Time Series Domain Adaptation via Latent Invariant Causal Mechanism

Ruichu Cai, Junxian Huang, Zhenhui Yang, Zijian Li, Emadeldeen Eldele, Min Wu, Fuchun Sun, *Fellow, IEEE*

Abstract—Time series domain adaptation aims to transfer the complex temporal dependence from the labeled source domain to the unlabeled target domain. Recent advances leverage the stable causal mechanism over observed variables to model the domain-invariant temporal dependence. However, modeling precise causal structures in high-dimensional data, such as videos, remains challenging. Additionally, direct causal edges may not exist among observed variables (e.g., pixels). These limitations hinder the applicability of existing approaches to real-world scenarios. To address these challenges, we find that the high-dimension time series data are generated from the low-dimension latent variables, which motivates us to model the causal mechanisms of the temporal latent process. Based on this intuition, we propose a latent causal mechanism identification framework that guarantees the uniqueness of the reconstructed latent causal structures. Specifically, we first identify latent variables by utilizing sufficient changes in historical information. Moreover, by enforcing the sparsity of the relationships of latent variables, we can achieve identifiable latent causal structures. Built on the theoretical results, we develop the Latent Causality Alignment (**LCA**) model that leverages variational inference, which incorporates an intra-domain latent sparsity constraint for latent structure reconstruction and an inter-domain latent sparsity constraint for domain-invariant structure reconstruction. Experiment results on eight benchmarks show a general improvement in the domain-adaptive time series classification and forecasting tasks, highlighting the effectiveness of our method in real-world scenarios. Codes are available at https://github.com/DMIRLAB-Group/LCA.

Index Terms—Time Series Data, Latent Causal Mechanism, Domain Adaptation

1 Introduction

To overcome the challenges of distribution shift between the training and the test time series datasets [1], time series domain adaptation seeks to transfer the temporal dependence from labeled source data to unlabeled target data. Mathematically, in the context of time series domain adaptation, we let $X = (\mathbf{x}_1, \cdots, \mathbf{x}_\tau, \cdots, \mathbf{x}_t)$ be multivariate time series with t timestamps and a channel size of n, where $\mathbf{x}_\tau \in \mathbb{R}^n$ and Y is the corresponding label, such that Y can be a time series or scalar, depending on the forecasting or classification tasks. By considering $\mathbf{u}$ as the domain label, we further assume that $P(X, Y|\mathbf{u}^S)$ and $P(X, Y|\mathbf{u}^T)$ are the source and target distributions, respectively. In the source domain, we can access m^S annotated X-Y pairs, represented as $(X^S, Y^S) = (X_i^S, Y_i^S)_{i=1}^{m^S}$. While in the target domain, only m_T unannotated time series data can be observed, denoted by $(X^T) = (X_i^T)_{i=1}^{m^T}$. The primary goal of unsupervised time series domain adaptation is to model the target joint distribution $P(X, Y|\mathbf{u}^T)$ by leveraging the labeled source and the unlabeled target data.

Several methods have been proposed to address this problem. Previous works have extended the conventional assumptions of domain adaptation for static data [2], [3] to time series data, including covariate shift [4], label shift [5], and conditional shift [6]. For instance, leveraging the covariate shift assumption, i.e., $P(Y|X)$ is stable, some researchers [7], [8] employ recursive neural network-based architecture as feature extractor and adopt adversarial training strategy to extract domain-invariant information. Others, e.g., Wilson et al. [9], utilize adversarial and contrastive learning to extract the domain-invariant representation for time series data. Moreover, other researchers assume that the distribution shift of Y varies across different domains, known as label shift. Specifically, He et al. [10] tackle the feature shift and label shift by aligning both temporal and frequency features. And Hoyez et al. [11] argue that conditional shift can manifest either as changes in both the input and output signals, or as changes in the relationships among multiple time series variables, where $P(X|Y)$ varies across domains in time series domain adaptation.

Instead of introducing assumptions from a statistical perspective, another promising direction is to harness the invariant temporal dependencies for time series domain

- *Ruichu Cai is with the School of Computer Science, Guangdong University of Technology, Guangzhou, China, 510006 and Peng Cheng Laboratory, Shenzhen, China, 518066. Email: cairuichu@gmail.com*
- *Junxian Huang is with the School of Computer Science, Guangdong University of Technology, Guangzhou China, 510006.Email: huangjunxian459@gmail.com*
- *Zhenhui Yang is with the School of Computing, Guangdong University of Technology, Guangzhou China, 510006.E-mail: yangzhenhui8@gmail.com*
- *Zijian Li is with the Machine Learning Department, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi.Email: leizigin@gmail.com*
- *Emadeldeen Eldele is with the Department of Computer Science, Khalifa University, Abu Dhabi, UAE, and the Institute for Infocomm Research, A*STAR, Singapore Email: emad0002@ntu.edu.sg*
- *Min Wu is with the Institute for Infocomm Research, A*STAR, Singapore. E-mail: wumin@i2r.a-star.edu.sg*
- *Fuchun Sun is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: fcsun@tsinghua.edu.cn*

This research was supported in part by National Science and Technology Major Project (2021ZD0111501), National Science Fund for Excellent Young Scholars (62122022), Natural Science Foundation of China (U24A20233, 62206064, 62206061, 62476163, 62406078), Guangdong Basic and Applied Basic Research Foundation (2023B1515120020). (Min Wu is the Corresponding author.)

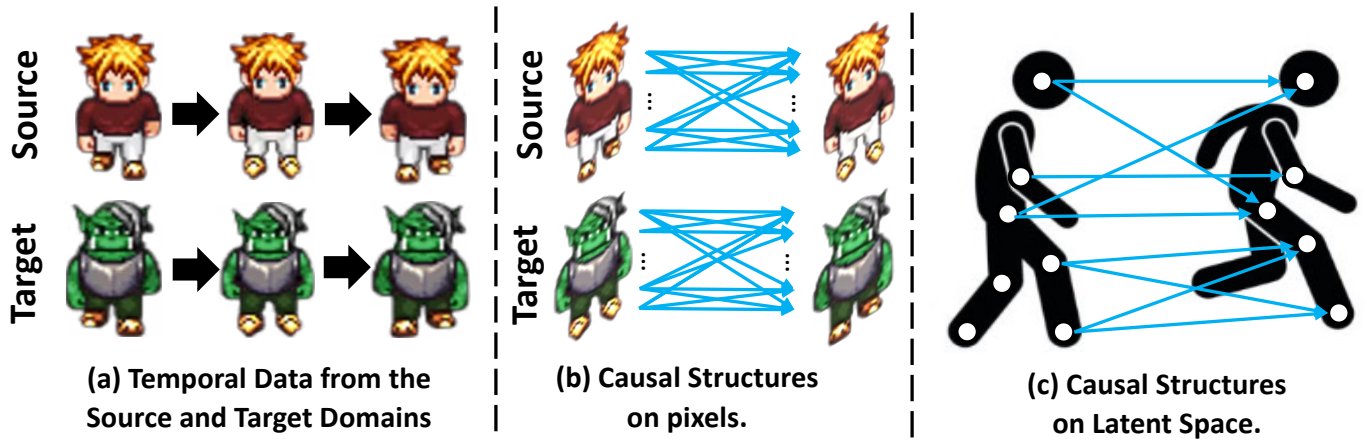


Figure 1: A toy domain adaptation example for video data, where the blue arrows denote the estimated causal relationships. (a) Two three-frame videos, one featuring a walking human and the other a monster, represent the source and target domains, respectively. (b) Since the skeleton variables are latent confounders and there are no causal relationships among pixels, the existing methods may learn dense and fault causal relationships among adjacent two frames, failing to extract the domain-invariant causal mechanism. (c) By identifying the latent variables like the joints in the skeleton, it is convenient to model the latent invariant causal mechanisms behind observed variables. Hence, we can address the time series domain adaptation problem in complex real-world scenarios.

adaptation. Specifically, Cai et al. [12] address time series domain adaptation by aligning sparse associate structures across different domains. Li et al. [13] further extend this approach by considering the domain-specific strengths of the domain-invariant sparse associative structures. Recently, Wang et al. [14] treat each sensor stream as one dimension of the time series. They construct sequential graphs to capture both sensor features and sensor correlations, and then perform the local and global levels alignment to reduce the distribution shift. Building upon this, Wang et al. [15] further extend the framework by incorporating multi-graph-based high-order alignment, which explicitly models evolving and complex distributions across domains. This higher-order correlation alignment enables more robust adaptation in challenging scenarios such as Human Activity Recognition (HAR)[1]. To further utilize domain-invariant temporal causal dependencies, Li et al. [16] proposed leveraging stable causal structures over observed variables, introducing the concept of causal conditional shift.

Although existing methods [16] demonstrate promising transfer performance by extracting domain-invariant causal mechanisms, they are limited in several critical ways. First, these approaches are primarily designed for low-dimensional time series (e.g., $n \leq 20$) with direct causal edges among observed variables. Such assumptions are rarely satisfied in practice, since many real-world datasets, such as electricity load diagrams, weather measurements, or video, are inherently high-dimensional. Second, [16] implicitly assumes that no latent confounders exist behind the observed time series. Moreover, these methods provide no identification guarantees for the estimated causal structures, which significantly restricts their applicability. Moreover, in high-dimensional settings, direct causal relationships may not exist at the observed level. For example, Figure 1(a) shows a toy three-frame video, where the walking human and monster represent the source and target domains. As illustrated in Figure 1(b), methods like [16], which estimate Granger causal structures over observed variables, tend to produce dense and spurious connections at the pixel level even if no ground truth causal relations exist among individual pixels. Consequently, such methods fail to capture domain-invariant mechanisms and become ineffective in addressing time series domain adaptation problems in realistic high-dimensional scenarios.

To address the aforementioned problems, an intuitive solution is based on the observation that videos depicting walking motion from different domains are generated from a domain-invariant skeleton, as shown in Figure 1 (c). This insight motivates us to exploit the causal structures over latent variables for time series domain adaptation. Building on this intuition, we prove that the latent causal process can be uniquely reconstructed, i.e., achieving identifiability, with the aid of nonlinear independent component analysis.

Guided by these theoretical results, we develop a latent causality alignment model (**LCA** in short). Specifically, the proposed **LCA** model employs variational inference to model the time series data from the source and target domains, utilizing a flow-based prior estimator. Moreover, we incorporate a gradient-based sparsity constraint to discover the Granger causal structures on latent variables. Furthermore, we harness the latent causality regularization to restrict the discrepancy of the causal structures from different domains. Our method is validated on eight time-series domain adaptation datasets, including video and high-dimension electricity load diagram datasets. The impressive performance, outperforming state-of-the-art methods, demonstrates the effectiveness of our approach.

Our contributions can be summarized as follows:

- Breaking out the limitation of causality alignment on observed variables, we develop a latent causal structure alignment method for time-series domain adaptation

1. Please refer to Appendix E for more introduction of HAR and how it is related to time series domain adaptation.

from the view of causal representation learning. we are the first to employ causal representation learning to the time series domain adaptation.
- Different from previous works that leverage causality structure for time series domain adaptation, we provide formal identification guarantees for both the causal representation and the latent causal structures.
- We propose a general framework for time series domain adaptation, validated through extensive experiments on time series classification and forecasting tasks across eight datasets, which show significant performance improvements over state-of-the-art methods.

