

# Modeling Time-evolving Causality over Data Streams

Naoki Chihara  
SANKEN, Osaka University  
Osaka, Japan  
naoki88@sanken.osaka-u.ac.jp

Ren Fujiwara  
SANKEN, Osaka University  
Osaka, Japan  
r-fujiwr88@sanken.osaka-u.ac.jp

Yasuko Matsubara  
SANKEN, Osaka University  
Osaka, Japan  
yasuko@sanken.osaka-u.ac.jp

Yasushi Sakurai  
SANKEN, Osaka University  
Osaka, Japan  
yasushi@sanken.osaka-u.ac.jp

## Abstract

Given an extensive, semi-infinite collection of multivariate co-evolving data sequences (e.g., sensor/web activity streams) whose observations influence each other, how can we discover the time-changing cause-and-effect relationships in co-evolving data streams? How efficiently can we reveal dynamical patterns that allow us to forecast future values? In this paper, we present a novel streaming method, MODEPLAIT, which is designed for modeling such causal relationships (i.e., time-evolving causality) in multivariate co-evolving data streams and forecasting their future values. The solution relies on characteristics of the causal relationships that evolve over time in accordance with the dynamic changes of exogenous variables. MODEPLAIT has the following properties: (a) *Effective*: it discovers the time-evolving causality in multivariate co-evolving data streams by detecting the transitions of distinct dynamical patterns adaptively. (b) *Accurate*: it enables both the discovery of time-evolving causality and the forecasting of future values in a streaming fashion. (c) *Scalable*: our algorithm does not depend on data stream length and thus is applicable to very large sequences. Extensive experiments on both synthetic and real-world datasets demonstrate that our proposed model outperforms state-of-the-art methods in terms of discovering the time-evolving causality as well as forecasting.

## CCS Concepts

• Information systems → Data stream mining.

## Keywords

Time series, Stream processing, Dynamical system

### ACM Reference Format:

Naoki Chihara, Yasuko Matsubara, Ren Fujiwara, and Yasushi Sakurai. 2025. Modeling Time-evolving Causality over Data Streams. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25)*, August 3–7, 2025, Toronto, ON, Canada. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3690624.3709283>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
KDD '25, Toronto, ON, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-1245-6/25/08  
<https://doi.org/10.1145/3690624.3709283>

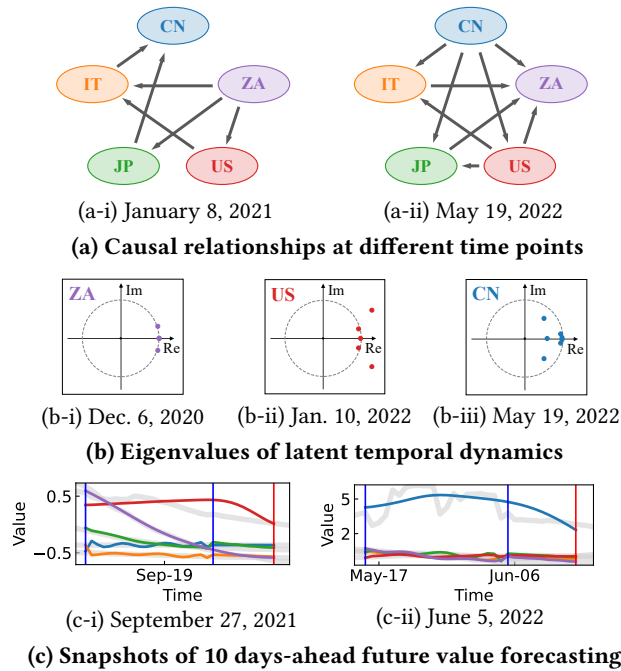
## 1 Introduction

In recent years, a substantial quantity of multivariate time series data has been collected from various events and applications related to the Internet of Things (IoT) [13, 35], web activities [29, 40], the spread of infectious diseases [30], and patterns of user behavior [38]. In real-world scenarios, in particular, these data are generated quickly and continuously, making it increasingly important to process them in a streaming fashion.

There are various relationships between observations in time series (e.g., correlation, independency) and they are critical features for a wide range of time series analyses, including clustering [24, 42], forecasting [41], imputation [56], and others [55]. Among them, causality [6, 53] offers particularly valuable insight, with many studies dedicated to its investigation [19, 26] and improving downstream tasks by employing it as an inductive bias, guiding the learning process towards more generalizable representations [9, 12]. However, most existing methods assume that causal relationships do not evolve over time in multivariate time series [48]. It is crucial to discover such causal relationships if we are to detect new causative factors promptly and accurately forecast in a streaming fashion. Their pivotal role becomes increasingly apparent upon recognizing that real-world data streams contain these relationships. For example, with the spread of infectious diseases, when a new virus strain emerges in a country, certain activities, such as cross-border travel, can lead to an increase in the number of infections in other countries, and the causative countries change over time. Here, we refer to such time-changing causal relationships as “*time-evolving causality*.” So, how can we discover the time-evolving causality in data streams and model semi-infinite multivariate data sequences?

Briefly, we design our model based on the structural equation model (SEM) [44], which is often utilized for learning causal structures. This model consists of two types of variables: endogenous variables and exogenous variables. The former are determined by the model itself, while the latter exist outside the model. In addition, exogenous variables are independent of each other and follow a non-Gaussian distribution. In our work, we consider that the causal relationships in data streams evolve over time in accordance with the dynamic changes of exogenous variables.

There are two difficulties involved in designing a model for discovering time-evolving causality. (i) *Latent temporal dynamics in univariate time series*: As pointed out above, exogenous variables in the structural equation model are independent of each other, so it is unsuitable to consider them as multivariate time series. Therefore,



**Figure 1: Modeling power of MODEPLAIT over an epidemiological data stream (i.e., #1 covid19): This original stream consists of daily COVID-19 infection numbers in five major countries. Our proposed method can (a) discover the causal relationships, which change over time, (b) extract the eigenvalues of the latent dynamics providing insight into them in terms of decay rate and temporal frequency, and (c) forecast future value in a stream fashion.**

we need to capture latent temporal dynamics in univariate time series to discover the time-evolving causality. However, it is challenging to design an appropriate system for univariate time series because the latent temporal dynamics in the system are generally multi-dimensional, and so a single dimension is insufficient for modeling the system. We solve this issue by expressing univariate time series as the superposition of computed basis vectors (i.e., modes) associated with decay rate and temporal frequency. (ii) *Distinct dynamical patterns*: Data streams typically contain various types of distinct dynamical patterns, and they are factors that change causal relationships over time. It is essential to understand their changes if we are to model an entire data stream effectively. For example, in the context of web search activities, we can identify various types of pattern changes caused by a multitude of reasons, such as a new item release. In addition, the event influences sales of other items, so the causal relationships could also change. We refer to these distinct dynamical patterns in data streams as “regimes.”

In this paper, we present MODEPLAIT<sup>1</sup>, which simultaneously and continuously discovers time-evolving causality in a multivariate co-evolving data stream and forecasts future values. In addition, thanks to desirable features of modes, MODEPLAIT extracts the temporal behavior of the latent dynamics in each exogenous variable in terms of decay rate and temporal frequency by analyzing the corresponding eigenvalues.

<sup>1</sup>Our source codes and datasets are available at [4]

In short, the problem we deal with is as follows:

*Given: a semi-infinite multivariate data stream  $X$ , which consists of  $d$ -dimensional vectors  $\mathbf{x}(t)$ , i.e.,  $X = \{\mathbf{x}(1), \dots, \mathbf{x}(t_c), \dots\}$ , where  $t_c$  is the current time,*

- **Find** distinct dynamical patterns (i.e., regimes),
- **Discover** causal relationships that changes in accordance with the transitions of regimes (i.e., time-evolving causality),
- **Forecast** an  $l_s$ -steps-ahead future value, i.e.,  $v(t_c + l_s)$ ,

*continuously and quickly, in a streaming fashion.*

## 1.1 Preview of Our Results

Figure 1 shows the results obtained with MODEPLAIT for modeling an epidemiological data stream (i.e., #1 covid19). This dataset consists of the number of COVID-19 infected patients in five countries (i.e., Japan, the United States, China, Italy, and the Republic of South Africa). Our method captures the following properties:

**Time-evolving causality.** Figure 1 (a) shows the causal relationships between observations, which change over time. The arrows indicate causality: the base of each arrow represents the “cause,” while the head represents the “effect.” MODEPLAIT successfully discovers the time-changing causal relationships between countries from an epidemiological data stream. For example, Figure 1 (a-i) shows that the Republic of South Africa had a causal influence on other countries. This finding corresponds to the fact that health officials announced the discovery of a new lineage of the coronavirus, namely 501.V2, in South Africa on December 18, 2020 [18], and indicates that MODEPLAIT adaptively discovered the influence of the new coronavirus on other countries. Additionally, Figure 1 (a-ii) shows that China had a causal influence on other countries in contrast to Figure 1 (a-i). This insight aligns with the period of one of the longest and toughest lockdowns in Shanghai, which lasted from early April 2022 to June 1, 2022 [8]. It implies that MODEPLAIT detected the influence of the spread of coronavirus infection in Shanghai, which led to a strict and long-term lockdown. In summary, the above discussions make it clear that MODEPLAIT can capture the transitions of distinct dynamical patterns in an epidemiological data stream and adaptively discover causal relationships at any given moment.

**Latent temporal dynamics.** Figure 1 (b) shows the eigenvalues of latent temporal dynamics in exogenous variables. These figures represent complex planes. The dotted gray lines are unit circles, and colored points are eigenvalues of latent temporal dynamics, whose magnitude and argument indicate the decay rate and temporal frequency of a specific mode, respectively. Specifically, if the absolute value of an eigenvalue is greater than 1, the corresponding mode exhibits growth; if it is less than 1, it exhibits decay (please see Section 3.1.1 for a detailed approach to reading these components). Figure 1 (b-i) shows the weak growth modes in exogenous variables for the Republic of South Africa, and implies an increase in infections in South Africa due to 501.V2 mentioned above. Figure 1 (b-ii) shows the strong growth modes in exogenous variables for the United States, where new infections surpassed 1 million in a single day for the first time [16]. Figure 1 (b-iii) shows the decay of exogenous variables for China. This period was toward the end of the lockdown in Shanghai, indicating that the spread of infections was beginning to ease, our result captures this precisely.

