reconstruction loss (i.e., mean squared error) curve of the encoder and decoder on the validation set as the epoch increases. After the training of the encoder and decoder is converged, we can also observe that the ground-truth feature reconstruction loss does not have a very large divergence from the forecast features. Now, we are ready to do the following anomaly detection experiments.

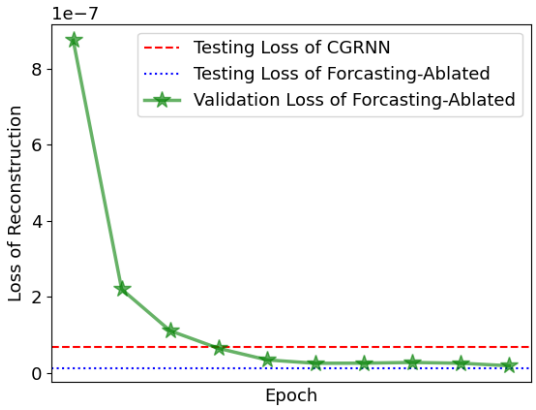


Figure 5: Ablation of TacSas on Cross-Validation Group #2 (i.e., 2018 as testing)

B.3. Ablation Study

As shown in Table 1, the GRU (Chung et al., 2014) method does not perform well. A latent reason is that it can not take any structural information from the time series. Motivated by this guess, we designed the following ablation study on the forecasting task. The ablated TacSas is designed by only keeping the forget gate in Eq. 9, i.e., the last equation in Eq. 9, then all the rest of the gates follow the GRU method. As shown in Figure 6, we can see that only taking partial time-respecting causal structural information could not enable TacSas to achieve the best performance, but accepting this partial information can help GRU improve the performance compared with Table 1.

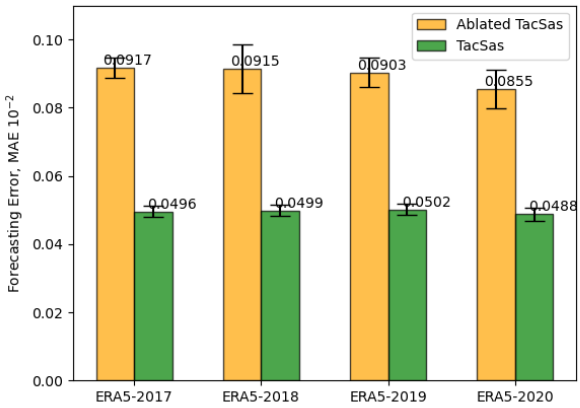


Figure 6: Ablation Study on Forecasting Task.

C. Implementation

C.1. Hyperparameter Search

In Eq. 5, instead of fixing the hyperparameter λ and c during the optimization. Increasing the values of hyperparameter λ and c can reduce the possibility that learned structures break the acyclicity (Yu et al., 2019), such that one iterative way to increase hyperparameters λ and c during the optimization can be expressed as follows.

$$\lambda_{i+1} \leftarrow \lambda_i + c_i \alpha(\boldsymbol{A}_i^{(t)}) \tag{18}$$

and

$$c_{i+1} = \begin{cases} \eta c_i & \text{if } |\alpha(\boldsymbol{A}_i^{(t)})| > \gamma |\alpha(\boldsymbol{A}_{i-1}^{(t)})| \\ c_i & \text{otherwise} \end{cases} \tag{19}$$

where $\eta > 1$ and $0 < \gamma < 1$ are two hyperparameters, the condition $|\alpha(\boldsymbol{A}_i^{(t)})| > \gamma |\alpha(\boldsymbol{A}_{i-1}^{(t)})|$ means that the current acyclicity $\alpha(\boldsymbol{A}_i^{(t)})$ at the i-th iteration is not ideal, because it is not decreased below the γ portion of $\alpha(\boldsymbol{A}_{i-1}^{(t)})$ from the last iteration $i-1$.

C.2. Reproducibility

For forecasting and anomaly detection, we have four cross-validation groups. For example, focusing on an interesting time interval each year (e.g., from May to August is the season for frequent thunderstorms), we set group #1 with [2018, 2019, 2020] as training, [2021] as validation, and [2017] as testing. Thus, we have 8856 hours, 45 weather features, and 238 counties in the training set. The rest three groups are {[2019, 2020, 2021], [2017], [2018]}, {[2020, 2021, 2017], [2018], [2019]}, and {[2021, 2017, 2018], [2019], [2020]}, respectively. Therefore, TacSas and baselines are required to forecast the testing set and detect the anomaly patterns in the testing set.