2 Related Works

2.1 Unsupervised Domain Adaptation

Unsupervised domain adaptation [3], [6], [17], [18], [19], [20] aims to transfer the knowledge from the labeled source domains to an unlabeled target domain. One mainstream solution is to learn domain-invariant representations [21]. To achieve it, researchers have adopted different directions to address this problem. For instance, Long et al. [22] propose to minimize a similarity measure, i.e., maximum mean discrepancy (**MMD**), to guide learning domain-invariant representations. To learn invariant representations, Tzeng et al. [23] further used an adaptation layer and a domain confusion loss. Other researchers assume that the conditional distributions are stable across different domains, and hence they aim to extract the label-wise domain-invariant representation [24], [25], [26]. Specifically, Xie et al. [27] constrained the label-wise domain discrepancy, and Shu et al. [28] considered that the decision boundaries should not cross high-density data regions, so they proposed the virtual adversarial domain adaptation model.

Another solution is to estimate or adapt the target joint distribution directly. Specifically, Turrisi et al. [29] leverage the technique of optimal transfer to optimize the Wasserstein distance between the estimated joint target distribution and a weighted sum of the joint source distributions; Chen et al. [30] leverage the linear projections to match the source and target joint distributions under L_2 distance. And Courty et al. [31] propose a prediction function and recover the target joint distribution by minimizing the optimal transport distance between the source and target domains. Some researchers also find it convenient to identify the target joint distribution by harnessing the causal knowledge. Specifically, by leveraging the independence between $P(Y)$ and $P(X|Y)$, Zhang et al. [2], [6] consider different types of distribution shifts, such as target shift [5], [6], [32], [33], conditional shift, and generalized target shift assumptions. By incorporating these causal priors, the target joint distribution can be further decomposed into domain-invariant and domain-variant distributions (e.g., in target shift, $P(X|Y)$ and $P(Y|U)$ are domain-invariant and domain-variant distributions, respectively). As a result, the problem of identifying the target joint distribution is decomposed into simpler subproblems. Based on this idea, Cai et al. [3] extract the disentangled semantic representations, and Petar et al. [19] further incorporate domain-specific knowledge for learning domain-invariant representations. Recently, Kong et al. [17] addressed the multi-source domain adaptation by identifying the latent variables with sufficient domains. And Li et al. [34] further relaxed the identifiability assumptions. In this paper, we identify the target joint distribution of time series data by identifying the latent variables.

2.2 Domain Adaptation on Temporal Data

In recent years, domain adaptation for time series data has garnered significant attention. Previous methods [7] adopted the backbone networks like recurrent neural networks and techniques originally designed for non-time series data to capture domain-invariant representations. For example, Purushotham et al. [8] refined this approach by employing variational recurrent neural networks [35] to enhance the extraction of domain-invariant features. However, such methods face challenges in effectively capturing domain-invariant information due to the intricate dependencies between time points. Other researchers leverage the domain-invariant relationships among observed variables for time series domain adaptation. For instance, Cai et al. [12] proposed the Sparse Associative Structure Alignment (**SASA**) method, based on the assumption that sparse associative structures among variables remain stable across domains. This method has been successfully applied to adaptive time series classification and regression tasks. Additionally, Jin et al. [36] introduced the Domain Adaptation Forecaster (**DAF**), which leverages statistical relationships from relevant source domains to improve performance in target domains. Li et al. [16] hypothesized that causal structures are consistent across domains, leading to the development of the Granger Causality Alignment (**GCA**) approach. However, these methods are limited by low-dimensional data with direct causal relations. In this paper, we address the aforementioned challenges by identifying the latent dynamics behind observed variables.

2.3 Domain Adaptation on Video Data

Our work is also related to domain adaptation on video data since video sequences provide a natural example of high-dimensional time series without direct causal relationships among observed variables. This characteristic directly aligns with our motivation of addressing domain adaptation under such challenging settings. Specifically, Luo et al. [37] focused on domain-agnostic classification using a bipartite graph network topology to model cross-domain correlations. Rather than relying on adversarial learning, Sahoo et al. [38] developed **CoMix**, an end-to-end temporal contrastive learning framework that employs background mixing and target pseudo-labels. More recently, Chen et al. [39] introduced multiple domain discriminators for multi-level temporal attentive features to achieve superior alignment, while Costa et al. [40] utilized a dual-headed deep architecture that combines cross-entropy and contrastive losses to learn a more robust target classifier. Additionally, Wei et al. [41] employed contrastive and adversarial learning to disentangle dynamic and static information in videos, leveraging shared dynamic information across domains for more accurate prediction. Different from the aforementioned methods, we consider that the video data is generated by low-dimensional latent dynamics and address

the video domain adaptation by leveraging the domain-invariant causal structure over latent variables.

2.4 Granger Causal Discovery

Several approaches have been developed to infer causal structures in time series data based on Granger causality [42], [43], [44], [45], [46], [47], [48]. Early works commonly employed vector autoregressive (**VAR**) models [49], [50] combined with sparsity-inducing regularizers such as Lasso or Group Lasso [51], [52]. More recent efforts leverage neural networks to enhance flexibility and scalability. For example, Tank et al. [42] proposed a neural autoregressive model with sparsity penalties on the network weights, while Marcinkevivcs et al. [45], motivated by the interpretability of self-explaining neural networks, introduced a generalized **VAR** framework for Granger causality. Li et al. [16] treated the Granger causal structure as a latent variable, and Cheng et al. [53], [54] designed neural algorithms for causal discovery from irregular time series. Lin et al. [47] further incorporated contrastive learning into a neural architecture to improve Granger causality inference. Despite these advances, most existing methods focus on Granger causal structures in low-dimensional observed time series and neglect the influence of latent variables, limiting their applicability to high-dimensional data or settings with hidden causal factors. To overcome this limitation, we propose to identify latent variables and infer the underlying latent Granger causal structures in high-dimensional time series.

2.5 Identifiability of Generative Model

To learn causal representation [55], [56], [57] in time series data, many studies leverage Independent Component Analysis (**ICA**) to recover latent variables with identifiability guarantees [58], [59], [60], [61]. Traditional **ICA** methods typically assume a linear mixing function from latent to observed variables [62], [63], [64], [65]. To move beyond linearity, researchers have established identifiability in nonlinear **ICA** under additional assumptions, such as the use of auxiliary variables or sparsity in the generative process [66], [67], [68], [69], [70]. Aapo et al. [71] first achieved identifiability by assuming that latent sources follow an exponential family distribution and by introducing auxiliary variables such as domain indices, time indices, or class labels. To further relax the exponential family assumption, Zhang et al. [17], [72], [73], [74] developed component-wise identifiability results for nonlinear **ICA**, requiring 2n+1 auxiliary variables for n latent variables. In parallel, to pursue identifiability in an unsupervised manner, other works exploited structural sparsity assumptions [66], [75], [76], [77]. More recently, Zhang et al. [78] established identifiability under distribution shift by leveraging sparse latent structures, and Li et al. [79] extended this idea to time series by employing sparse causal influences with instantaneous dependencies. Similar to the aforementioned works [79], [80], we achieve identifiability of latent variables by exploiting the variability of historical information.

3 PRELIMINARIES

In this section, we first describe the data generation process from multiple distributions under latent Granger Causality. Sequentially, we further provide the definition of generalized causal conditional shift as well as identifiability of latent causal process.

3.1 Data Generation Process from Multiple Distributions under Latent Granger Causality

To illustrate how we address time series domain adaptation with latent causality alignment, we first consider the data generation process under latent Granger causality. Specifically, we first let $X = (\mathbf{x}_1, \cdots, \mathbf{x}_\tau, \cdots, \mathbf{x}_t)$ be multivariate time series with t time steps. Each $\mathbf{x}_\tau \in \mathbb{R}^m$ is generated from latent variables $\mathbf{z}_\tau \in \mathbb{R}^n, m \gg n$ via an invertible nonlinear mixing function $\mathbf{g}$ as follows:

$$\mathbf{x}_t = \mathbf{g}(\mathbf{z}_t). \tag{1}$$

Moreover, the i-th dimension of the latent variables $z_{t,i}$ is generated through the latent causal process, which is assumed to be influenced by the time-delayed parent variables $\mathrm{Pa}(z_{t,i})$ and the different domains, simultaneously. Formally, this relationship can be expressed using a structural equation model as follows:

$$z_{t,i} = f_i(\mathrm{Pa}(z_{t,i}), \mathbf{u}, \epsilon_{t,i}) \quad \text{with} \quad \epsilon_{t,i} \sim p_{\epsilon_i}, \tag{2}$$

where $\epsilon_{\tau,i}$ denotes the temporally and spatially independent noise extracted from a distribution p_{ϵ_i}, and $\mathbf{u}$ denotes the domain index that could be either the source (S) or target (T) domains. Sequentially, we assume that the label Y is generated from the latent variables shown as Equation (3)

$$Y = C(\mathbf{z}_1, \cdots, \mathbf{z}_t), \tag{3}$$

where C is the labeling function. Importantly, this labeling mechanism is assumed to be independent of the domain index $\mathbf{u}$. In other words, although the latent variables $\mathbf{z}_t$ are generated under domain-specific influences (e.g., source or target domains), the rule that maps latent causal dynamics to the label remains the same across domains. This assumption is natural in many real-world scenarios. For example, in time series forecasting, the label $Y = (\mathbf{x}_{t+1}, \cdots, \mathbf{x}_{t+\rho})$ corresponds to the values of the future ρ steps, which depend on the intrinsic latent causal dynamics rather than the region from which the data are collected. In time series classification, Y denotes the discrete class labels, where the mapping from latent motion patterns (e.g., gait cycles, arm swings) to activity categories (e.g., walking, running) is invariant across domains. Under this data generation process, our goal is to leverage labeled source data and unlabeled target data to identify the target joint distribution.

3.2 Generalized Causal Conditional Shift Assumption

Inspired by the domain-invariant causal mechanism between the source and target domains, we generalize the causal conditional shift assumption [16] from observed variables to latent variables, which is discussed as follows.

Assumption 1. *(Generalized Causal Conditional Shift) Given the latent variables $\mathbf{z}_1, \cdots, \mathbf{z}_t, \mathbf{z}_{t+1}$, where $\mathbf{z}_t \in \mathbb{R}^n$, we let the latent causal structure $A \in \{0,1\}^{n \times n}$ be the latent causal structure over latent variables. Moreover, we assume that the conditional distribution $P(\mathbf{z}_{t+1}|\mathbf{z}_1, \cdots, \mathbf{z}_t)$ varies with different*

domains, while the latent causal structures are stable across different domains. This can be formalized as follows:

$$P(\mathbf{z}_{t+1}|\mathbf{z}_1,\cdots,\mathbf{z}_t,S) \neq P(\mathbf{z}_{t+1}|\mathbf{z}_1,\cdots,\mathbf{z}_t,T) \quad A_S = A_T, \tag{4}$$

where A_S and A_T are the causal structures over latent variables from the source and target domains, respectively.