**Table 1: Capabilities of approaches.**

	ARIMA/++	TICC	NOTEARS/++	OrbitMap	TimesNet	MODEPLAIT
Stream Processing	-	-	-	✓	-	✓
Forecasting	✓	-	-	✓	✓	✓
Data Compression	-	✓	-	✓	-	✓
Interdependency	-	✓	✓	-	-	✓
Time-evolving Causality	-	-	-	-	-	✓

**$l_s$ -steps-ahead future values.** Figure 1 (c) shows snapshots of the  $l_s = 10$ -steps-ahead future forecasting when given a current window. The blue vertical axes show the beginning of a current window  $t_m$  and the current time point  $t_c$ , and the red vertical axis shows the  $l_s$ -step-ahead time point  $t_c + l_s$ . In addition, we show our estimated values with bold-colored lines (the originals are shown in gray). MODEPLAIT successfully finds the current distinct dynamical pattern and generates future values continuously at any time.

## 1.2 Contributions

In this paper, we propose MODEPLAIT, which has all of the following desirable properties:

- *Effective*: it discovers time-changing relationships between observations (i.e., time-evolving causality) based on monitoring transitions of distinct dynamical patterns (i.e., regimes).
- *Accurate*: it theoretically discovers the time-evolving causality in data streams (please see Lemma 2 for details), and accurately forecast future values based on these relationships.
- *Scalable*: it is fast and requires only constant computational time with regard to the entire stream length.

## 2 Related Work

In this section, we briefly describe investigations related to our work. Table 1 summarizes the relative advantages of MODEPLAIT with regard to five aspects.

**Time series modeling and forecasting.** Time series modeling and forecasting are important areas that have attracted huge interest in many fields. Autoregressive integrated moving average (ARIMA) [7] and Kalman filters (KF) [15] are representative examples of traditional modeling and forecasting methods, and there have been many studies of their derivatives [14, 33, 43, 51]. TICC [24] characterizes the interdependence between different observations based on a Markov random field but cannot capture the causal relationships. Additionally, streaming algorithms have become more critical in terms of processing a substantial amount of data under time/memory limitations, and they have proved highly significant to the data mining and database community [5, 23, 29, 36]. OrbitMap [37] is the latest general method focusing on stream forecasting, and it can find the transitions between major dynamic time series patterns. However, it cannot discover the time-evolving causality between observations. Research on deep learning models for forecasting has been very active in recent years [41, 49, 61]. TimesNet [58] is a TCN-based method that transforms a 1D time

**Table 2: Symbols and definitions.**

Symbol	Definition
$d$	Number of dimensions
$t_c$	Current time point
$X$	Co-evolving multivariate data stream (semi-infinite)
$X^c$	Current window, i.e., $X^c = X[t_m : t_c] \in \mathbb{R}^{d \times N}$
$h$	Embedding dimension
$g(\cdot)$	Observable for time-delay embedding, i.e., $g: \mathbb{R} \rightarrow \mathbb{R}^h$
$H$	Hankel matrix
$k$	Number of modes
$\Phi$	Modes of the system, i.e., $\Phi \in \mathbb{R}^{h \times k}$
$\Lambda$	Eigenvalues of the system, i.e., $\Lambda \in \mathbb{R}^{k \times k}$
$W$	Demixing matrix, i.e., $W = [w_1, \dots, w_d]^T \in \mathbb{R}^{d \times d}$
$B$	Causal adjacency matrix, i.e., $B \in \mathbb{R}^{d \times d}$
$e(t)$	Inherent signal at time point $t$ , i.e., $e(t) = \{e_{(i)}(t)\}_{i=1}^d$
$S(t)$	Latent vectors at time point $t$ , i.e., $S(t) = \{s_{(i)}(t)\}_{i=1}^d$
$v(t)$	Estimated vector at time point $t$ , i.e., $v(t) = \{v_{(i)}(t)\}_{i=1}^d$
$\mathcal{D}$	Self-dynamics factor set, i.e., $\mathcal{D} = \{\Phi, \Lambda\}$
$\theta$	Regime parameter set, i.e., $\theta = \{W, \mathcal{D}_{(1)}, \dots, \mathcal{D}_{(d)}\}$
$R$	Number of regimes
$\Theta$	Regime set, i.e., $\Theta = \{\theta^1, \dots, \theta^R\}$
$\mathcal{B}$	Time-evolving causality, i.e., $\mathcal{B} = \{B^1, \dots, B^R\}$
$\Omega$	Update parameter set, i.e., $\Omega = \{\omega^1, \dots, \omega^R\}$
$\mathcal{F}$	Full parameter set, i.e., $\mathcal{F} = \{\Theta, \Omega\}$

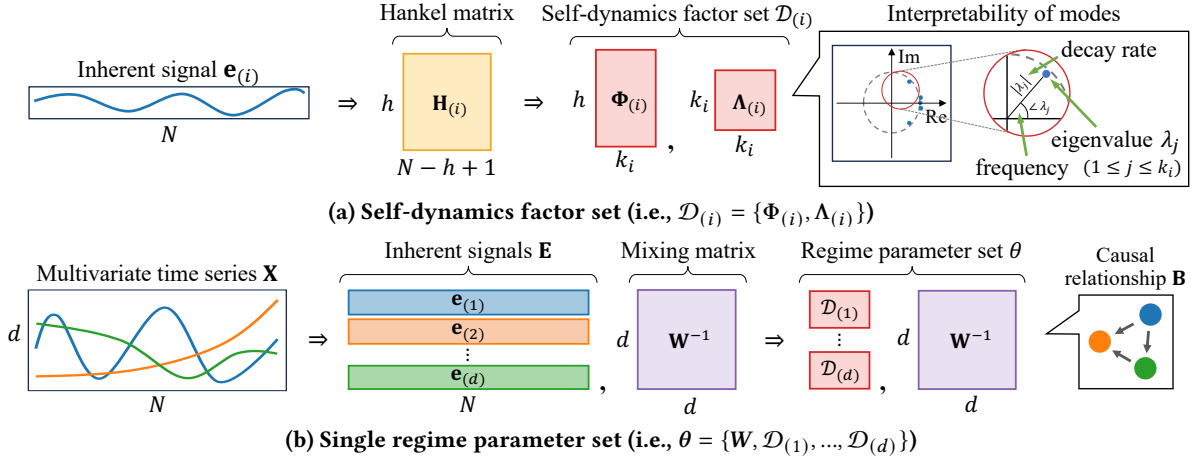
series into 2D space based on multiple periods and captures complex temporal variations for forecasting. Although deep learning-based methods are compelling, their applicability to forecasting in a streaming fashion is limited due to the prohibitively high computational costs associated with time series analysis, which hinders continuous model updating with the most recent observations.

**Causal inference/discovery.** Over decades, a wide range of studies have been conducted on causal inference/discovery [19, 26, 27, 34, 52] and addressing challenges based on the concept of causality [12, 47, 57]. NOTEARS [62] is a new differentiable optimization framework for learning directed acyclic graphs, utilizing an acyclic regularization as a replacement for a combinatorial constraint. Granger causality [21] has been widely used to analyze multivariate time series data. However, Granger causality only indicates the presence of a predictive relationship [46]. Specifically, typical causality represents whether one observation causes another, while Granger causality represents whether one observation forecasts another [22]. In this paper, we focus on the cause-and-effect relationships that evolve over time in a data stream. We try to discover them based on the structural equation model [44], which is one of the most general formulations of causality.

Consequently, none of these methods specifically focused on the discovery of the time-evolving causality and forecasting future values in a streaming fashion, simultaneously.

## 3 Proposed Model

In this section, we present our proposed model. The main symbols we use in this paper are described in Table 2. Here, before introducing the main topic, we briefly describe the principles and concepts of MODEPLAIT. We design our proposed model based on the structural equation model (SEM) [44], which is written as



**Figure 2: Illustration of MODEPLAIT:** (a) we extract the latent temporal dynamics from the  $i$ -th univariate inherent signal  $e_{(i)}$ , which behaves as a dynamical system. (b) The multivariate time series is described by mixing matrix  $W^{-1}$  and a collection of  $d$  self-dynamics factor sets  $\{\mathcal{D}_{(1)}, \dots, \mathcal{D}_{(d)}\}$ . The mixing matrix  $W^{-1}$  is not the same matrix as the causal adjacency matrix  $B$ , it is instrumental in identifying the time-evolving causality.

$X_{\text{sem}} = B_{\text{sem}}X_{\text{sem}} + E_{\text{sem}}$ , where  $X_{\text{sem}}$  is the observed variables,  $B_{\text{sem}}$  is the causal adjacency matrix, and  $E_{\text{sem}}$  is a set of mutually independent exogenous variables with a non-Gaussian distribution. Note that we assume that the data generating process is linear, the causal network is a directed acyclic graph, and there are no unobserved confounders in this paper. The structural equation model can express a typical causality, however, in real-world applications, causal relationships change over time in accordance with the transitions of distinct dynamical patterns. In our model, we assume that the exogenous variables behave as a dynamical system; however, it is inappropriate to consider their time-evolution as a single dynamical system due to their independence from each other. In summary, given a multivariate data stream which contains various distinct dynamical patterns (i.e., regimes), we aim to discover the time-evolving causality and summarize an entire data stream on the above assumption. Specifically, we need to capture the following properties to achieve the above objective:

- (P1) latent temporal dynamics of exogenous variables
- (P2) dynamical pattern in a single regime

So, how can we build our model that expresses both (P1) and (P2)? What is the acceptable mathematical model that summarizes a data stream and discovers the time-evolving causality? To handle (P1), we express each of the exogenous variables as the superposition of computed basis vectors (i.e., modes). We model (P2) by combining the above components. We provide our answers below.

### 3.1 Proposed Solution: MODEPLAIT

We now present our model in detail. First, we provide the definition for our proposed method.

**DEFINITION 1 (INHERENT SIGNALS:  $E$ ).** Let  $E$  be a bundle of  $d$  mutually independent signals with a non-Gaussian distribution, i.e.,  $E = \{e_{(i)}\}_{i=1}^d$ , where  $e_{(i)} = \{e_{(i)}(1), \dots, e_{(i)}(t)\}$  is the  $i$ -th univariate inherent signal. The main property is that they evolve over time.

Figure 2 is an illustration of our proposed model. In the first half of this section, we describe (P1) the latent temporal dynamics of the  $i$ -th univariate inherent signal  $e_{(i)}$  by introducing the self-dynamics factor set  $\mathcal{D}_{(i)}$ , and next, we propose the parameter set  $\theta$  to represent (P2) regimes and an entire data stream.