The persistence forecasting can be expressed as

$$\boldsymbol{X}_{TacSas++}^{(t)} = \alpha \boldsymbol{X}_{TacSas}^{(t)} + (1-\alpha)\boldsymbol{X}^{(t-\tau)} \text{ s.t. } \boldsymbol{X}_{TacSas}^{(t)} = \text{TacSas}(\boldsymbol{X}^{(t-\tau)}) \tag{20}$$

where τ is the time window, for example, in the experiments, τ = 24h.

TacSas is published [7]. The experiments are programmed based on Python and Pytorch on a Windows machine with 64GB RAM and a 16GB RTX 5000 GPU.

D. Tensor Time Series Dataset

D.1. Geographic Distribution of the Time Series Data

The geographic distribution of 238 selected counties in the United States of America is shown in Figure 7, where the circle with numbers denotes the aggregation of spatially near counties. Of 238 selected counties, 100 are selected for the top-ranked counties based on the yearly frequency of thunderstorms. The rest are selected randomly and try to provide extra information (e.g., causality discovery).

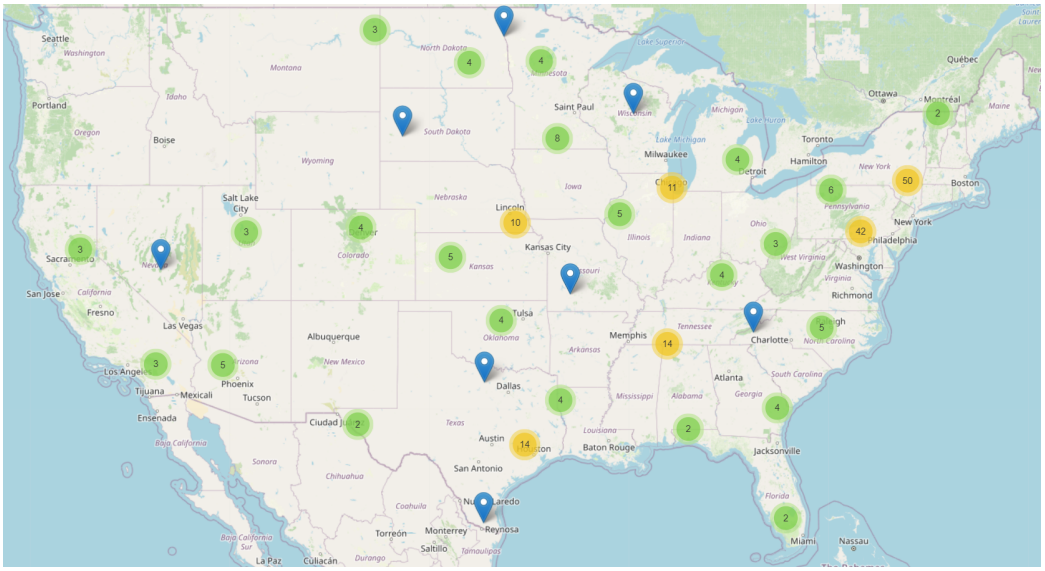


Figure 7: Geographic Distribution of Covered Counties in the Time Series Dataset (The number in the circle stands for the aggregation of nearby counties).

D.2. Abnormal Patterns of the Time Series Data

Table 3: Statistics of Anomaly Weather Patterns (i.e., Thunderstorm Winds) Occurrence in 238 Selected Counties in US.

Year	2017	2018	2019	2020	2021
Jan	26	3	2	41	7
Feb	53	6	9	50	8
Mar	85	16	26	63	62
Apr	93	44	140	170	60
May	245	207	263	175	218
Jun	770	302	348	331	452
Jul	306	291	457	453	701
Aug	294	269	415	354	435
Sep	61	80	122	29	123
Oct	32	32	82	60	55
Nov	20	22	9	114	11
Dec	5	15	11	8	58

[7] https://github.com/DongqiFu/TacSas