Implication: Compared with the assumption in [16], our setting is more general. While [16] assumes that the domain index $\mathbf{u}$ affects only the distribution of observed variables $\mathbf{x}$ with invariant causal relations among them, we instead assume that u influences the distribution of latent variables $\mathbf{z}$, which further generate $\mathbf{x}$, while the causal structure among $\mathbf{z}$ remains invariant. This generalization is crucial for high-dimensional time series (e.g., video frames or sensor streams), where direct causal edges among observed variables are rare and causal dependencies are better captured in the latent space. The implication is that although conditional distributions of latent variables may vary across domains, the invariant latent causal structures provide a stable foundation for adaptation. For example, in climate forecasting, the distribution of future temperatures may differ between tropical and polar regions due to domain-specific exogenous factors, yet the causal relations among latent climate drivers (e.g., greenhouse gases, solar radiation, ocean circulation) remain stable. Similarly, in human activity recognition, sensor readings may vary across devices or environments, but the causal relations among latent motion primitives (e.g., leg movements driving body displacement) are domain-invariant. Hence, focusing on latent causal structures enables robust knowledge transfer across domains even when statistical distributions shift.

3.3 Identifiability of Target Joint Distribution

In this subsection, we discuss how to identify the target joint distribution. By introducing the latent variables and combining the data generation process, we can factorize the target joint distribution as shown in Equation (5).

$$P(X,Y|\mathbf{u}_T) = \int_Z P(X|Z)P(Y|Z)P(Z|\mathbf{u}_T)dZ, \tag{5}$$

where $Z = (\mathbf{z}_1,\cdots,\mathbf{z}_t)$ denotes the latent causal process. According to the aforementioned equation, we can identify the target joint distribution by modeling the conditional distribution of observed data given latent variables and identifying the latent causal process under theoretical guarantees. In particular, once the latent processes are uniquely identified up to permutation and component-wise invertible transformations (as guaranteed in Lemma 1), there remain no unobserved confounders among them. Under this setting, the causal relationships among the latent variables are identifiable (as defined in Definition 1), which is derived from Proposition 2. Therefore, the identifiability of latent processes ensures that the latent causal structure and the target joint distribution can also be recovered.

Definition 1 (Identifiable Latent Causal Process)**.** *Let $\mathbf{X} = \{\mathbf{x}_1,\ldots,\mathbf{x}_l\}$ represent a sequence of observed variables generated according to the true latent causal processes characterized by $(f_i, p(\epsilon_i), \mathbf{g})$, as described in Equations (1) and (2). A learned generative model $(\hat{f}_i, \hat{p}(\epsilon_i), \hat{\mathbf{g}})$ is said to be observationally equivalent to $(f_i, p(\epsilon_i), \mathbf{g})$ if the distribution $p_{\hat{f}_i,\hat{p}(\epsilon_i),\hat{\mathbf{g}}}(\{\mathbf{x}\}_{t=1}^l)$ produced by the model matches the true data distribution $p_{f_i,p(\epsilon_i),\mathbf{g}}(\{\mathbf{x}\}_{t=1}^l)$ for all possible values of $\mathbf{x}_t$. We say that the latent causal processes are identifiable if such observational equivalence implies that the generative model can be transformed into the true generative process by means of a permutation π and a component-wise invertible transformation $\mathcal{T}$:*

$$p_{\hat{f}_i,\hat{p}_{\epsilon_i},\hat{\mathbf{g}}}\big(\{\mathbf{x}\}_{t=1}^T\big) = p_{f_i,p_{\epsilon_i},\mathbf{g}}\big(\{\mathbf{x}\}_{t=1}^T\big) \;\Rightarrow\; \hat{\mathbf{g}} = \mathbf{g}\circ\pi\circ\mathcal{T}. \tag{6}$$

Upon the identification of the mixing process, the latent variables will be identified up to a permutation and a component-wise invertible transformation:

$$\begin{aligned}\hat{\mathbf{z}}_t &= \hat{\mathbf{g}}^{-1}(\mathbf{x}_t) = \big(\mathcal{T}^{-1}\circ\pi^{-1}\circ\mathbf{g}^{-1}\big)(\mathbf{x}_t) \\ &= \big(\mathcal{T}^{-1}\circ\pi^{-1}\big)\big(\mathbf{g}^{-1}(\mathbf{x}_t)\big) = \big(\mathcal{T}^{-1}\circ\pi^{-1}\big)(\mathbf{z}_t).\end{aligned} \tag{7}$$

4 Identifying Latent Causal Process

Based on the definition of the identification of latent causal processes, we demonstrate how to identify the latent causal process within the context of a sparse latent process. Specifically, we first utilize the connection between conditional independence and cross derivatives [81], along with the sufficient variability of temporal data, to uncover the relationships between the estimated and true latent variables. Furthermore, we establish the identifiability of the latent variables by imposing constraints on sparse causal influences, as shown in Lemma 1.

We demonstrate that given certain assumptions, the latent variables are also identifiable. To integrate contextual information for identifying latent variables $\mathbf{z}_t$, we consider latent variables across L consecutive timestamps, including $\mathbf{z}_t$. For simplicity, we analyze the case where the sequence length is 2 (i.e., $L = 2$) and the time lag is 1.

Lemma 1. ***(Identifiability of Temporally Latent Process)*** *[80] Suppose there exists invertible function $\hat{\mathbf{g}}$ that maps $\mathbf{x}_t$ to $\hat{\mathbf{z}}_t$, i.e., $\hat{\mathbf{z}}_t = \hat{\mathbf{g}}(\mathbf{x}_t)$, such that the components of $\hat{\mathbf{z}}_t$ are mutually independent conditional on $\hat{\mathbf{z}}_{t-1}$.Let*

$$\begin{aligned}\mathbf{v}_{t,k} &= \Big(\frac{\partial^2 \log p(z_{t,k}|\mathbf{z}_{t-1})}{\partial z_{t,k}\partial z_{t-1,1}}, \frac{\partial^2 \log p(z_{t,k}|\mathbf{z}_{t-1})}{\partial z_{t,k}\partial z_{t-1,2}}, \ldots, \\ &\qquad \frac{\partial^2 \log p(z_{t,k}|\mathbf{z}_{t-1})}{\partial z_{t,k}\partial z_{t-1,n}}\Big)^\top, \\ \mathring{\mathbf{v}}_{t,k} &= \Big(\frac{\partial^3 \log p(z_{t,k}|\mathbf{z}_{t-1})}{\partial z_{t,k}^2\partial z_{t-1,1}}, \frac{\partial^3 \log p(z_{t,k}|\mathbf{z}_{t-1})}{\partial z_{t,k}^2\partial z_{t-1,2}}, \ldots, \\ &\qquad \frac{\partial^3 \log p(z_{t,k}|\mathbf{z}_{t-1})}{\partial z_{t,k}^2\partial z_{t-1,n}}\Big)^\top.\end{aligned} \tag{8}$$

If for each value of $\mathbf{z}_t, \mathbf{v}_{t,1}, \mathring{\mathbf{v}}_{t,1}, \mathbf{v}_{t,2}, \mathring{\mathbf{v}}_{t,2}, \ldots, \mathbf{v}_{t,n}, \mathring{\mathbf{v}}_{t,n}$, as $2n$ vector function in $z_{t-1,1}, z_{t-1,2}, \ldots, z_{t-1,n}$, are linearly independent, then $\mathbf{z}_t$ must be an invertible, component-wise transformation of a permuted version of $\hat{\mathbf{z}}_t$.

Proof Sketch and implication. The proof can be found in Appendix A. First, we establish a bijective transformation between the ground truth $\mathbf{z}_t$ and the estimated $\hat{\mathbf{z}}_t$ to connect them. Sequentially, we leverage the historical information to construct a full-rank linear system, where the only solution

of $\frac{\partial z_{t,i}}{\partial \hat{z}_{t,j}} \cdot \frac{\partial z_{t,i}}{\partial \hat{z}_{t,k}}$ is zero. Since the Jacobian of h is invertible, for each $z_{t,i}$, there exists a h_i such that $z_{t,i} = h(\hat{z}_{t,j})$, implying that $z_{t,i}$ is component-wise identifiable. Intuitively, the validity of the linear independence assumption shown in Equation (8) means that the historical information $\mathbf{z}_{t-1}$ affects the future variables sufficiently, which can be achieved when we have sufficient time series data.

For the sake of simplicity, Lemma 1 consider only one special case with lag=1. Our identifiability results can actually be extended to arbitrary lags and subsequences easily. For any given lag=τ. In this case, the vector function $\mathbf{v}_{t,k}$ and $\mathring{\mathbf{v}}_{t,k}$ should be modified as follows:

$$\begin{aligned}
\mathbf{v}_{t,k} = \Big(& \frac{\partial^2 \log p(z_{t,k}|\mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau})}{\partial z_{t,k} \partial z_{t-1,1}}, \cdots, \\
& \frac{\partial^2 \log p(z_{t,k}|\mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau})}{\partial z_{t,k} \partial z_{t-1,n}}, ..., \\
& \frac{\partial^2 \log p(z_{t,k}|\mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau})}{\partial z_{t,k} \partial z_{t-\tau,1}}, \cdots, \\
& \frac{\partial^2 \log p(z_{t,k}|\mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau})}{\partial z_{t,k} \partial z_{t-\tau,n}} \Big)^\top, \\
\mathring{\mathbf{v}}_{t,k} = \Big(& \frac{\partial^3 \log p(z_{t,k}|\mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau})}{\partial z_{t,k}^2 \partial z_{t-1,1}}, \cdots, \\
& \frac{\partial^3 \log p(z_{t,k}|\mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau})}{\partial z_{t,k}^2 \partial z_{t-1,n}}, ..., \\
& \frac{\partial^3 \log p(z_{t,k}|\mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau})}{\partial z_{t,k}^2 \partial z_{t-\tau,1}}, \cdots, \\
& \frac{\partial^3 \log p(z_{t,k}|\mathbf{z}_{t-1}, \cdots, \mathbf{z}_{t-\tau})}{\partial z_{t,k}^2 \partial z_{t-\tau,n}} \Big)^\top.
\end{aligned} \tag{9}$$

Moreover, Granger Causality among $\mathbf{z}_t$ and $\mathbf{z}_{t-1}$ can be identified by using a sparsity constraint on the relationships among latent variables, which is shown in Proposition 2 (also refer to the proof in Appendix B for more details).

Proposition 2. ***(Identification of Granger Causality among latent variables)*** *Suppose the estimated transition function f_i accurately models the relationship between $\mathbf{z}_t$ and $\mathbf{z}_{t-1}$. Let the ground truth Granger Causal structure be represented as $\mathcal{G} = (V, E_V)$ with the maximum lag of 1, where V and E_V denote the nodes and edges, respectively. If the data are generated according to Equation (1) and (2), and sparsity is enforced among the latent variables, then $z_{t-1,j} \to z_{t,i} \notin E_V$ if and only if $\frac{\partial z_{t,i}}{\partial z_{t-1,j}} = 0$.*

Implication. This result indicates that by enforcing sparsity among latent variables, the underlying Granger causal structure can be identified. In particular, the zero partial derivative condition ensures that non-causal relations are pruned, so that only genuine temporal dependencies remain. This provides a theoretical guarantee that latent Granger causality can be learned from data through sparsity constraints, enabling the recovery of interpretable and domain-invariant causal structures.