**3.1.1 Latent temporal dynamics of an inherent signal (P1).** First, we answer the fundamental question, namely, how can we extract the latent temporal dynamics from the  $i$ -th inherent signal (i.e.,  $e_{(i)} = \{e_{(i)}(1), \dots, e_{(i)}(t)\}$ ) and express it as a superposition of the modes? The difficulty arises from the fact that the latent dynamics in the system are generally multi-dimensional, making a single dimension inadequate for modeling the system. Here, we utilize state space augmentation methods to compensate for this inadequacy. In particular, we adopt time-delay embedding, which is effective in capturing nonlinear dynamics. Specifically, this is an established method for the geometric reconstruction of attractors for nonlinear systems based on the measurement of generic observables,  $g(e_{(i)}(t)) := (e_{(i)}(t), e_{(i)}(t-1), \dots, e_{(i)}(t-h+1)) \in \mathbb{R}^h$ , where  $h$  is the embedding dimension. We form the Hankel matrix  $H_{(i)} \in \mathbb{R}^{h \times (t-h+1)}$  by using the above observable  $g(\cdot)$ .

$$H_{(i)} = \begin{bmatrix} g(e_{(i)}(h)) & g(e_{(i)}(h+1)) & \cdots & g(e_{(i)}(t)) \\ | & | & & | \\ | & | & & | \\ | & | & & | \end{bmatrix} \quad (1)$$

As seen in Eq. (1), each state represented by a single measurement function is augmented with its past history. Furthermore, according to Takens' embedding theorem [54], it is guaranteed that time-delay embedding produces a vector whose dynamics are diffeomorphic to the dynamics of the original state under certain conditions. Intuitively, the reconstruction theoretically preserves the properties of the original dynamical system, allowing an analysis of Hankel matrix to reveal important features that may not be directly apparent in the original data alone. In many cases, an embedding dimension may be chosen without sacrificing the diffeomorphism.

We now extract the latent temporal dynamics expressed as the superposition of the modes from the  $i$ -th univariate inherent signal  $\mathbf{e}_{(i)}$  using the above method. We thus introduce a time-evolving activity for describing the dynamical system of the  $i$ -th inherent signal  $\mathbf{e}_{(i)}$ . This activity is a latent vector  $\mathbf{s}_{(i)}(t) \in \mathbb{C}^{k_i}$ , which is  $k_i$ -dimensional complex-valued latent vector at time point  $t$ , where  $k_i$  is the number of modes. Consequently, the dynamical system for the  $i$ -th univariate inherent signal  $\mathbf{e}_{(i)}$  can be described with the following equations:

**MODEL 1.** Let  $\mathbf{s}_{(i)}(t)$  be the  $k_i$ -dimensional latent vector at time point  $t$  for  $i \in \{1, \dots, d\}$ . The following equations govern the  $i$ -th univariate inherent signal  $\mathbf{e}_{(i)}$ ,

$$\begin{aligned} \mathbf{s}_{(i)}(t+1) &= \Lambda_{(i)} \mathbf{s}_{(i)}(t) \\ \mathbf{e}_{(i)}(t) &= g^{-1}(\Phi_{(i)} \mathbf{s}_{(i)}(t)) \end{aligned} \quad (2)$$

where  $g^{-1}(\cdot) : \mathbb{R}^h \rightarrow \mathbb{R}^h$  is the inverse of the observables  $g(\cdot)$ , each column of  $\Phi_{(i)}$  is one mode, and  $\Lambda_{(i)}$  is a diagonal matrix containing the  $k_i$  eigenvalues corresponding to those modes.

Note that the latent vector  $\mathbf{s}_{(i)}(t)$  is expressed as a superposition of  $k_i$  modes. The eigenvalues  $\Lambda_{(i)} \in \mathbb{C}^{k_i \times k_i}$  describe latent dynamical activities, and the modes  $\Phi_{(i)} \in \mathbb{C}^{h \times k_i}$  and  $g^{-1}(\cdot)$  are the observation projections that generate the  $i$ -th univariate inherent signal  $\mathbf{e}_{(i)}(t)$  at each time point  $t$ . Consequently, we have the following:

**DEFINITION 2 (SELF-DYNAMICS FACTOR SET:  $\mathcal{D}_{(i)}$ ).** Let  $\mathcal{D}_{(i)}$  be a set of modes  $\Phi_{(i)}$  and eigenvalues  $\Lambda_{(i)}$ , i.e.,  $\mathcal{D}_{(i)} = \{\Phi_{(i)}, \Lambda_{(i)}\}$ , which represents the latent temporal dynamics of the  $i$ -th univariate inherent signal  $\mathbf{e}_{(i)}$ .

We can interpret the features of the above model by focusing on the self-dynamics factor set  $\mathcal{D}_{(i)}$ . Specifically, the eigenvalues  $\Lambda_{(i)}$  imply the temporal dynamics of the modes  $\Phi_{(i)}$ , such as exponential growth/decay and oscillations. These are derived from a characteristic of a discrete dynamical system. Considering the eigenvalues  $\Lambda_{(i)}$  represent the behavior of a discrete dynamical system with sampling interval  $\Delta t$ , decay rate  $a$  and temporal frequency  $b$  of the  $j$ -th mode  $\varphi_j$  are shown as follows, using the  $j$ -th eigenvalue  $\lambda_j$ :

$$a = \frac{\text{Re}(\log \lambda_j)}{\Delta t}, \quad b = \frac{\text{Im}(\log \lambda_j)}{\Delta t}$$

where  $\text{Re}$  and  $\text{Im}$  are the real and imaginary parts, respectively. In addition, note that  $\log \lambda_j = \ln |\lambda_j| + i \arg \lambda_j$ , and it can be said that the decay rates and temporal frequencies of the modes are given by the absolute values and arguments of the eigenvalues, respectively.

**3.1.2 Dynamical pattern in a single regime (P2).** Thus, we have seen how to model the latent temporal dynamics of the  $i$ -th univariate inherent signal  $\mathbf{e}_{(i)}$  using self-dynamics factor set  $\mathcal{D}_{(i)}$ . Here, let us tackle the next question, namely how can we describe the major dynamical pattern (i.e., regime) considering the time-evolving causality between observations in a data stream? We establish a model to combine the  $d$  self-dynamics factor sets (i.e.,  $\mathcal{D}_{(1)}, \dots, \mathcal{D}_{(d)}$ ) for generating the estimated vector  $\mathbf{v}(t) \in \mathbb{R}^d$  at time point  $t$ . Also, we need a set of  $d$  latent vectors (i.e.,  $\mathbf{S}(t) = \{\mathbf{s}_{(i)}(t)\}_{i=1}^d$ ). Consequently, we extend Model 1, and the dynamical system for  $d$ -dimensional multivariate time series can be described with the following equations:

**MODEL 2.** Let  $\mathbf{s}_{(i)}(t)$  be the  $k_i$ -dimensional latent vector for the  $i$ -th univariate inherent signal  $\mathbf{e}_{(i)}(t)$  at time point  $t$ ,  $\mathbf{e}(t)$  be the  $d$ -dimensional inherent signals at time point  $t$  (i.e.,  $\mathbf{e}(t) = \{\mathbf{e}_{(i)}(t)\}_{i=1}^d$ ), and  $\mathbf{v}(t)$  be the  $d$ -dimensional estimated vector at time point  $t$ . The following equations govern the single regime,

$$\begin{aligned} \mathbf{s}_{(i)}(t+1) &= \Lambda_{(i)} \mathbf{s}_{(i)}(t) \quad (1 \leq i \leq d) \\ \mathbf{e}_{(i)}(t) &= g^{-1}(\Phi_{(i)} \mathbf{s}_{(i)}(t)) \quad (1 \leq i \leq d) \\ \mathbf{v}(t) &= \mathbf{W}^{-1} \mathbf{e}(t) \end{aligned} \quad (3)$$

In Model 2, we require an additional parameter, demixing matrix  $\mathbf{W}$ , which represents the relationships among  $d$  inherent signals (i.e.,  $\mathbf{e}_{(1)}, \dots, \mathbf{e}_{(d)}$ ) and is instrumental in identifying the time-evolving causality in data streams. However, there are indeterminacies in a mixing matrix (i.e., the inverse of a demixing matrix), so it is not the same matrix as the causal adjacency matrix  $\mathbf{B}$ . We present the algorithm for obtaining  $\mathbf{B}$  from  $\mathbf{W}$  in Section 4.2.3. Consequently, we have the following:

**DEFINITION 3 (SINGLE REGIME PARAMETER SET:  $\theta$ ).** Let  $\theta$  be a parameter set of a single regime, i.e.,  $\theta = \{\mathbf{W}, \mathcal{D}_{(1)}, \dots, \mathcal{D}_{(d)}\}$ , where  $\mathbf{W}$  serves as the basis for generating the causal adjacency matrix  $\mathbf{B}$ .

Furthermore, we want to detect the transitions of regimes, which induce changes in causal relationships. Let  $R$  denote the optimal number of regimes up to the time point  $t$ . Then, a data stream  $\mathbf{X}$  is summarized using a set of  $R$  regimes (i.e.,  $\theta^1, \dots, \theta^R$ ). Consequently, a regime set for a data stream  $\mathbf{X}$  and time-evolving causality are defined as follows:

**DEFINITION 4 (REGIME SET:  $\Theta$ ).** Let  $\Theta$  be a parameter set of multiple regimes, i.e.,  $\Theta = \{\theta^1, \dots, \theta^R\}$ , which describes multiple distinct dynamical patterns in an entire data stream.

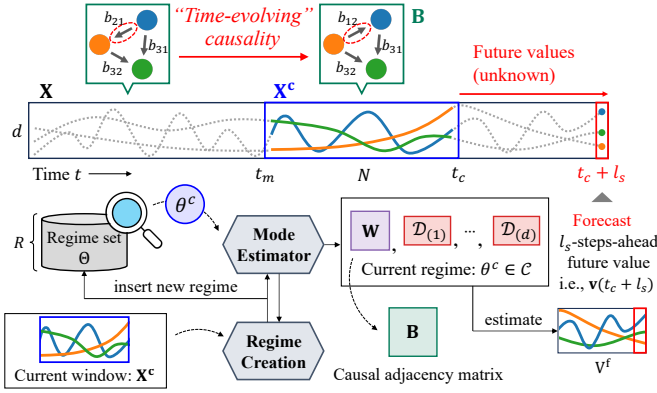
**DEFINITION 5 (TIME-EVOLVING CAUSALITY:  $\mathcal{B}$ ).** Let  $\mathcal{B}$  be a parameter set of causal adjacency matrices, i.e.,  $\mathcal{B} = \{\mathbf{B}^1, \dots, \mathbf{B}^R\}$ , which describes time-changing of causal relationships. Note that  $\mathbf{B}^i$  is a causal adjacency matrix corresponding to the  $i$ -th regime  $\theta^i$ .