5 Latent Causality Alignment Model

Based on the theoretical results, we propose the latent causality alignment (**LCA**), shown in Figure 2, for time series domain adaptation. This figure shows a variational-inference-based neural architecture to model time series data, a prior estimation network, and a downstream neural forecaster for different downstream tasks. Please refer to Appendix F for more details of model architecture.

5.1 Variational Autoencoder for Static Data

Before introducing the implementation of the proposed LCA model, we first provide a preliminary introduction to the variation autoencoder. Suppose we have a data generation process for static data as shown in Equation (10)

$$\mathbf{x} = \mathbf{g}(\mathbf{z}), \tag{10}$$

where $\mathbf{x}, \mathbf{z}$, and $\mathbf{g}$ denote the observed variables, latent variables, and mixing procedure, respectively. To model this data generation process, we begin by modeling the marginal distribution of observed variables and derive the evidence lower bound (ELBO) as follows:

$$\begin{aligned}
\ln P(\mathbf{x}) &= \ln \frac{P(\mathbf{x}, \mathbf{z})}{P(\mathbf{z}|\mathbf{x})} = \mathbb{E}_{Q(\mathbf{z}|\mathbf{x})} \ln \frac{P(\mathbf{x}, \mathbf{z}) Q(\mathbf{z}|\mathbf{x})}{P(\mathbf{z}|\mathbf{x}) Q(\mathbf{z}|\mathbf{x})} \\
&= \mathbb{E}_{Q(\mathbf{z}|\mathbf{x})} \frac{\ln P(\mathbf{x}|\mathbf{z}) P(\mathbf{z})}{Q(\mathbf{z}|\mathbf{x})} + D_{KL}(Q(\mathbf{z}|\mathbf{x}) || P(\mathbf{z}|\mathbf{x})) \\
&\geq \mathbb{E}_{Q(\mathbf{z}|\mathbf{x})} \ln P(\mathbf{x}|\mathbf{z}) - D_{KL}(Q(\mathbf{z}|\mathbf{x}) | P(\mathbf{z})),
\end{aligned} \tag{11}$$

where D_{KL} denotes the KL divergence. And we use the posterior distribution $Q(\mathbf{z}|\mathbf{x})$ to approximate the prior distribution of latent variables $P(\mathbf{z})$. And $P(\mathbf{x}|\mathbf{z})$ is parameterized by a decoder which models the likelihood of observed variables given latent variables.

5.2 Temporally Variational Inference Architecture

Based on the aforementioned variation inference for static data, we further extend it to the temporal data generation process shown in Section 3.1. Similarly, we first derive the ELBO to model the observed data, as follows:

$$\begin{aligned}
\ln P(X, Y) &= \ln P(\mathbf{x}_{1:t}, Y) = \ln \frac{P(\mathbf{x}_{1:t}, Y, \mathbf{z}_{1:t})}{P(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}, Y)} \\
&\geq \underbrace{\mathbb{E}_{Q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})} \ln P(\mathbf{x}_{1:t}|\mathbf{z}_{1:t})}_{L_R} + \underbrace{\mathbb{E}_{Q(\mathbf{z}_{1:t})} \ln P(Y|\mathbf{z}_{1:t})}_{L_Y} \\
&\quad \underbrace{- D_{KL}(Q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}) || P(\mathbf{z}_{1:t}))}_{L_{KL}},
\end{aligned} \tag{12}$$

where $Q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$ and $P(\mathbf{x}_{1:t}|\mathbf{z}_{1:t})$ are used to approximate the prior distribution of latent variables and reconstruct the observations, respectively. Technologically, we consider $Q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})$ and $P(\mathbf{x}_{1:t}|\mathbf{z}_{1:t})$ as the encoder and decoder networks, respectively, which can be formalized as follows:

$$\hat{\mathbf{z}}_{1:t} = \psi(\mathbf{x}_{1:t}) \quad \hat{\mathbf{x}}_{1:t} = \phi(\mathbf{z}_{1:t}), \tag{13}$$

where ψ and ϕ denote the encoder and decoder respectively.

5.3 Prior Estimation Networks

As discussed in the implication of Lemma 1, the latent variables can be identified by leveraging the properties of how historical latent variables $\mathbf{z}_{t-1}$ influence the future ones $\mathbf{z}_t$. To further harness this property in implementation, we need to model the latent dynamics of $z_{t,i} = f(\mathbf{Pa}(z_{t,i}), \epsilon_{t,i})$. By further leveraging the property of independent noise, we follow [82] and model the latent dynamic by estimating the exogenous noises and further constraining their

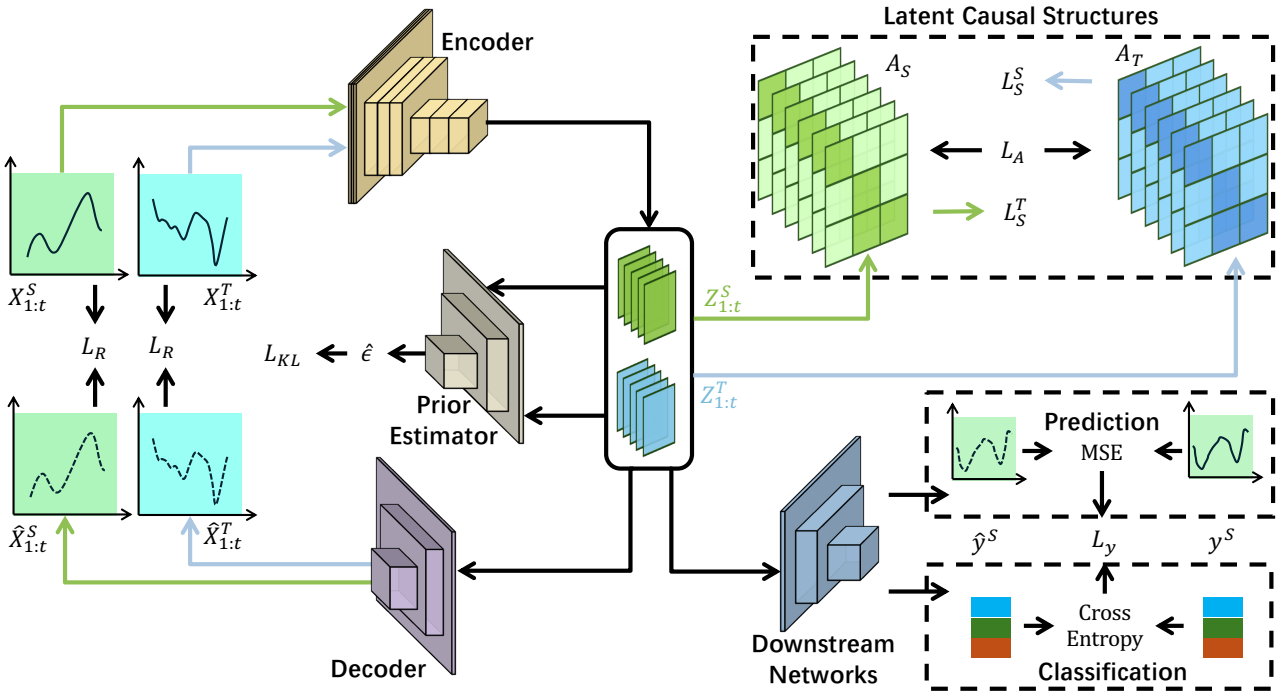


Figure 2: The illustration of the Latent Causality Alignment model. The model is built on the temporal variational inference framework, which contains an encoder and a decoder, while the prior estimator captures temporal structures via exogenous noise with KL regularization. Sparsity constraints (L_S^S, L_S^T) estimate latent Granger causal structures, and the alignment loss L_A extracts domain-invariant causality. The estimated latent variables are used for different downstream tasks.

independence. Therefore, we propose the prior estimation networks. Specifically, we first let r_i be a set of learned inverse transition functions that receive the estimated latent variables with the superscript symbol $\hat{}$ as input, and use it to estimate the noise term $\hat{\epsilon}_i$, i.e., $\hat{\epsilon}_{t,i} = r_i(\hat{\mathbf{z}}_{t,i}, \hat{\mathbf{z}}_{t-1})$, where each r_i is implemented by Multi-layer Perceptron networks (MLPs). Sequentially, we devise a transformation $\kappa := \{\hat{\mathbf{z}}_{t-1}, \hat{\mathbf{z}}_t\} \rightarrow \{\hat{\mathbf{z}}_{t-1}, \hat{\boldsymbol{\epsilon}}_t\}$, whose Jacobian can be formalized as $\mathbf{J}_\kappa = \begin{pmatrix} \mathbb{I} & 0 \\ \mathbf{J}_d & \mathbf{J}_e \end{pmatrix}$, where $\mathbf{J}_d(i,j) = \frac{\partial r_i}{\partial \hat{z}_{t-1,j}}$ and $\mathbf{J}_e$=diag$\left(\frac{\partial r_i}{\partial \hat{z}_{t,i}}\right)$. Hence we have Equation (14) via the change of variables formula.

$$\log p(\hat{\mathbf{z}}_t, \hat{\mathbf{z}}_{t-1}) = \log p(\hat{\mathbf{z}}_{t-1}, \epsilon_t) + \log |\frac{\partial r_i}{\partial z_{t,i}}|. \tag{14}$$

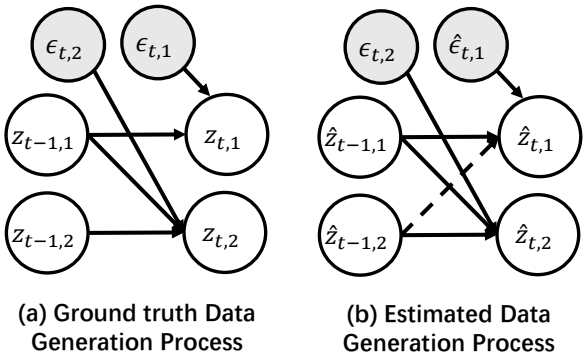
According to the generation process, the noise term $\epsilon_{t,i}$ is independent of $\mathbf{z}_{t-1}$. Therefore, we can impose independence on the estimated noise term $\hat{\epsilon}_{t,i}$. Consequently, Equation (14) can be further expressed as:

$$\log p(\hat{\mathbf{z}}_t \mid \{\hat{\mathbf{z}}_{t-\tau}\}) = \sum_{i=1}^{n} \log p(\hat{\epsilon}_{t,i}) + \sum_{i=1}^{n} \log \left| \frac{\partial r_i}{\partial \hat{z}_{t,i}} \right|. \tag{15}$$

Assuming a τ-order Markov process, the prior $p(\hat{\mathbf{z}}_{1:T})$ can be expressed as $p(\hat{z}_{1:\tau}) \prod_{t=\tau+1}^{T} p(\hat{\mathbf{z}}_t \mid \{\hat{\mathbf{z}}_{t-\tau}\})$. Hence, the log-likelihood $\log p(\hat{\mathbf{z}}_{1:T})$ can be estimated as:

$$\log p(\hat{\mathbf{z}}_{1:T}) = \log p(\hat{z}_{1:\tau}) + \sum_{t=\tau+1}^{T} \left(\sum_{i=1}^{n} \log p(\hat{\epsilon}_{t,i}) + \sum_{i=1}^{n} \log \left| \frac{\partial r_i}{\partial \hat{z}_{t,i}} \right| \right). \tag{16}$$

Here, we assume that the noise term $p(\hat{\epsilon}_{\tau,i})$ and the initial latent variables $p(\hat{\mathbf{z}}_{1:\tau})$ follow Gaussian distributions. For



(a) Ground truth Data Generation Process (b) Estimated Data Generation Process

Figure 3: To describe the relationship between partial derivatives and the existence of edges clearly.

$\tau = 1$, this simplifies to:

$$\log p(\hat{\mathbf{z}}_{1:T}) = \log p(\hat{z}_1) + \sum_{t=2}^{T} \left(\sum_{i=1}^{n} \log p(\hat{\epsilon}_{t,i}) + \sum_{i=1}^{n} \log \left| \frac{\partial r_i}{\partial \hat{z}_{t,i}} \right| \right). \tag{17}$$

Similarly, the distribution $p(\hat{\mathbf{z}}_{t+1:T} \mid \hat{\mathbf{z}}_{1:t})$ can be estimated using analogous methods. For further details on the derivation of the prior, please refer to Appendix C.

5.4 Sparsity Constraint for Latent Granger Causality

By using the prior estimation networks to estimate the independent noise, the latent variables can be identified. However, without additional constraints, it is difficult to recover the causal structures among latent variables, as the expressive power of neural networks may lead to redundant

edges. To address this issue, Proposition 2 shows that the true latent Granger causal structure can be identified if sparsity is enforced among latent variables. Motivated by this, we incorporate the sparsity constraint on partial derivatives regarding independent noise and latent variables for the latent Granger Causality, which prunes non-causal dependencies and guides the network to capture the genuine latent causal relations that are invariant across domains.

To make the intuition clearer, we illustrate with the example in Figure 3. In the ground truth process (Figure 3(a)), $z_{t,1}$ depends only on $z_{t-1,1}$ and $\epsilon_{t,1}$:

$$z_{t,1} = f_1(z_{t-1,1}, \epsilon_{t,1}). \tag{18}$$

Hence, the partial derivatives satisfy $\frac{\partial \epsilon_{t,1}}{\partial z_{t-1,1}} \neq 0$ and $\frac{\partial \epsilon_{t,1}}{\partial z_{t-1,2}} = 0$. This indicates that the Jacobian encodes the presence of the true edge $z_{t-1,1} \rightarrow z_{t,1}$ and the absence of the edge $z_{t-1,2} \rightarrow z_{t,1}$.

In the estimated process (Figure 3(b)), however, $\hat{z}_{t,1}$ is incorrectly modeled as

$$\hat{z}_{t,1} = f_1(\hat{z}_{t-1,1}, \hat{z}_{t-1,2}, \hat{\epsilon}_{t,1}), \tag{19}$$

which introduces a spurious dependence from $\hat{z}_{t-1,2}$. As a result, $\frac{\partial \hat{\epsilon}_{t,1}}{\partial \hat{z}_{t-1,2}} \neq 0$, and the Jacobian of the estimated model contains a nonzero entry corresponding to this false edge.

To remove such spurious edges, we apply an $\mathcal{L}_1$ penalty on the Jacobian:

$$L_S = \|\boldsymbol{J}\|_1, \tag{20}$$

which drives unnecessary derivatives toward zero while retaining the true nonzero ones. This enforces sparsity in the Jacobian and ensures that the recovered causal graph reflects only the genuine dependencies among latent variables.

5.5 Latent Causality Alignment Constraint

Since the latent structures are stable across different domains, we need to align the latent structures of the source domain and the target domain. However, unlike the structural alignment of the observed variables [16], the alignment of latent variables can be a more challenging task for two main reasons. First, the structure between latent variables is implicit, and the causal structure cannot be directly extracted for alignment as in GCA [16]. In addition, although the partial derivatives regarding estimated noise and latent variables can reflect the causal structure over latent variables, directly aligning the gradient can affect the correct gradient descent direction and result in suboptimal performance downstream tasks. To overcome these challenges, we propose the latent causality alignment constraint.

Specifically, we choose a threshold u to determine the causal relations among the latent variables. For instance, if $\mathbf{J}^*_{i,j} > u$, then there is an edge from $z_{t-1,i}$ to $z_{t,j}$, otherwise, no edge is present. Formally, we can obtain the estimated causal relationships of the source or target domains (the superscript * shows the source or target domains) as follows:

$$\hat{\mathbf{J}}^*_{i,j} = \begin{cases} 1, \text{ if } \mathbf{J}^*_{i,j} > u; \\ 0, \text{ otherwise.} \end{cases} \tag{21}$$

A direct solution to align the latent causal structures is to restrict the discrepancy between the latent structures of source and target domains, following [16]. However, since these causal structures are represented using gradients with respect to latent variables, the direct alignment of the gradients can interfere with the model optimization, thereby increasing the difficulty of training. To overcome these issues, we find that it is sufficient to focus on reducing the differences in $\mathbf{J}^*$ between the source and target domains while ignoring the identical parts. This approach achieves causal structure alignment while minimizing the impact on the gradients. Based on this idea, we first obtain a masking matrix through an XOR operation. In this matrix, elements with a value of 0 represent identical structures between the source and target domains, while elements with a value of 1 represent differing structures. We then constrain only the differing parts, which are formalized as follows:

$$\begin{aligned} \mathcal{M} &= (\mathbf{J}^S_d > u) \oplus (\mathbf{J}^T_d > u) \\ L_A &= ||\mathbf{G}(\mathbf{J}^S_d \odot \mathcal{M}) - \mathbf{J}^T_d \odot \mathcal{M}||_1, \end{aligned} \tag{22}$$

where $\oplus$ and $\odot$ denote the XOR and element-wise product operations, respectively, and $\mathbf{G}(\cdot)$ denotes the gradient-stopping operation [16], which is used to enforce the target latent structures closer to the source latent structures.

5.6 Model Summary

By employing the latent causality alignment constraint, the domain-invariant latent causal mechanisms are preserved. As mentioned in Section 3.1, the mapping from $\{\mathbf{z}_1, \cdots, \mathbf{z}_t\}$ to Y is not influenced by different domains, so we can directly use the latent variables for downstream tasks without considering different domain information[2]. Specifically, we leverage a Multi-layer Perceptron network $\hat{C}$ to generate predictions $\hat{Y}$ shown as follows:

$$\hat{Y} = \hat{C}(\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \cdots, \hat{\mathbf{z}}_t), \tag{23}$$

where $\hat{C}$ is optimized by source labeled data with the loss L_Y. Sequentially, by combining Equations (12) and (20), we can achieve the total loss of our method, as follows:

$$L_{total} = L_Y + \alpha L_R + \beta L_{KL} + \gamma L_S + \delta L_A, \tag{24}$$

where α, β, γ, and δ are tunable hyper-parameters.

6 EXPERIMENTS

6.1 Domain Adaptation for Time Series Forecasting

6.1.1 Datasets

In this section, we provide an overview of the five real-world datasets used to evaluate the **LCA** model. **PPG-DaLiA**[3] is a publicly available multimodal dataset, used for PPG-based heart rate estimation. It includes physiological and motion data collected from 15 volunteers using wrist and chest-worn devices during various activities. We categorize the data into four domains based on activities: Cycling (C), Sitting (S), Working (W), and Driving (D). **Human Motion**[4] is a dataset for human motion prediction and cross-domain adaptation. We select four motion types as domains: Walking (W), Greeting (G), Eating (E), and

2. Please refer to Section 6.4 for the case that domains influence the mapping from $\{\mathbf{z}_1, \cdots, \mathbf{z}_t\}$ to Y.

3. https://archive.ics.uci.edu/ml/datasets/PPG-DaLiA

4. http://vision.imar.ro/human3.6m/description.php

Table 1: MAE and MSE of various methods on the PPG-DaLiA dataset, where R-COAT, iTrans, and TMixer are the abbreviation of RAINCOAT, iTransformer, and TimeMixer, respectively.

Metric	Task	SASA	GCA	DAF	CLUDA	R-COAT	AdvSKM	iTrans	TMixer	TSLANet	SegRNN	Ours
MSE	C → D	0.7421	0.8260	0.7085	0.8869	0.9117	0.7795	0.6036	0.5901	0.6162	0.6333	**0.5797**
	C → S	0.6279	0.6898	0.5726	0.7851	0.8675	0.6998	0.3433	0.2962	0.3139	0.4259	**0.2842**
	C → W	0.8767	0.9076	0.8565	0.9709	0.9356	0.8846	0.8117	0.8162	0.8226	0.8452	**0.7946**
	D → C	0.9415	0.9858	0.9258	1.0562	1.0134	0.9539	0.8603	0.8476	0.8623	0.8599	**0.8294**
	D → S	0.3643	0.4976	0.3751	0.5719	0.4645	0.4179	0.3121	0.3012	0.3128	0.3668	**0.2697**
	D → W	0.8666	0.9197	0.8542	0.9844	0.9270	0.9055	0.8365	0.8211	0.8335	0.8366	**0.7989**
	S → C	0.9613	0.9568	0.9224	1.0834	1.1211	0.9459	0.8552	0.8360	0.8569	0.8571	**0.8212**
	S → D	0.5708	0.6306	0.5756	0.8250	0.7340	0.6727	0.5855	0.5909	0.6178	0.5936	**0.5358**
	S → W	0.8660	0.9002	0.8439	1.0098	1.0181	0.8852	0.8310	0.8128	0.8358	0.8283	**0.7999**
	W → C	0.9302	0.9786	0.8965	0.9321	0.9396	0.8992	0.8690	0.8340	0.8532	0.8768	**0.8171**
	W → D	0.7634	0.7800	0.6983	0.8647	0.8124	0.7736	0.6010	0.5997	0.6287	0.6408	**0.5577**
	W → S	0.7244	0.6561	0.5982	0.7484	0.8592	0.6459	0.3365	0.3000	0.3238	0.4535	**0.2807**
MAE	C → D	0.5963	0.6565	0.5817	0.7061	0.7147	0.6371	0.5007	0.4836	0.5001	0.5306	**0.4678**
	C → S	0.5139	0.5520	0.4817	0.6354	0.6982	0.5663	0.3300	0.2656	0.3044	0.3956	**0.2373**
	C → W	0.6453	0.6750	0.6350	0.7214	0.6983	0.6658	0.6121	0.6138	0.6124	0.6354	**0.5898**
	D → C	0.6723	0.6927	0.6597	0.7578	0.7384	0.6820	0.6172	0.6128	0.6237	0.6334	**0.5942**
	D → S	0.3620	0.4326	0.3695	0.5130	0.4442	0.4032	0.2782	0.2736	0.2784	0.3465	**0.2264**
	D → W	0.6484	0.6733	0.6372	0.7295	0.6983	0.6780	0.6116	0.6066	0.6182	0.6214	**0.5886**
	S → C	0.6980	0.6787	0.6858	0.7816	0.7972	0.6918	0.6137	0.6060	0.6139	0.6279	**0.5920**
	S → D	0.4814	0.5169	0.4861	0.6663	0.5941	0.5675	0.4750	0.4745	0.4894	0.4873	**0.4381**
	S → W	0.6387	0.6583	0.6319	0.7500	0.7344	0.6760	0.6130	0.6051	0.6116	0.6094	**0.5927**
	W → C	0.6648	0.7078	0.6448	0.6921	0.6900	0.6675	0.6258	0.6134	0.6233	0.6546	**0.5911**
	W → D	0.6060	0.6321	0.5673	0.6922	0.6490	0.6252	0.5032	0.4991	0.5078	0.5391	**0.4595**
	W → S	0.5330	0.5557	0.4813	0.6283	0.6249	0.5328	0.3361	0.2858	0.3171	0.4241	**0.2447**

Table 2: MAE and MSE for various methods on the Human Motion dataset.