## 4 Optimization Algorithm

Thus far, we have shown how we describe the major dynamical pattern using the demixing matrix  $\mathbf{W}$  which can be transformed into a causal adjacency matrix  $\mathbf{B}$ . In this section, we present an effective algorithm to identify the time-evolving causality  $\mathcal{B}$  and estimate the regime set  $\Theta$ . Figure 3 shows an overview of our proposed algorithm. We first present an effective way to estimate a new model parameter set from a multivariate time series, where we assume it has only a single regime. We then describe a streaming algorithm to identify  $\mathcal{B}$  and maintain  $\Theta$  incrementally for multiple distinct dynamical patterns, simultaneously.

### 4.1 REGIME CREATION

We first propose an algorithm, namely REGIME CREATION, for estimating a single regime parameter set  $\theta = \{\mathbf{W}, \mathcal{D}_{(1)}, \dots, \mathcal{D}_{(d)}\}$  from a data stream  $\mathbf{X}$ . The algorithm consists of two main steps: (i) it decomposes  $\mathbf{X}$  into a demixing matrix  $\mathbf{W}$  and inherent signals  $\mathbf{E}$  and (ii) it computes  $d$  self-dynamics factor sets  $\{\mathcal{D}_{(1)}, \dots, \mathcal{D}_{(d)}\}$  according to Eq. (2). We use ICA for the decomposition of  $\mathbf{X}$ , where it is essential for identifying the optimal causality. Next, as regards



**Figure 3: Overview of MODEPLAIT algorithm:** Given a data stream  $X$ , it performs all the following tasks at every time point  $t_c$ . Firstly, it searches for the best regime  $\theta^c$  for the current window  $X^c$ . It then forecasts the  $l_s$ -steps-ahead future value, i.e.,  $v(t_c + l_s)$  by utilizing the best one. When the algorithm encounters an unknown pattern in  $X^c$ , it estimates a new regime  $\theta$  and inserts it into  $\Theta$ .

computing the  $i$ -th self-dynamics factor set  $\mathcal{D}_{(i)}$ , assume that we have the following data matrices based on the Hankel matrix  $H_{(i)}$ :

$$\begin{aligned} L_{(i)} &= [g(e_{(i)}(h+1)) \quad \cdots \quad g(e_{(i)}(t))] \in \mathbb{R}^{h \times (t-h)} \\ R_{(i)} &= [g(e_{(i)}(h)) \quad \cdots \quad g(e_{(i)}(t-1))] \in \mathbb{R}^{h \times (t-h)} \end{aligned}$$

And, we use the following weight cost function:

$$\begin{aligned} \min_A \sum_{t'=h}^{t-1} \mu^{2(t-1-t')} \|g(e_{(i)}(t'+1)) - A_{(i)}g(e_{(i)}(t'))\|_2^2 \\ = \min_A \|(L_{(i)} - A_{(i)}R_{(i)})M\|_F^2 \end{aligned} \quad (4)$$

where,  $A_{(i)}$  is the transition matrix, from which the eigendecomposition yields the modes  $\Phi_{(i)}$  and the corresponding eigenvalues  $\Lambda_{(i)}$ , and  $M = \text{diag}(\mu^{t-h-1}, \dots, \mu^0) \in \mathbb{R}^{(t-h) \times (t-h)}$  is defined as a forgetting matrix, based on the recursive least squares principle. In addition, according to Koopman theory [32], while the transition matrix  $A_{(i)}$  is a linear operator, it is applicable even to nonlinear dynamical systems, unlike the classical modal decomposition of linear time-invariant systems. Specifically, the analytical algorithm proceeds as follows:

- I. Compute the ICA of  $X = W^{-1}E$ .
- II. Form the Hankel matrix  $H_{(i)}$  according to Eq. (1).
- III. Build a pair of data matrices  $(L_{(i)}, R_{(i)})$ .
- IV. Compute the SVD of  $R_{(i)}M = U_{(i)}\Sigma_{(i)}V_{(i)}^T$ . We automatically determine the optimal number of singular values  $k_i$  by [20].
- V. Project the transition matrix  $A_{(i)}$  onto the  $k_i$ -dimensional subspace spanned by the left singular vector  $U_{(i)}$ .

$$\tilde{A}_{(i)} = U_{(i)}^T A_{(i)} U_{(i)} = U_{(i)}^T L_{(i)} M V_{(i)} \Sigma_{(i)}^{-1} \in \mathbb{R}^{k_i \times k_i}$$

- VI. Compute the eigendecomposition of  $\tilde{A}_{(i)} Z_{(i)} = Z_{(i)} \Lambda_{(i)}$ . Note that the eigenvalues  $\Lambda_{(i)}$  are identical to the  $k_i$  leading eigenvalues of  $A_{(i)}$  because the left singular vector  $U_{(i)}$  is an orthogonal matrix.

- VII. Compute the modes  $\Phi_{(i)} = U_{(i)} Z_{(i)}$ .

**LEMMA 1 (TIME COMPLEXITY OF REGIMECREATION).** *The time complexity of REGIMECREATION is  $O(N(d^2 + h^2) + k^3)$ , where  $k = \max_i(k_i)$ . Please see Appendix A for details.*

## 4.2 Streaming Algorithm

Our next step is to answer the most important question: how can we employ our proposed model for identifying the causal adjacency matrix  $B$  from the demixing matrix  $W \in \theta$  and forecasting future values in a streaming fashion? Before turning to the main topic, we provide the definitions of some key concepts.

**DEFINITION 6 (UPDATE PARAMETER:  $\omega$ ).** *Let  $\omega$  be a parameter set for updating a regime  $\theta$ , i.e.,  $\omega = \{\{P_{(i)}\}_{i=1}^d, \{\epsilon_{(i)}\}_{i=1}^d\}$ , where  $P_{(i)} = (R_{(i)} M R_{(i)}^T)^{-1}$  and  $\epsilon_{(i)}$  is the energy.*

**DEFINITION 7 (FULL PARAMETER SET:  $\mathcal{F}$ ).** *Let  $\mathcal{F}$  be a full parameter set of MODEPLAIT, i.e.,  $\mathcal{F} = \{\Theta, \Omega\}$ , where  $\Theta$  and  $\Omega$  consist of  $R$  regimes and update parameters, respectively, namely,  $\Theta = \{\theta^1, \dots, \theta^R\}$ , and  $\Omega = \{\omega^1, \dots, \omega^R\}$ .*

With the above definitions, the formal problem is as follows:

**PROBLEM 1.** *Given a multivariate data stream  $X$ , where  $x(t_c)$  is the most recent value at time point  $t_c$ ,*

- **Find** the optimal full parameter set, i.e.,  $\mathcal{F} = \{\Theta, \Omega\}$ ,
- **Discover** the time-evolving causality, i.e.,  $\mathcal{B}$ ,
- **Forecast** an  $l_s$ -steps-ahead future value, i.e.,  $v(t_c + l_s)$ .

Here, we refer to the regime for the current window  $X^c = X[t_m : t_c]$  as  $\theta^c$ , and the update parameter corresponding to  $\theta^c$  as  $\omega^c$ . In addition, we need the latent vectors  $S(t_c)$  at the current time  $t_c$  for forecasting an  $l_s$ -steps-ahead future value  $v(t_c + l_s)$ , and so keep it as  $S_{en}^c$ . In summary, our proposed algorithm keeps them as the model candidate  $C = \{\theta^c, \omega^c, S_{en}^c\}$  for stream processing.

**4.2.1 Overview.** We now introduce our streaming algorithm, MODEPLAIT, which consists of the following algorithms. For detailed descriptions and pseudocode, we would refer you to Algorithm 1 in Appendix A.

- **MODEESTIMATOR:** Estimates the optimal full parameter set  $\mathcal{F}$  and the model candidate  $C$  which appropriately describes the current window  $X^c$ .
- **MODEGENERATOR:** Forecasts an  $l_s$ -steps-ahead future value, i.e.,  $v(t_c + l_s)$ , and identifies the causal adjacency matrix  $B$ , using the model candidate  $C$ .
- **REGIMEUPDATER:** Updates the current regime  $\theta^c$  using update parameter  $\omega^c \in C$  and the most recent value  $x(t_c)$ . This is only performed if a new regime is not created.

The following sections provide detailed explanations.

**4.2.2 MODEESTIMATOR.** Given a new value  $x(t_c)$  at the current time  $t_c$ , we first need to update incrementally the full parameter set  $\mathcal{F}$  and the model candidate  $C$ , which best describes the current window  $X^c$ . Algorithm 2 (See Appendix A) is the MODEESTIMATOR algorithm in detail. Here, let  $f(X^c; S_0^c, \theta^c)$  be a new function for estimating the optimal parameter so that it minimizes the mean square errors between the current window  $X^c$  and the estimated window  $V^c$  in Model 2, i.e.,  $f(X^c; S_0^c, \theta^c) = \sum_{t=t_m+h-1}^{t_c} \|x(t) - v(t)\|$ , where  $S_0^c$  represents the latent vectors at time point  $t_m + h - 1$ . Note that when

embedding the time series using  $g(\cdot)$ , the number of data points (namely, the number of columns in the Hankel matrix  $H$ ) is partially reduced compared with before embedding. The most straightforward way to determine  $S_0^c$  is to adopt  $\{\Phi_{(i)}^\dagger g(e_{(i)}(t_m + h - 1))\}_{i=1}^d$  according to Eq. (3). However, the noisy initial conditions give rise to unexpected forecasting. Therefore, we optimize  $S_0^c$  by using the Levenberg-Marquardt (LM) algorithm [39] and thus enable the effects of noise in observations to be removed. Here, we return to the MODEESTIMATOR algorithm, which proceeds as follows:

- I. It optimizes initial condition  $S_0^c$ , so that it minimizes the errors between the current window  $X^c$  and the current regime  $\theta^c$ .
- II. If  $f(X^c; S_0^c, \theta^c) > \tau$ , it searches for a better regime  $\theta \in \Theta$ .
- III. If  $f(X^c; S_0^c, \theta^c) > \tau$  still holds, it creates a new regime for  $X^c$  using REGIMECREATION, and inserts it into  $\Theta$ .