Metric	Task	SASA	GCA	DAF	CLUDA	R-COAT	AdvSKM	iTrans	TMixer	TSLANet	SegRNN	Ours
MSE	G → E	0.1845	0.2701	0.2210	0.7697	0.4857	0.4330	0.1061	0.0611	0.0845	0.0680	**0.0543**
	G → W	0.1666	0.2499	0.2033	0.6710	0.4078	0.3856	0.1387	0.1363	0.1354	0.1378	**0.1066**
	G → S	0.1405	0.1645	0.1655	0.5964	0.3492	0.3562	0.0630	0.0593	0.0512	0.0599	**0.0439**
	E → G	0.2220	0.2535	0.2460	0.7796	0.5953	0.5227	0.0806	0.0787	0.0930	0.0821	**0.0588**
	E → W	0.2250	0.2353	0.2264	0.6706	0.4090	0.4043	0.2412	0.1414	0.1748	0.1469	**0.1199**
	E → S	0.1173	0.1210	0.1247	0.5465	0.3794	0.3204	0.0568	0.0502	0.0456	0.0493	**0.0399**
	W → G	0.2295	0.2719	0.2754	0.8068	0.6679	0.5823	0.0992	0.0775	0.0840	0.0858	**0.0582**
	W → E	0.2183	0.2616	0.2386	0.8086	0.6266	0.5130	0.1382	0.0741	0.0605	0.0965	**0.0566**
	W → S	0.1455	0.1506	0.1687	0.6043	0.3765	0.3481	0.0705	0.0525	0.0501	0.0712	**0.0423**
	S → G	0.1637	0.1958	0.1790	0.7551	0.5090	0.3823	0.0909	0.0767	0.0728	0.0841	**0.0637**
	S → E	0.1555	0.1716	0.1505	0.7503	0.5252	0.3510	0.0866	0.0567	0.0677	0.0794	**0.0564**
	S → W	0.2012	0.2597	0.2524	0.6825	0.4572	0.4051	0.2182	0.1514	0.1632	0.1419	**0.1274**
MAE	G → E	0.2982	0.3700	0.3307	0.6734	0.5035	0.4933	0.2159	0.1274	0.1656	0.1469	**0.1146**
	G → W	0.2816	0.3479	0.3231	0.6339	0.4431	0.4618	0.2043	0.1910	0.2069	0.1958	**0.1595**
	G → S	0.2517	0.2751	0.2714	0.5885	0.3939	0.4386	0.1450	0.0975	0.1088	0.1278	**0.0807**
	E → G	0.3200	0.3499	0.3502	0.6759	0.5531	0.5384	0.1628	0.1601	0.1890	0.1696	**0.1310**
	E → W	0.3204	0.3345	0.3301	0.6032	0.4569	0.4661	0.3211	0.1879	0.2171	0.2060	**0.1658**
	E → S	0.2383	0.2331	0.2474	0.5641	0.4390	0.4222	0.1280	0.0821	0.0898	0.0950	**0.0785**
	W → G	0.3164	0.3466	0.3612	0.6755	0.5688	0.5534	0.2058	0.1688	0.1798	0.1674	**0.1327**
	W → E	0.3138	0.3566	0.3517	0.7005	0.5800	0.5419	0.2510	0.1606	0.1221	0.1827	**0.1191**
	W → S	0.2505	0.2556	0.2738	0.5988	0.4265	0.4373	0.1517	0.1034	0.0989	0.1534	**0.0819**
	S → G	0.2783	0.3091	0.3004	0.6746	0.5065	0.4577	0.1917	0.1574	0.1521	0.1690	**0.1369**
	S → E	0.2799	0.2971	0.2784	0.6591	0.4972	0.4494	0.1812	0.1163	0.1319	0.1526	**0.1152**
	S → W	0.2946	0.3423	0.3422	0.6153	0.4572	0.4766	0.2833	0.1949	0.2007	0.1885	**0.1689**

Smoking (S). **Electricity Load Diagrams**[5] is a dataset containing electricity consumption data from 370 substations in Portugal, recorded from January 2011 to December 2014. We account for seasonal domain shifts by dividing the data into four domains based on the months: Domain 1 (January, February, March), Domain 2 (April, May, June), Domain 3 (July, August, September), and Domain 4 (October, November, December). **PEMS**[6] dataset comprises traffic speed data collected by the California Transportation Agencies over a 6-month period from January 1st, 2017 to May 31st, 2017. To account for seasonal domain shifts, we divide the data into three domains based on months: Domain 1 (January), Domain 2 (February), and Domain 3 (March). **ETT**[7] dataset describes the electric power deployment. We use the ETT-small subset, which includes data from two stations, treating each station as a separate domain.

5. https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014
6. https://pems.dot.ca.gov/
7. https://github.com/zhouhaoyi/ETDataset

6.1.2 Baselines

Here, we introduce the benchmarks for unsupervised domain adaptation in time series forecasting, including SASA [12], AdvSKM [83], DAF [36], GCA [16], CLUDA [84], and Raincoat [10]. In addition, we include approaches that integrate the Gradient Reversal Layer (GRL) with state-of-the-art time series forecasting techniques such as SegRNN [85], TSLANet [86], TimeMixer [87], and iTransformer [88], which are based on RNN, CNN, MLP, and Transformer architectures, respectively. For all the above methods, we ensure consistency by employing the same experimental settings among them.

Table 3: MAE and MSE of various methods on the ETT dataset.

Metric	Task	SASA	GCA	DAF	CLUDA	R-COAT	AdvSKM	iTrans	TMixer	TSLANet	SegRNN	Ours
MSE	1 → 2	0.2843	0.2820	0.1812	0.4097	0.3930	0.2839	0.1439	0.1306	0.1419	0.1775	**0.1023**
	2 → 1	0.8068	0.8421	0.6438	0.9711	0.8921	0.8352	0.5848	0.6620	0.6790	0.8740	**0.4395**
MAE	1 → 2	0.3977	0.4188	0.3248	0.5126	0.5083	0.4117	0.2764	0.2585	0.2751	0.3141	**0.2195**
	2 → 1	0.6686	0.6670	0.5925	0.7636	0.6796	0.6609	0.5493	0.5696	0.5693	0.6508	**0.4482**

Table 4: MAE and MSE of various methods on the PEMS dataset.

Metric	Task	SASA	GCA	DAF	CLUDA	R-COAT	AdvSKM	iTrans	TMixer	TSLANet	SegRNN	Ours
MSE	1 → 2	0.3249	0.5068	0.4412	0.4223	0.4751	0.4176	0.4134	0.3698	0.3558	0.3312	**0.2629**
	1 → 3	0.3210	0.4788	0.3806	0.4055	0.4220	0.4045	0.3695	0.3342	0.3055	0.2769	**0.2244**
	2 → 1	0.3035	0.4866	0.4356	0.4139	0.4024	0.3975	0.3499	0.3023	0.2886	0.2996	**0.2393**
	2 → 3	0.3100	0.4383	0.3694	0.3832	0.3858	0.3802	0.3682	0.2980	0.2871	0.2681	**0.2221**
	3 → 1	0.3423	0.5112	0.4492	0.4627	0.4553	0.4155	0.4064	0.3377	0.3225	0.2952	**0.2372**
	3 → 2	0.2894	0.4427	0.3820	0.3620	0.4485	0.3572	0.4083	0.3729	0.3657	0.3515	**0.2655**
MAE	1 → 2	0.3237	0.3988	0.3906	0.3585	0.4041	0.3577	0.3415	0.3398	0.3162	0.2944	**0.2392**
	1 → 3	0.3463	0.4209	0.3780	0.3715	0.4044	0.3778	0.3533	0.3393	0.3061	0.2815	**0.2327**
	2 → 1	0.3332	0.4105	0.4025	0.3727	0.3996	0.3630	0.3308	0.3200	0.2961	0.3058	**0.2445**
	2 → 3	0.3385	0.3872	0.3629	0.3532	0.3727	0.3565	0.3472	0.3139	0.3009	0.2848	**0.2327**
	3 → 1	0.3534	0.4326	0.4063	0.3980	0.4238	0.3813	0.3855	0.3311	0.3310	0.2971	**0.2425**
	3 → 2	0.3116	0.3691	0.3496	0.3163	0.3928	0.3164	0.3503	0.3343	0.3245	0.3163	**0.2433**

6.1.3 *Experimental Settings*

We consider multivariate time series forecasting across all datasets, using an input length of 30 and an output length of 10. Each dataset is partitioned into train, validation, and test sets. For robustness, we run each method three times with different random seeds and report the average performance. The model yielding the best validation results is selected, and its performance is subsequently assessed on the test set. In all experiments, we employed the ADAM optimizer [89] and used Mean Squared Error (MSE) as the loss function for prediction. We report the Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics.

6.1.4 *Quantization Result*

We conducted comparative experiments between our method and baseline models across five datasets, and the results are presented in Tables 1, 2, 3, 4, and 9 in Appendix D, respectively. The results show that our method significantly outperforms the baseline models across all metrics, particularly on the Human Motion Dataset, where causal mechanisms are more pronounced (since the relationships between human joints represent a natural causal structure). This further highlights the superiority of our approach.

Additionally, we find that combining state-of-the-art time series forecasting methods with a gradient reversal layer to address UDA in time series forecasting also yielded excellent results. This is clear in the results of iTransformer, TimeMixer, TSLAnet, and SegRNN specifically, where these methods leverage the most advanced models in their model architectures (iTransformer for Transformers, TimeMixer for MLPs, TSLAnet for CNNs, and SegRNN for RNNs). When combined with a gradient reversal layer, these methods effectively capture domain-shared features, enhancing prediction performance. Despite their strong backbones, our approach still outperforms these baselines. Unlike TimeMixer, which captures complex macro and micro temporal information by decomposing trends and seasonal information using a carefully designed MLP, our method uses a simpler MLP learning network, demonstrating our superiority.

Furthermore, existing UDA methods in the time series domain, such as DAF, CLUDA, Raincoat, and AdvSKM, exhibited relatively weaker performance, possibly due to their less optimal backbone models. However, we notice that SASA and GCA perform better than other time series UDA methods, such as DAF, CLUDA, Raincoat, and AdvSKM. This can be attributed to their consideration of sparse correlation structures among observed variables and the consistency of Granger causality structures, despite having relatively simple backbones.