**4.2.3 MODEGENERATOR.** The next algorithm is MODEGENERATOR, which incrementally identifies the causal adjacency matrix  $B$  and forecasts an  $l_s$ -steps-ahead future value  $v(t_c + l_s)$  by using the model candidate  $C$ . As for forecasting, it generates the value of  $v(t_c + l_s)$  according to Eq. (3) with the most suitable regime  $\theta^c$  for  $X^c$ , which is selected by MODEESTIMATOR. On the other hand, we identify the causal adjacency matrix  $B$  from the demixing matrix  $W \in \theta^c$ . A mixing matrix (i.e., the inverse of a demixing matrix) typically has the two major indeterminacies: the order and scaling of the independent components; however, we must address the above difficulties if we are to identify the optimal causal adjacency matrix [52]. Consequently, the algorithm for identifying the causal adjacency matrix  $B$  proceeds as follows:

- I. Find the permutation of rows of  $W$  that yields a matrix  $\tilde{W}$  without any zeros on the main diagonal.
- II. Divide each row of  $\tilde{W}$  by its corresponding diagonal element to yield a new matrix  $\tilde{W}'$  with all ones on the diagonal.
- III. Compute an estimate  $\hat{B}$  of  $B$  using  $\hat{B} = I - \tilde{W}'$ .
- IV. Finally, to find a causal order, compute the permutation matrix  $K$  of  $\hat{B}$  that yields a matrix  $\tilde{B} = K\hat{B}K^\top$ , which minimizes the sum of the elements in the upper triangular part of  $\tilde{B}$ .

This algorithm resolves two major indeterminacies in a mixing matrix in steps I and II. Moreover, it finds the causal order, in other words, it removes the insufficient connection in step IV. The causal relationships are already identified up to step III, but this step is important for visualizing the resulting directed acyclic graph.

**LEMMA 2 (CAUSAL IDENTIFIABILITY).** *Causal discovery in MODEPLAIT is equivalent to finding the causal adjacency matrix  $B$  in MODEGENERATOR. Please see Appendix A for details.*

This lemma demonstrates theoretically that our proposed algorithm is capable of discovering causal relationships.

**4.2.4 REGIMEUPDATER.** Finally, when an existing regime is selected as the current regime  $\theta^c$  from the regime set  $\Theta$ , we update its parameters (i.e.,  $W, \mathcal{D}^{(1)}, \dots, \mathcal{D}^{(d)}$ ) using a new value  $x(t_c)$  to ensure that this regime represents a more sophisticated dynamical pattern. In short, REGIMEUPDATER has two parts: (i) update the demixing matrix  $W$  and (ii) update each self-dynamics factor set  $\mathcal{D}_{(i)}$ . In part (i), we use an algorithm based on adaptive filtering techniques [25, 59].

It is so efficient in terms of computational and memory requirements, while converging quickly, with no special parameters to tune. Here, we briefly describe the update process for  $W$ :

- I. Compute the  $i$ -th univariate inherent signal  $g(e_{(i)}(t_c))$  at the current time  $t_c$ , by projecting  $x(t_c)$  onto  $w_i$ , which is the  $i$ -th row vector of  $W$ , before the update.
- II. Estimate the reconstruction error and the energy  $\epsilon_{(i)}$  based on the value of  $g(e_{(i)}(t_c))$ ,
- III. Update the estimates of  $w_i$  using error and energy  $\epsilon_{(i)}$ .

In part (ii), we update the self-dynamics factor set  $\mathcal{D}_{(i)}$  by utilizing the following recurrence:

$$\begin{aligned} A_{(i)}^{new} &= A_{(i)}^{prev} + (g(e_{(i)}(t_c)) - A_{(i)}^{prev} g(e_{(i)}(t_c - 1))) \gamma_{(i)} \\ \gamma_{(i)} &= \frac{g(e_{(i)}(t_c - 1))^\top P_{(i)}^{prev}}{\mu + g(e_{(i)}(t_c - 1))^\top P_{(i)}^{prev} g(e_{(i)}(t_c - 1))} \\ P_{(i)}^{new} &= \frac{1}{\mu} (P_{(i)}^{prev} - P_{(i)}^{prev} g(e_{(i)}(t_c - 1)) \gamma_{(i)}) \end{aligned} \quad (5)$$

In this equation,  $\Phi_{(i)}$  and  $\Lambda_{(i)}$  are eigenvectors and eigenvalues of  $A_{(i)}$ , respectively. This recurrence minimizes the weighted cost function as outlined in Eq. (4), focusing on a new value  $x(t_c)$ , thereby adapting to most recent patterns. See Appendix A for details regarding the above formula. In summary, REGIMEUPDATER proceeds in the following manner:

- I. In accordance with the part (i) algorithm, update the demixing matrix  $W$  using a new value  $x(t_c)$ .
- II. Compute the current inherent signals  $E^c$  from the current window  $X^c$  using the updated demixing matrix  $W$ .
- III. Update each self-dynamics factor set  $\mathcal{D}_{(i)}$  according to Eq. (5).

**LEMMA 3 (TIME COMPLEXITY OF MODEPLAIT).** *Based on Lemma 1, the time complexity of MODEPLAIT is at least  $O(N \sum_i k_i + dh^2)$ , and at most  $O(RN \sum_i k_i + N(d^2 + h^2) + k^3)$  per process. Please see Appendix A for details.*

This theoretical analysis indicates that our proposed algorithm requires only constant computational time with regard to the entire data stream length  $t_c$ . Therefore, MODEPLAIT is practical for semi-infinite data streams in terms of execution speed.

## 5 Experiments

In this section, we evaluate the performance of MODEPLAIT using both synthetic and real-world datasets. We performed extensive experiments to answer the following questions.

- Q1. *Effectiveness:* How well does it find the time-evolving causality?
- Q2. *Accuracy:* How accurately does it discover time-evolving causality and forecast future values?
- Q3. *Scalability:* How does it scale in terms of computational time?

**Datasets & experimental setup.** We used the following datasets:

- (#0) *synthetic:* was generated based on a structural equation model [44]. Details are provided in Appendix B.
- (#1) *covid19:* was obtained from Google COVID-19 Open Data [2] and consists of the number of infections in Japan, the United States, China, Italy, and the Republic of South Africa, covering over 900 daily entries.

**Table 3: Causal discovering results with multiple temporal sequences to encompass various types of real-world scenarios.**

Models	MODEPLAIT		CASPER		DARING		NoCurl		NO-MLP		NOTEARS		LiNGAM		GES	
Metrics	SHD	SID	SHD	SID	SHD	SID	SHD	SID	SHD	SID	SHD	SID	SHD	SID	SHD	SID
1, 2, 1	<b>3.82</b>	<b>4.94</b>	5.58	<u>7.25</u>	5.75	8.58	6.31	9.90	6.36	8.74	<u>5.03</u>	9.95	7.13	8.23	7.49	11.7
1, 2, 3	<b>4.48</b>	<b>6.51</b>	5.97	8.44	5.81	9.17	6.13	9.51	6.44	8.77	<u>5.69</u>	9.56	6.79	<u>7.33</u>	7.03	10.1
1, 2, 2, 1	<b>4.32</b>	<b>5.88</b>	5.41	<u>8.41</u>	6.54	9.17	6.69	10.0	6.55	8.72	<u>5.23</u>	9.54	7.12	8.65	7.08	9.77
1, 2, 3, 4	<b>4.21</b>	<b>5.76</b>	6.22	<u>8.33</u>	6.12	9.58	6.10	9.61	6.62	8.87	<u>5.73</u>	10.1	7.10	8.50	7.29	11.3
1, 2, 3, 2, 1	<b>4.50</b>	<b>6.11</b>	6.02	8.28	<u>5.45</u>	<u>7.77</u>	6.20	9.83	6.56	8.83	<u>5.57</u>	9.11	7.46	8.05	7.74	12.1

**Table 4: Multivariate forecasting results for both synthetic and real-world datasets. We used forecasting steps  $l_s \in \{5, 10, 15\}$ .**

Models	MODEPLAIT		TimesNet		PatchTST		DeepAR		OrbitMap		ARIMA		
Metrics	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	
#0 synthetic	5	<b>0.722</b>	<b>0.528</b>	0.805	0.578	<u>0.768</u>	0.581	1.043	0.821	0.826	<u>0.567</u>	0.962	0.748
	10	<b>0.829</b>	<b>0.607</b>	<u>0.862</u>	0.655	0.898	0.649	1.073	0.849	0.896	<u>0.646</u>	0.966	0.752
	15	<b>0.923</b>	<b>0.686</b>	<u>0.940</u>	<u>0.699</u>	0.973	0.706	1.137	0.854	0.966	<u>0.710</u>	0.982	0.765
#1 covid19	5	<b>0.588</b>	<b>0.268</b>	0.659	0.314	<u>0.640</u>	<u>0.299</u>	1.241	0.691	1.117	0.646	1.259	0.675
	10	<b>0.740</b>	<b>0.361</b>	<u>0.841</u>	<u>0.410</u>	1.053	0.523	1.255	0.693	1.353	0.784	1.260	0.687
	15	<b>0.932</b>	<b>0.461</b>	<u>1.026</u>	<u>0.516</u>	1.309	0.686	1.265	0.690	1.351	0.792	1.277	0.718
#2 web-search	5	<b>0.573</b>	<b>0.442</b>	<u>0.626</u>	<u>0.469</u>	0.719	0.551	1.255	1.024	0.919	0.640	1.038	0.981
	10	<b>0.620</b>	<b>0.481</b>	<u>0.697</u>	<u>0.514</u>	0.789	0.604	1.273	1.044	0.960	0.717	1.247	1.037
	15	<b>0.646</b>	<b>0.505</b>	<u>0.701</u>	<u>0.527</u>	0.742	0.571	1.300	1.069	0.828	0.631	1.038	0.795
#3 chicken-dance	5	<b>0.353</b>	<b>0.221</b>	0.759	0.490	<u>0.492</u>	<u>0.303</u>	0.890	0.767	0.508	0.316	2.037	1.742
	10	<b>0.511</b>	<b>0.325</b>	0.843	0.564	0.838	0.535	0.886	0.753	<u>0.730</u>	<u>0.476</u>	1.863	1.530
	15	<b>0.653</b>	<b>0.419</b>	0.883	0.592	0.972	0.654	<u>0.862</u>	0.718	0.903	<u>0.565</u>	1.792	1.481
#4 exercise	5	<b>0.309</b>	<b>0.177</b>	0.471	<u>0.275</u>	0.465	0.304	<u>0.408</u>	0.290	0.424	<u>0.275</u>	1.003	0.748
	10	<b>0.501</b>	<b>0.309</b>	0.630	0.381	0.789	0.518	<u>0.509</u>	0.382	0.616	<u>0.377</u>	1.104	0.814
	15	<u>0.687</u>	<b>0.433</b>	0.786	0.505	1.147	0.758	<b>0.676</b>	0.475	0.691	<u>0.434</u>	1.126	0.901

- (#2) *web-search*: consists of web-search counts collected over ten years related to beer queries on Google [3].
- (#3) *chicken-dance*, (#4) *exercise*: were obtained from the CMU motion capture database [1] and consist of four dimensional vectors (left/right legs and arms).