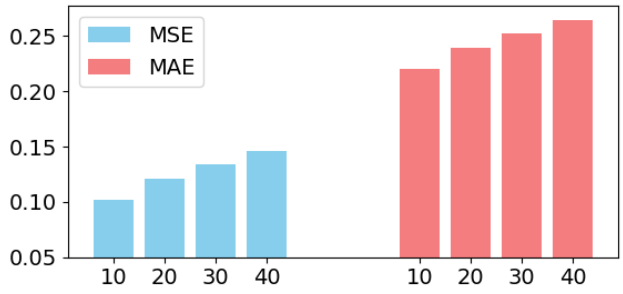


Figure 4: The MSE and MAE values after predicting different lengths in the transition from domain 1 to domain 2 in the ETT dataset. Subfigures (a), (b), (c), and (d) show the forecasting results on lengths of different time steps.

Visualization Result: To demonstrate the robustness of our method, we performed predictions of varying lengths on the ETT dataset, specifically from domain 1 to domain 2. For each input of the 30 time steps, we predicted future values over 10, 20, 30, and 40 steps. The resulting MSE and MAE values for these different forecast lengths are illustrated in Figure 10 in Appendix D. Notably, our method consistently outperforms most baseline methods that only predict 10 steps, particularly for longer forecast horizons (20, 30, and 40 steps), highlighting the superior performance of our approach. Additionally, we visualize the predictions for the last dimension of the ETT dataset, as shown in Figure 4. The visualization confirms that our method effectively captures the temporal variations in the data.

6.2 Domain Adaptation for Time Series Classification

In classification tasks, we validate the proposed method on the UCIHAR and HHAR datasets, following AdaTime [90] framework. Furthermore, we validate the performance of our method on the high-dimensional video classification datasets like UCF101 and HMDB51.

6.2.1 Datasets

Time Series Datasets: We consider the UCIHAR [91] dataset, which comprises sensor data from accelerometers, gyroscopes, and body sensors collected from 30 subjects performing six activities: walking, walking upstairs, walking downstairs, standing, sitting, and lying down. Due to the variability between individuals, each subject is treated as a separate domain. We also consider the HHAR [92] dataset, which contains sensor readings from smartphones and smartwatches, collected from 9 subjects, with each subject similarly treated as an individual domain.

Video Datasets: We adopted two widely recognized and frequently used datasets, i.e., UCF101 and HMDB51. The UCF101 dataset, curated by the University of Central Florida, consists of videos primarily sourced from YouTube, capturing a wide variety of daily activities and sports actions. In contrast, the HMDB51 dataset, collected by the University of Massachusetts Amherst, includes videos from movies, public databases, and YouTube, offering a more diverse range of real-world action scenes. Both datasets serve as key benchmarks for evaluating video classification algorithms. For data processing, we followed the approach used in TranSVAE, extracting relevant and overlapping action categories from both UCF101 and HMDB51. This resulted in a combined dataset of 3,209 videos, where UCF101 provided 1,438 training videos and 571 validation videos, and HMDB51 contributed 840 training videos and 360 validation videos. Based on these datasets, we defined two video-based unsupervised domain adaptation tasks: U $\rightarrow$ H and H $\rightarrow$ U.

6.2.2 Baselines

Since the time series classification UDA methods use constraints that can be only applied to discrete labels, we selected several state-of-the-art methods implemented in the AdaTime framework as baselines for comparison. These methods include AdvSKM [83], CODATS [93], CoTMix [94], DANN [95], DDC [23], DeepCoral [96], DIRT [28], MMDA [97], and SASA [12].

For the video classification task, we conducted a comprehensive comparison based on the experimental settings of the latest TranSVAE approach [41]. Specifically, we included five widely-used image-based UDA methods, which perform domain adaptation by disregarding temporal information: DANN [95], JAN [22], ADDA [98], AdaBN [99], and MCD [38]. Additionally, we benchmarked our approach against the latest state-of-the-art video-based UDA methods, including TA3N [100], SAVA [101], TCoN [102], ABG [37], CoMix [38], CO2A [40], and MA2L-TD [39].

6.2.3 Experimental Settings

For each of the time series datasets, we randomly generated 10 transfer directions. Each experiment was run three times and the average F1-score is reported. We generally followed the validation methods provided by AdaTime benchmark.

For video classification, we strictly followed the data processing procedure used in TranSVAE. TranSVAE employs a pretrained I3D model on the Kinetics dataset as the backbone to extract features from each video frame, which are stored and subsequently used as input to the network. To maintain consistency, we directly downloaded these precomputed features from TranSVAE and used them as input for our method. For the comparison methods, we referred to the experimental data from the TranSVAE network, where classification accuracy was recorded. Consistent with the experimental settings for prediction tasks, we employed the Adam optimizer. All experiments were implemented in PyTorch and conducted on a single NVIDIA GTX 3090 GPU with 24GB of memory.

6.2.4 Quantization and Visual Results

Here, we provide a detailed analysis of our method's performance on classification tasks. Tables 5 and 6 present the performance of our proposed method alongside several baseline methods on the UCIHAR and HHAR time series classification datasets. It is worth noting that AdvSKM, CoDATS, DANN, and DIRT are adversarial-based methods, while DDC, Deep-CORAL, and MMDA methods align domains by minimizing a distance metric. On the other hand, CoTMix utilizes contrastive learning, while SASA enforces the consistency of sparse correlation structures to align domains.

The experimental results on the UCIHAR dataset show that our approach achieves the highest F1-scores across multiple cross-domain scenarios (e.g., 18 $\rightarrow$ 14, 6 $\rightarrow$ 13, 20 $\rightarrow$ 9), reaching a perfect score of 1 in some of them, significantly outperforming other comparative models such as AdvSKM, CoDATS, CoTMix, and DANN. Moreover, our model demonstrates exceptional stability across all tasks, further validating its robustness in temporal classification tasks. On the other temporal classification dataset, HHAR, our method remains the best-performing approach. For the cross-domain task 0 $\rightarrow$ 5, our method achieved an F1-score of 0.7899, significantly outperforming other comparative models, such as CoDATS (0.5714), MMDA (0.5444), and CoTMix (0.5064). Additionally, our method consistently achieves the best results across other tasks, including 3 $\rightarrow$ 1, 7 $\rightarrow$ 4, and 5 $\rightarrow$ 8.

Table 7 presents a comparative analysis of our method against the baseline on the UCF-HMDB video classification dataset. We notice that methods utilizing the I3D backbone [103] generally outperform those using ResNet-101. Our method consistently exceeds the performance of all prior approaches, achieving an average accuracy of 94.28%. These results highlight the exceptional performance of our approach even on high-dimensional data.

TranSVAE represents the previous state-of-the-art method, modeling videos as being generated by both dynamic and static latent variables, and has demonstrated remarkable performance in video unsupervised domain adaptation (UDA). Similarly, our approach models the data generation process where videos are a specific instance from a causal perspective. It assumes that the underlying data generation mechanisms remain consistent across different

Table 5: F1-scores of various methods on the UCIHAR time series dataset.

Task	AdvSKM	CoDATS	CoTimix	DANN	DDC	DeepCoral	DIRT	MMDA	SASA	Ours
18 → 14	0.9773	0.8988	0.9232	0.7729	0.9804	0.9902	1.0000	0.9869	0.9869	**1.0000**
6 → 13	0.9864	0.9932	0.9386	0.7064	0.9931	1.0000	1.0000	0.9778	0.9935	**1.0000**
20 → 9	0.3835	0.5649	0.5527	0.3885	0.3730	0.5871	0.5458	0.5356	0.4860	**0.6946**
7 → 18	0.7826	0.7813	0.8429	0.4479	0.7604	0.8246	0.8501	0.7723	0.7414	**0.9108**
19 → 11	0.6765	0.9890	0.9388	0.5348	0.6999	0.9823	0.8651	0.9386	0.8341	**0.9963**
17 → 18	0.8742	0.8730	0.9011	0.4884	0.8767	0.9507	0.8820	0.9191	0.7520	**0.9601**
9 → 19	0.4269	0.6956	0.8939	0.6113	0.3869	0.7467	0.8290	0.6182	0.6000	**0.9692**
2 → 12	0.9633	0.8191	0.8775	0.6701	0.9857	0.9886	0.9859	0.9963	1.0000	**1.0000**
12 → 3	0.9609	0.9744	0.9904	0.5566	0.9678	0.9968	0.9266	0.9936	0.9936	**1.0000**
17 → 14	0.8804	0.7389	0.8423	0.4021	0.8435	0.8572	0.7738	0.9062	0.8694	**0.9673**

Table 6: F1-scores of various methods on the HHAR time series dataset.

Task	AdvSKM	CoDATS	CoTMix	DANN	DDC	DeepCoral	DIRT	MMDA	SASA	Ours
3 → 1	0.9077	0.9582	0.9601	0.8925	0.9120	0.9731	0.9727	0.9472	0.9747	**0.9802**
7 → 4	0.8477	0.9207	0.9391	0.9587	0.8245	0.8990	0.8932	0.9274	0.9216	**0.9723**
2 → 0	0.7286	0.6855	0.6447	0.7594	0.7390	0.7426	0.7474	0.7835	0.7398	**0.8058**
5 → 7	0.5064	0.8987	0.9218	0.9186	0.5028	0.8234	0.8540	0.8200	0.8200	**0.9378**
0 → 5	0.4140	0.5714	0.5064	0.4410	0.4138	0.4365	0.4794	0.5444	0.5055	**0.7899**
3 → 5	0.8653	0.9701	0.9704	0.9799	0.8828	0.9317	0.9595	0.9750	0.9696	**0.9824**
0 → 2	0.5332	0.6743	0.7588	0.7083	0.5723	0.6217	0.6673	0.7239	0.6980	**0.8028**
5 → 8	0.9279	0.9796	0.9739	0.9915	0.8242	0.9818	0.9776	0.9874	0.9868	**0.9927**
1 → 6	0.6978	0.9047	0.9313	0.9353	0.6160	0.8751	0.9089	0.9144	0.8890	**0.9415**
7 → 3	0.9155	0.9338	0.9637	0.9547	0.8912	0.9314	0.9379	0.9575	0.9397	**0.9688**

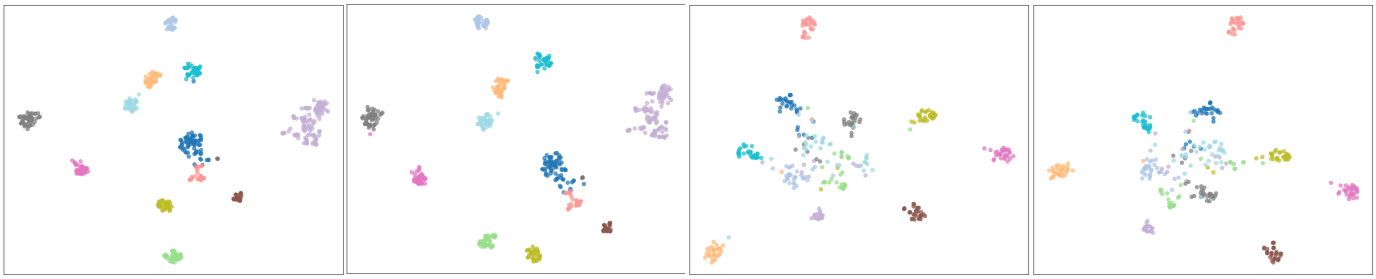


Figure 5: Classification Visualization: (a) and (b) show scatter plots of the latent variables and probability vectors, respectively, after dimensionality reduction using t-SNE for the H→U scenario. (c) and (d) present the corresponding scatter plots for the U→H scenario. These visualizations illustrate how the model's latent representations and probability vectors effectively capture and distinguish between categories.

domains. This causal modeling approach, combined with advanced deep learning architectures, has also delivered impressive results, reinforcing the potential of modeling problems from a causal perspective and leveraging deep learning networks to fit the data generation process.