We compared our algorithm with the following seven baselines for causal discovery, namely CASPER [34], DARING [26], NoCurl [60], NOTEARS-MLP (NO-MLP) [63], NOTEARS [62], LiNGAM [52], and GES [10]. Besides, we also compared with the five following competitors for forecasting, namely TimesNet [58], PatchTST [41], DeepAR [49], OrbitMap [37], and ARIMA [7]. Details regarding the experimental settings are also provided in Appendix B.

**Q1. Effectiveness.** We first demonstrated how effectively MODEPLAIT discovers the time-evolving causality and forecasts future values in a streaming fashion using the epidemiological data stream (i.e., #1 *covid19*). Recall that Figure 1 shows MODEPLAIT modeling and forecasting results. Figure 1 (a/b) shows graphical representations of the causal adjacency matrix  $B$  and the eigenvalues  $\Lambda$ . Most importantly, the causal relationships evolve over time in accordance with the transitions of distinct dynamical patterns in the inherent signals  $E$ . MODEPLAIT can continuously detect new actual causative events around the world (e.g., the discovery of a new lineage of the coronavirus in South Africa, the abrupt increase in coronavirus infections in the United States, and the strict, long-term

lockdown in Shanghai). Figure 1 (c) shows streaming time series forecasting results. There have been multiple distinct patterns (e.g., a rapid decrease in infection numbers in the Republic of South Africa), MODEPLAIT adaptively captures the exponential patterns and forecasts future values close to the originals.

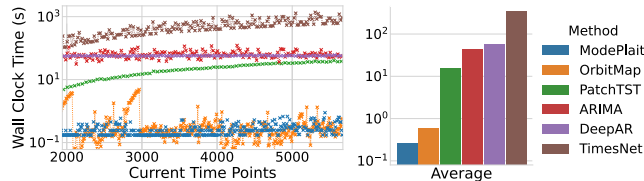
**Q2-1. Causal discovering accuracy.** We next showed how accurately MODEPLAIT can discover the time-evolving causality. We reported the structural Hamming distance (SHD) and the structural intervention distance (SID) [45]. SHD quantifies the difference in the causal adjacency matrix by counting missing, extra, and reversed edges and SID is particularly suited for evaluating causal discovering accuracy since it counts the number of couples  $(i, j)$  such that the interventional distribution  $p(x_j | \text{do}(X_i = \bar{x}))$  would be miscalculated if we used the estimated causal adjacency matrix. Both metrics should be lower to represent better estimated adjacency matrices. Table 3 shows the causal discovering results of MODEPLAIT and its baselines for various synthetic datasets, where the best and second-best levels of performance are shown in **bold** and underlined, respectively. Our method outperformed all baselines for every temporal sequence, which is consistent with the analysis provided in Lemma 2. This is because none of the competitors can handle the time-evolving causality in data streams.

**Q2-2. Forecasting accuracy.** We evaluated the quality of MODEPLAIT in terms of  $l_s$ -steps-ahead forecasting accuracy. For this evaluation, we adopted the root mean square error (RMSE) and the



**Table 5: Ablation study results with forecasting steps  $l_s \in \{5, 10, 15\}$  for both synthetic and real-world datasets.**

Datasets		#0 synthetic		#1 covid19		#2 web-search		#3 chicken-dance		#4 exercise	
Metrics		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
MODEPLAIT (full)	5	0.722	0.528	0.588	0.268	0.573	0.442	0.353	0.221	0.309	0.177
	10	0.829	0.607	0.740	0.361	0.620	0.481	0.511	0.325	0.501	0.309
	15	0.923	0.686	0.932	0.461	0.646	0.505	0.653	0.419	0.687	0.433
w/o causality	5	0.759	0.563	0.758	0.374	0.575	0.437	0.391	0.262	0.375	0.218
	10	0.925	0.696	0.848	0.466	0.666	0.511	0.590	0.398	0.707	0.433
	15	1.001	0.760	1.144	0.583	0.708	0.545	0.821	0.537	0.856	0.533



**Figure 4: Scalability of MODEPLAIT: (left) Wall clock time vs. data stream length  $t_c$  and (right) average time consumption for (#4) exercise. The vertical axis of these graphs is a logarithmic scale. MODEPLAIT is superior to its competitors. It is up to 1,500x faster than its competitors.**

mean absolute error (MAE), both of which provide good results when they are close to zero. For all methods, we used one-third of the sequences to tune their parameters. Table 4 presents the overall forecasting results, where the best results are in **bold** and the second-best are underlined. For brevity, we only reported results of a representative synthetic dataset, which has the most complicated temporal sequence, “1, 2, 3, 2, 1”. We compared the two metrics when we varied the forecasting step (i.e.,  $l_s \in \{5, 10, 15\}$ ). Our method achieved remarkable improvements over its competitors. While deep learning models (TimesNet, PatchTST, and DeepAR) exhibit high generality for time series modeling, their forecasting accuracy was poorer because they could not adjust the model parameters incrementally. OrbitMap is capable of handling multiple discrete nonlinear dynamics but misses the time-evolving causality, and thus was outperformed by our proposed method. ARIMA assumes linear relationships between time series data and so fails to accommodate complex and nonlinear data resulting in decreased forecasting accuracy.

**Q2-3. Ablation study.** To quantitatively evaluate the impact of causal relationships on forecasting effectiveness, we additionally performed an ablation study by comparing a limited version of our method, namely *w/o causality*, whose demixing matrix  $W$  was fixed to the identity matrix. Table 5 presents the overall results of our ablation study on MODEPLAIT using both synthetic and real-world datasets. We can see that the *w/o causality* causes a significant drop in forecasting accuracy across all experimental settings. Therefore, we observed that the discovery of time-evolving causality in data streams boosts forecasting accuracy.

**Q3. Scalability.** Finally, we evaluated the computational time needed by our streaming algorithm. Figure 4 compares the computational efficiencies of MODEPLAIT and its competitors. It presents the computational time at each time point  $t_c$  on the left, and the average computational time on the right. Note that both figures are

shown on linear-log scales. Our method consistently outperformed its competitors in terms of computational time thanks to our incremental update, which aligns with the discussion presented in Lemma 3. OrbitMap was competitive, but it estimates model parameters via iterative optimization, the expectation-maximization algorithm, which makes it slower than our proposed algorithm. Other methods require a significant amount of learning time because they cannot update their models incrementally.

## 6 Conclusion

In this paper, we focused on the summarization of an entire data stream, discovering the time-evolving causality in data streams, and forecasting future values incrementally. Our proposed method, namely MODEPLAIT, exhibits all of the following desirable properties that we listed in the introduction:

- It is *Effective*: It provides the time-evolving causality, namely insightful time-changing causal relationships in data streams.
- It is *Accurate*: Our experiments demonstrated that MODEPLAIT precisely discovers the time-evolving causality and forecasts future values in a streaming fashion.
- It is *Scalable*: The computational time for our proposed algorithm does not depend on the data stream length.

**Acknowledgment.** We would like to thank the anonymous referees for their valuable and helpful comments. This work was partly supported by “Program for Leading Graduate Schools” of the Osaka University, Japan, JST CREST JPMJCR23M3, JSPS KAKENHI Grant-in-Aid for Scientific Research Number JP24KJ1618.

## References

- [1] CMU motion capture database. <http://mocap.cs.cmu.edu/>.
- [2] Google COVID-19 Open Data Repository. <https://health.google.com/covid-19/open-data/>.
- [3] Google Trends. <https://trends.google.co.jp/trends/>.
- [4] Source codes and datasets. <https://github.com/C-Naoki/ModePlait>.
- [5] Charu C. Aggarwal (Ed.). 2007. *Data Streams - Models and Algorithms*. Advances in Database Systems, Vol. 31. Springer.
- [6] Kenneth A Bollen. 1989. *Structural equations with latent variables*. John Wiley & Sons.
- [7] George EP Box and Gwilym M Jenkins. 1976. *Time series analysis: forecasting and control (Revised Edition)*. John Wiley & Sons.
- [8] C. Buckley. 2022. Relief, Reunions and Some Anxiety as Shanghai (Mostly) Reopens. *New York Times* (Jun. 1, 2022).
- [9] Yuxiao Cheng, Runzhao Yang, Tingxiang Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. 2023. CUTS: Neural Causal Discovery from Irregular Time-Series Data. In *The Eleventh International Conference on Learning Representations*.
- [10] David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *Journal of machine learning research* 3, Nov (2002), 507–554.
- [11] Pierre Comon. 1994. Independent component analysis, a new concept? *Signal processing* 36, 3 (1994), 287–314.

- [12] Enyan Dai and Jie Chen. 2022. Graph-Augmented Normalizing Flows for Anomaly Detection of Multiple Time Series. In *The Tenth International Conference on Learning Representations*.
- [13] Gianmarco De Francisci Morales, Albert Bifet, Latifur Khan, Joao Gama, and Wei Fan. 2016. Iot big data stream mining. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2119–2120.
- [14] Alysha M De Livera, Rob J Hyndman, and Ralph D Snyder. 2011. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American statistical association* 106, 496 (2011), 1513–1527.
- [15] James Durbin and Siem Jan Koopman. 2012. *Time series analysis by state space methods*. Vol. 38. OUP Oxford.
- [16] H. Ellyatt. 2022. U.S. reports over 1 million new daily Covid cases as omicron surges. *CNBC* (Jan. 4 2022).
- [17] Paul Erdos, Alfréd Rényi, et al. 1960. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5, 1 (1960), 17–60.
- [18] S. Fink. 2020. South Africa announces a new coronavirus variant. *New York Times* (Dec. 21, 2020).
- [19] Daigo Fujiwara, Kazuki Koyama, Keisuke Kiritoshi, Tomomi Okawachi, Tomonori Izumitani, and Shohei Shimizu. 2023. Causal discovery for non-stationary non-linear time series data using just-in-time modeling. In *Conference on Causal Learning and Reasoning*. PMLR, 880–894.
- [20] Matan Gavish and David L Donoho. 2014. The optimal hard threshold for singular values is  $4/\sqrt{3}$ . *IEEE Transactions on Information Theory* 60, 8 (2014), 5040–5053.
- [21] Clive WJ Granger. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society* (1969), 424–438.
- [22] Clive William John Granger and Paul Newbold. 2014. *Forecasting economic time series*. Academic press.
- [23] Michael Hahsler and Matthew Bolaños. 2016. Clustering data streams based on shared density between micro-clusters. *IEEE transactions on knowledge and data engineering* 28, 6 (2016), 1449–1461.
- [24] David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. 2017. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 215–223.
- [25] Simon Haykin. 2002. Adaptive Filter Theory. *Prentice Hall google schola* 2 (2002), 67–94.
- [26] Yue He, Peng Cui, Zheyang Shen, Renzhe Xu, Furui Liu, and Yong Jiang. 2021. Daring: Differentiable causal discovery with residual independence. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 596–605.
- [27] Song Jiang, Zijie Huang, Xiao Luo, and Yizhou Sun. 2023. CF-GODE: Continuous-Time Causal Inference for Multi-Agent Dynamical Systems. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 997–1009.
- [28] Diviyani Kalainathan, Olivier Goudet, and Ritik Dutta. 2020. Causal Discovery Toolbox: Uncovering causal relationships in Python. *Journal of Machine Learning Research* 21 (2020), 37:1–37:5.
- [29] Koki Kawabata, Yasuko Matsubara, Takato Honda, and Yasushi Sakurai. 2020. Non-Linear Mining of Social Activities in Tensor Streams. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2093–2102.
- [30] Tasuku Kimura, Yasuko Matsubara, Koki Kawabata, and Yasushi Sakurai. 2022. Fast Mining and Forecasting of Co-evolving Epidemiological Data Streams. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3157–3167.
- [31] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *The Third International Conference on Learning Representations*.
- [32] Bernard O Koopman. 1931. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences* 17, 5 (1931), 315–318.
- [33] Lei Li, B Aditya Prakash, and Christos Faloutsos. 2010. Parsimonious linear fingerprinting for time series. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 385–396.
- [34] Fangfu Liu, Wenchang Ma, An Zhang, Xiang Wang, Yueqi Duan, and Tat-Seng Chua. 2023. Discovering Dynamic Causal Space for DAG Structure Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1429–1440.
- [35] Mohammad Saeid Mahdavejad, Mohammadreza Rezvan, Mohammadamin Berekatain, Peyman Adibi, Payam Barnaghi, and Amit P Sheth. 2018. Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks* 4, 3 (2018), 161–175.
- [36] Yasuko Matsubara and Yasushi Sakurai. 2016. Regime shifts in streams: Real-time forecasting of co-evolving time sequences. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1045–1054.
- [37] Yasuko Matsubara and Yasushi Sakurai. 2019. Dynamic modeling and forecasting of time-evolving data streams. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 458–468.
- [38] Yasuko Matsubara, Yasushi Sakurai, Christos Faloutsos, Tomoharu Iwata, and Masatoshi Yoshikawa. 2012. Fast mining and forecasting of complex time-stamped events. In *Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 271–279.
- [39] Jorge J Moré. 2006. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*. Springer, 105–116.
- [40] Kota Nakamura, Yasuko Matsubara, Koki Kawabata, Yuhei Umeda, Yuichiro Wada, and Yasushi Sakurai. 2023. Fast and Multi-aspect Mining of Complex Time-stamped Event Streams. In *Proceedings of the ACM Web Conference*. 1638–1649.
- [41] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.
- [42] Kohei Obata, Koki Kawabata, Yasuko Matsubara, and Yasushi Sakurai. 2024. Dynamic Multi-Network Mining of Tensor Time Series. In *Proceedings of the ACM on Web Conference*. 4117–4127.
- [43] Spiros Papadimitriou, Anthony Brockwell, and Christos Faloutsos. 2003. Adaptive, hands-off stream mining. In *Proceedings of 29th International Conference on Very Large Data Bases*. Elsevier, 560–571.
- [44] Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge university press.
- [45] Jonas Peters and Peter Bühlmann. 2015. Structural intervention distance for evaluating causal graphs. *Neural computation* 27, 3 (2015), 771–799.
- [46] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.
- [47] Jonathan G Richens, Ciarán M Lee, and Saurabh Johri. 2020. Improving the accuracy of medical diagnosis with causal machine learning. *Nature communications* 11, 1 (2020), 3923.
- [48] Jakob Runge. 2018. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28, 7 (2018).
- [49] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.
- [50] Jack Sherman and Winifred J Morrison. 1950. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics* 21, 1 (1950), 124–127.
- [51] Qiquan Shi, Jiaming Yin, Jiajun Cai, Andrzej Cichocki, Tatsuya Yokota, Lei Chen, Mingxuan Yuan, and Jia Zeng. 2020. Block Hankel tensor ARIMA for multiple short time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 5758–5766.
- [52] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. 2006. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research* 7, 10 (2006).
- [53] Peter Spirtes, Clark N Glymour, and Richard Scheines. 2000. *Causation, prediction, and search*. MIT press.
- [54] Floris Takens. 2006. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*. Springer, 366–381.
- [55] Veronica Tozzo, Federico Ciecch, Davide Garbarino, and Alessandro Verri. 2021. Statistical models coupling allows for complex local multivariate time series analysis. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1593–1603.
- [56] Dingsu Wang, Yuchen Yan, Ruizhong Qiu, Yada Zhu, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. 2023. Networked time series imputation via position-aware graph enhanced variational autoencoders. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- [57] Chenwang Wu, Xiting Wang, Defu Lian, Xing Xie, and Enhong Chen. 2023. A Causality Inspired Framework for Model Interpretation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.
- [58] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*.
- [59] Bin Yang. 1995. Projection approximation subspace tracking. *IEEE Transactions on Signal processing* 43, 1 (1995), 95–107.
- [60] Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji. 2021. DAGs with no curl: An efficient DAG structure learning approach. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 12156–12166.
- [61] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting?. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [62] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. 2018. Dags with no tears: Continuous optimization for structure learning. In *Advances in neural information processing systems*, Vol. 31. 9492–9503.
- [63] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. 2020. Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*.

## Appendix

### A Optimization Algorithm

---

**Algorithm 1** MODEPLAIT ( $x(t_c), \mathcal{F}, C$ )

---

```

1: Input: (a) New value  $x(t_c)$  at time point  $t_c$ 
           (b) Full parameter set  $\mathcal{F} = \{\Theta, \Omega\}$ 
           (c) Model candidate  $C = \{\theta^c, \omega^c, s_{en}^c\}$ 
2: Output: (a) Updated full parameter set  $\mathcal{F}'$ 
            (b) Updated model candidate  $C'$ 
            (c)  $l_s$ -steps-ahead future value  $v(t_c + l_s)$ 
            (d) Causal adjacency matrix  $B$ 
3: /* Update current window  $X^c$  */
4:  $X^c \leftarrow X[t_m : t_c]$ 
5: /* Estimate optimal regime  $\theta^*$  */
6:  $\{\mathcal{F}', C'\} \leftarrow \text{MODEESTIMATOR}(X^c, \mathcal{F}, C)$ 
7: /* Forecast future value and discover causal relationship */
8:  $\{v(t_c + l_s), B\} \leftarrow \text{MODEGENERATOR}(C')$ 
9: /* Update regime  $\theta^*$  */
10: if NOT create new regime then
11:    $C' \leftarrow \text{REGIMEUPDATER}(X^c, C')$ 
12: end if
13: return  $\{\mathcal{F}', C', v(t_c + l_s), B\}$ 

```

---



---

**Algorithm 2** MODEESTIMATOR ( $X^c, \mathcal{F}, C$ )

---

```

1: Input: (a) Current window  $X^c$ 
           (b) Full parameter set  $\mathcal{F}$ 
           (c) Model candidate  $C$ 
2: Output: (a) Updated full parameter set  $\mathcal{F}'$ 
            (b) Updated model candidate  $C'$ 
3: /* Calculate optimal initial conditions */
4:  $S_0^c \leftarrow \arg \min_{S_0^c} f(X^c; S_0^c, \theta^c)$ 
5: if  $f(X^c; S_0^c, \theta^c) > \tau$  then
6:   /* Find better regime in  $\Theta$  */
7:    $\{S_0^c, \theta^c\} \leftarrow \arg \min_{S_0^c, \theta \in \Theta} f(X^c; S_0^c, \theta^c)$ 
8:   if  $f(X^c; S_0^c, \theta^c) > \tau$  then
9:     /* Create new regime */
10:     $\{\theta^c, \omega^c\} \leftarrow \text{REGIMECREATION}(X^c)$ 
11:     $\Theta \leftarrow \Theta \cup \theta^c; \Omega \leftarrow \Omega \cup \omega^c$ 
12:   end if
13: end if
14:  $\mathcal{F}' \leftarrow \{\Theta, \Omega\}; C' \leftarrow \{\theta^c, \omega^c, S_{en}^c\}$ 
15: return  $\mathcal{F}', C'$ 

```

---

#### A.1 Details of Eq. (5)

Here, we introduce the recurrence relation of transition matrix  $A_{(i)}$ . As mentioned earlier, we use the following cost function (below, index  $i$  denoting  $i$ -th dimension is omitted for the sake of simplicity, e.g., we write  $A_{(i)}$  as  $A$ ):

$$\begin{aligned} \mathcal{E} &= \sum_{t'=t_m+h}^{t_c} \mu^{t_c-t'} \|g(e(t')) - A g(e(t'-1))\|_2^2 \\ &= \sum_{l=1}^h (L(l, :) - A(l, :)) M (L(l, :) - A(l, :)) R^\top \end{aligned}$$

where,  $M$ ,  $L$  and  $R$  are synonymous with the definition in Section 4.1. Because we want to obtain  $A$  that minimizes this cost function  $\mathcal{E}$ , we differentiate it with respect to  $A$ .

$$\frac{\partial}{\partial A(l, :)} \mathcal{E} = -2(L(l, :) - A(l, :)) R M R^\top$$

Solving the equation  $\partial \mathcal{E} / \partial A(l, :) = 0$  for each  $l$ ,  $1 \leq l \leq h$ , the optimal solution for  $A$  is given by

$$A = (L M R^\top) (R M R^\top)^{-1}$$

where we define

$$Q = L M R^\top, \quad P = (R M R^\top)^{-1}$$

The recurrence relations of  $Q$  can be written as

$$\begin{aligned} Q &= \sum_{t'=t_m+h}^{t_c} \mu^{t_c-t'} g(e(t')) g(e(t'-1))^\top \\ &= \mu \sum_{t'=t_m+h}^{t_c-1} \mu^{t_c-t'-1} g(e(t')) g(e(t'-1))^\top + g(e(t_c)) g(e(t_c-1))^\top \end{aligned}$$

$$\therefore Q^{new} = \mu Q^{prev} + g(e(t_c-1)) g(e(t_c))^\top \quad (6)$$

and similarly

$$P^{new} = (\mu P^{prev} + g(e(t_c)) g(e(t_c))^\top)^{-1} \quad (7)$$

Here, we apply the Sherman-Morrison formula [50] to the RHS of Eq. (7). Note that  $g(e(t_c))^\top P^{prev} g(e(t_c)) > 0$  because  $P^{-1} = R M R^\top$  is positive definite by definition.

$$\therefore P^{new} = \frac{1}{\mu} \left( P^{prev} - \frac{P^{prev} g(e(t_c-1)) g(e(t_c-1))^\top P^{prev}}{\mu + g(e(t_c-1))^\top P^{prev} g(e(t_c-1))} \right) \quad (8)$$

Finally, combining Eq. (6) and Eq. (8) gives the recurrence relations of  $A$  for Eq. (5).

$$A^{new} = A^{prev} + (g(e(t_c)) - A^{prev} g(e(t_c-1))) \gamma$$

$$\gamma = \frac{g(e(t_c-1))^\top P^{prev}}{\mu + g(e(t_c-1))^\top P^{prev} g(e(t_c-1))}$$

#### A.2 Proof of Lemma 1

**PROOF.** The dominant steps in REGIMECREATION are I, IV, and VI. The decomposition  $X$  into  $W^{-1}$  and  $E$  using ICA requires  $O(d^2N)$ . For each observation, the SVD of  $R_{(i)} M$  requires  $O(h^2N)$ , and the eigendecomposition of  $\tilde{A}_{(i)}$  takes  $O(k_i^3)$ . The straightforward way to process IV and VI is to perform the calculation  $d$  times sequentially, i.e., they require  $O(dh^2N + \sum_i k_i^3)$  in total. However, since these operations do not interfere with each other, they are simultaneously computed by parallel processing. Therefore, the time complexity of REGIMECREATION is  $O(N(d^2 + h^2) + k^3)$ , where  $k = \max_i(k_i)$ .  $\square$

#### A.3 Proof of Lemma 2

**PROOF.** First, we need to formulate the causal structure. Here, we utilize the structural equation model [44], denoted by  $X_{sem} = B_{sem} X_{sem} + E_{sem}$ . Because this model is known as the general formulation of causality, if  $B_{sem}$  in this model is identified, then it can be said that we discover causality. In other words, we need to prove that our proposed algorithm can find the causal adjacency matrix  $B$  aligning with this model. Solving the structural equation model

for  $X_{\text{sem}}$ , we obtain  $X_{\text{sem}} = W_{\text{sem}}^{-1} E_{\text{sem}}$  where  $W_{\text{sem}} = I - B_{\text{sem}}$ . It is shown that we can identify  $W_{\text{sem}}$  in the above equation by ICA, except for the order and scaling of the independent components, if the observed data is a linear, invertible mixture of non-Gaussian independent components [11]. Thus, demonstrating that MODEGENERATOR precisely resolves the two indeterminacies of a mixing matrix  $W^{-1}$  (i.e., the inverse of  $W \in \theta^c$ ) suffices to complete the proof because  $W$  is computed by ICA in REGIMECREATION.

First, we reveal that our algorithm can resolve the order indeterminacy. We can permute the causal adjacency matrix  $B$  to strict lower triangularity thanks to the acyclicity assumption [6]. Thus, correctly permuted and scaled  $W$  is a lower triangular matrix with all ones on the diagonal. It is also said that there would only be one way to permute  $W$ , which meets the above condition [52]. Thus, MODEGENERATOR can identify the order of a mixing matrix by the process in step I (i.e., finding the permutation of rows of a mixing matrix that yields a matrix without any zeros on the main diagonal). Next, with regard to the scale of indeterminacy, it is apparent that we only need to focus on the diagonal element, remembering that the permuted and scaled  $W$  has all ones on the diagonal. Therefore, we prove that MODEGENERATOR can resolve the order and scaling of the indeterminacies of a mixing matrix  $W^{-1}$ .  $\square$

#### A.4 Proof of Lemma 3

**PROOF.** For each time point, MODEPLAIT first runs MODEESTIMATOR, which estimates the optimal full parameter set  $\mathcal{F}$  and the model candidate  $C$  for the current window  $X^c$ . If the current regime  $\theta^c$  fits well, it takes  $O(N \sum_i k_i)$  time. Otherwise, it takes  $O(RN \sum_i k_i)$  time to find a better regime in  $\Theta \in \mathcal{F}$ . Furthermore, if MODEPLAIT encounters an unknown pattern, it runs REGIMECREATION, which takes  $O(N(d^2 + h^2) + k^3)$  time. Subsequently, it runs MODEGENERATOR to identify the causal adjacency matrix and forecast an  $l_s$ -steps-ahead future value, which takes  $O(d^2)$  and  $O(l_s)$  time, respectively. Note that  $l_s$  is negligible because of the small constant value. Finally, when MODEPLAIT does not create a new regime, it executes REGIMEUPDATER, which needs  $O(dh^2)$  time. Thus, the total time complexity is at least  $O(N \sum_i k_i + dh^2)$  time and at most  $O(RN \sum_i k_i + N(d^2 + h^2) + k^3)$  time per process.  $\square$

## B Experimental Setup

In this section, we describe the experimental setting in detail. We conducted all our experiments on an Intel Xeon Platinum 8268 2.9GHz quad core CPU with 512GB of memory and running Linux. We normalized the values of each dataset based on their mean and variance (z-normalization). The length of the current window  $N$  was 50 steps in all experiments.

**Generating the Datasets.** We randomly generated synthetic multivariate data streams containing multiple clusters, each of which exhibited a certain causal relationship. For each cluster, the causal adjacency matrix  $B$  was generated from a well-known random graph model, namely Erdős-Rényi (ER) [17] with edge density 0.5 and the number of observed variables  $d$  was set at 5. The data generation process was modeled as a structural equation model [44], where each value of the causal adjacency matrix  $B$  was sampled from a uniform distribution  $\mathcal{U}(-2, -0.5) \cup (0.5, 2)$ . In addition, to demonstrate the time-changing nature of exogenous variables, we allowed

the inherent signals variance  $\sigma_{i,t}^2$  (i.e.,  $e_{(i)}(t) \sim \text{Laplace}(0, \sigma_{i,t}^2)$ ) to change over time. Specifically, we introduced  $h_{i,t} = \log(\sigma_{i,t}^2)$ , which evolves according to an autoregressive model, where the coefficient and noise variance of the autoregressive model were sampled from  $\mathcal{U}(0.8, 0.998)$  and  $\mathcal{U}(0.01, 0.1)$ , respectively.

The overall data stream was then generated by constructing a temporal sequence of cluster segments and each segment had 500 observations (e.g., “1, 2, 1” consists of three segments containing two types of causal relationships and its total sample size is 1, 500). We ran our experiments on five different temporal sequences: “1, 2, 1”, “1, 2, 3”, “1, 2, 2, 1”, “1, 2, 3, 4”, and “1, 2, 3, 2, 1” to encompass various types of real-world scenarios.

**Baselines.** The details of the baselines we used throughout our extensive experiments are summarized as follows:

#### (1) Causal discovering methods

- CASPER [34]: is a state-of-the-art method for causal discovery, integrating the graph structure into the score function and reflecting the causal distance between estimated and ground truth causal structure. We tuned the parameters by following the original paper setting.
- DARING [26]: introduces an adversarial learning strategy to impose an explicit residual independence constraint for causal discovery. We searched for three types of regularization penalties  $\{\alpha, \beta, \gamma\} \in \{0.001, 0.01, 0.1, 1.0, 10\}$ .
- NoCurl [60]: uses a two-step procedure: initialize a cyclic solution first and then employ the Hodge decomposition of graphs. We set the optimal parameter presented in the original paper.
- NOTEARS-MLP [63]: is an extension of NOTEARS [62] (mentioned below) for nonlinear settings, which aims to approximate the generative structural equation model by MLP. We used the default parameters provided in authors’ codes<sup>2</sup>.
- NOTEARS [62]: is a differentiable optimization method with an acyclic regularization term to estimate a causal adjacency matrix. We used the default parameters provided in authors’ codes<sup>2</sup>.
- LiNGAM [52]: exploits the non-Gaussianity of data to determine the direction of causal relationships. It has no parameters to set.
- GES [10]: is a traditional score-based bayesian algorithm that discovers causal relationships in a greedy manner. It has no parameters to set. We employed BIC as the score function and utilized the open-source in [28].

#### (2) Time series forecasting methods

- TimesNet/PatchTST [41, 58]: are state-of-the-art TCN-based and Transformer-based methods, respectively. The past sequence length was set as 16 (to match the current window length). Other parameters followed the original paper setting.
- DeepAR [49]: models probabilistic distribution in the future, based on RNN. We built the model with 2-layer 64-unit RNNs. We used Adam optimization [31] with a learning rate of 0.01 and let it learn for 20 epochs with early stopping.
- OrbitMap [37]: finds important time-evolving patterns for stream forecasting. We determined the optimal transition strength  $\rho$  to minimize the forecasting error in training.
- ARIMA [7]: is one of the traditional time series forecasting approaches based on linear equations. We determined the optimal parameter set using AIC.

<sup>2</sup><https://github.com/xunzheng/notears>