Additionally, we visualize the classification results for both H→U and U→H scenarios. Specifically, we reduced the dimensionality of the model's output probability vectors or latent variable representations using t-SNE and depicted the results in scatter plots shown in Figure 5. The visualizations reveal that both the latent variables and probability vectors effectively distinguish between categories.

6.3 Ablation Study

To rigorously evaluate the contribution of each component in the proposed loss function, we devised four model variants, described as follows.

- **LCA-L_{KL}**: We remove L_{KL}, a critical term in the ELBO.
- **LCA-L_R**: We remove L_R, another essential term in the ELBO.
- **LCA-L_S**: We remove L_S, which enforces sparsity in the causal structure.

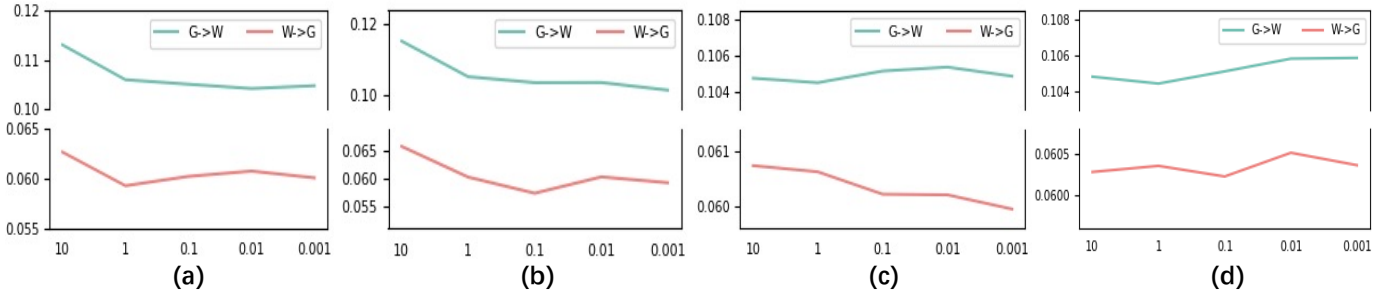


Figure 6: Sensitivity analysis of the parameters α, β, γ, and δ (from Equation 24) is shown in (a), (b), (c), and (d), respectively. The x-axis represents parameter values, and the y-axis shows the MSE value.

Table 7: Classification accuracy on the HMDB-UCF video dataset under ResNet-101 and I3D backbone networks.

Backbone	Method	U → H	H → U	Average
ResNet-101	DANN	75.28	76.36	75.82
	JAN	74.72	76.69	75.71
	AdaBN	72.22	77.41	74.82
	MCD	73.89	79.34	76.62
	TA3N	78.33	81.79	80.06
	ABG	79.17	85.11	82.14
	TCoN	87.22	89.14	88.18
	MA2L-TD	85	86.59	85.8
I3D	DANN	80.83	88.09	84.46
	ADDA	79.17	88.44	83.81
	TA3N	81.38	90.54	85.96
	SAVA	82.22	91.24	86.73
	CoMix	86.66	93.87	90.22
	CO2A	87.78	95.79	91.79
	TranSVAE	87.78	98.95	93.37
I3D	Ours	**89.44**	**99.12**	**94.28**

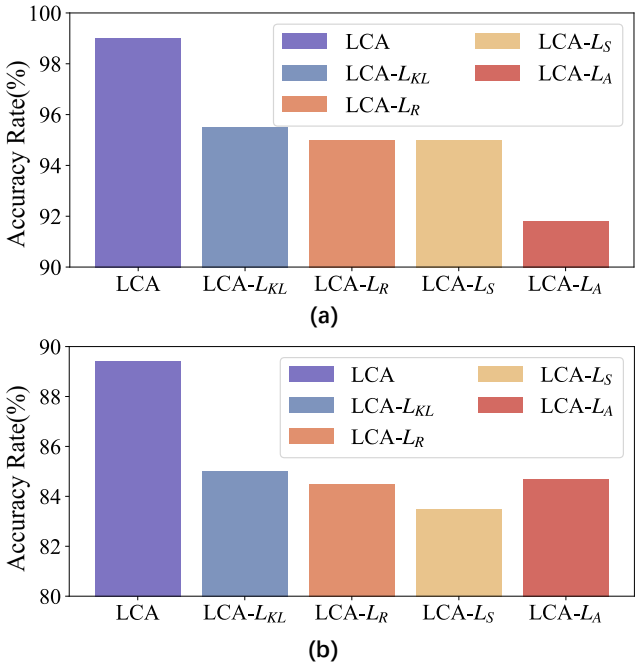


Figure 7: Ablation Results Visualization: (a) presents the results for the H→U scenario, while (b) shows the results for the U→H scenario. These visualizations illustrate the impact of different components on the performance of our model across the two scenarios.

- **LCA-L_A**: We remove L_A, which ensures consistency of causal structures across different domains.

We conducted the ablation experiments on the high-dimensional video datasets, HMDB, and UCF. In these experiments, we systematically set the weight of each corresponding loss term to zero to observe the performance of the resulting model variants. The results are shown in Figure 7. Experiment results show that each loss component plays a crucial role in model performance. Specifically, $\mathcal{L}_{KL}$ and $\mathcal{L}_R$ are vital elements of the ELBO, and maximizing them enables the model to accurately learn the joint distribution of the data. From a theoretical standpoint, $\mathcal{L}_{KL}$ ensures that the learned distribution of the latent variables closely approximates the true distribution, while $\mathcal{L}_R$ ensures that the latent variables capture as much information from the observed variables as possible. The term $\mathcal{L}_S$ enforces the sparsity of relationships between adjacent latent variables, a requirement rooted in identifiability theory. This constraint aligns with real-world scenarios, where causal relationships are typically sparse, and it helps prevent the model from learning redundant associations. Lastly, $\mathcal{L}_A$ aligns the causal mechanisms in the target domain with those in the source domain. The substantial improvement in performance observed in the experiments validates the importance of enforcing consistency in causal mechanisms across domains, thereby further affirming the soundness of our approach.

6.4 Results on Semi-supervised Domain Adaptation

According to the introduction of the data generation process in Section 3.1, the mapping from latent variables $\mathbf{z}_t$ to labeled is not influenced by different domains, so that we can use the target latent variables to generate target labels. However, suppose different domains influence the aforementioned, it is hard to achieve an ideal adaptation performance since the mapping $(\mathbf{z}_1, \cdots, \mathbf{z}_t), \mathbf{u}_T \rightarrow Y$ is hard to estimate without extra target labeled data.

Table 8: Semi-supervised results on the PPG-DaLiA dataset.

Metric	Task	SASA	GCA	DAF	CLUDA	R-COAT	AdvSKM	iTrans	TMixer	TSLANet	SegRNN	Ours
MSE	C → D	0.6463	0.6436	0.6673	0.7121	0.7102	0.6273	0.6284	0.6181	0.6290	0.6282	**0.5418**
	C → S	0.3087	0.3001	0.3054	0.6589	0.6308	0.5896	0.3613	0.2975	0.3135	0.4233	**0.2680**
	C → W	0.8392	0.8332	0.8193	0.8740	0.8770	0.8171	0.8677	0.8234	0.8436	0.8329	**0.7667**
	D → C	0.8335	0.8974	0.8955	0.9003	0.8921	0.8638	0.8983	0.8357	0.8771	0.8492	**0.8200**
	D → S	0.2933	0.2982	0.3120	0.4863	0.4470	0.3882	0.3955	0.2983	0.3080	0.3037	**0.2532**
	D → W	0.8195	0.8651	0.8216	0.8870	0.8559	0.8256	0.8831	0.8242	0.8453	0.8338	**0.7431**
	S → C	0.8723	0.9006	0.8815	0.9084	0.9103	0.8642	0.8916	0.8362	0.8649	0.8448	**0.8174**
	S → D	0.6655	0.6098	0.6336	0.6332	0.6519	0.5552	0.6285	0.6212	0.6245	0.6245	**0.5300**
	S → W	0.8427	0.8578	0.8193	0.8555	0.9146	0.8172	0.8709	0.8242	0.8451	0.6406	**0.7686**
	W → C	0.8326	0.8564	0.8810	0.8749	0.8564	0.8533	0.8938	0.8349	0.8698	0.8519	**0.8148**
	W → D	0.6362	0.6350	0.6587	0.7279	0.7055	0.6220	0.6398	0.6187	0.6272	0.6351	**0.5461**
	W → S	0.2951	0.2923	0.3071	0.5959	0.7142	0.5188	0.3797	0.2975	0.3120	0.3374	**0.2737**
MAE	C → D	0.4987	0.4971	0.5038	0.5928	0.5762	0.5326	0.5002	0.4812	0.4921	0.5147	**0.4463**
	C → S	0.2391	0.2336	0.2349	0.5342	0.5209	0.4960	0.3161	0.2401	0.2773	0.3729	**0.2266**
	C → W	0.6049	0.6057	0.6001	0.6589	0.6574	0.6164	0.6222	0.6020	0.6089	0.6045	**0.5750**
	D → C	0.5911	0.6172	0.6137	0.6678	0.6612	0.6420	0.6371	0.5984	0.6215	0.5985	**0.5901**
	D → S	0.2289	0.2296	0.2364	0.4385	0.4030	0.3622	0.3218	0.2471	0.2493	0.2333	**0.2118**
	D → W	0.5974	0.6155	0.6003	0.6686	0.6452	0.6207	0.6385	0.6028	0.6093	0.6039	**0.5596**
	S → C	0.6104	0.6172	0.6089	0.6790	0.6820	0.6490	0.6271	0.5989	0.6091	0.5969	**0.5886**
	S → D	0.5059	0.4725	0.4902	0.5446	0.5388	0.4824	0.4981	0.4866	0.4884	0.4830	**0.4368**
	S → W	0.6072	0.6141	0.6000	0.6547	0.6720	0.6281	0.6233	0.6038	0.6109	0.5148	**0.5729**
	W → C	0.5904	0.6022	0.6094	0.6497	0.6370	0.6257	0.6288	0.5969	0.6125	0.6001	**0.5889**
	W → D	0.4918	0.4938	0.5001	0.6042	0.5799	0.5271	0.5083	0.4844	0.4890	0.5089	**0.4486**
	W → S	0.2305	0.2301	0.2345	0.5025	0.4829	0.4434	0.3353	0.2440	0.2655	0.2418	**0.2284**

To solve this problem, we allow a few labeled samples from the target domain and further evaluate the performance of the proposed method under the setting of semi-supervised domain adaptation. Specifically, since the mapping from $\mathbf{z}_t$ to Y are influenced by $\mathbf{u}$, we follow [16] and employ the trainable parameters to encode the domain-specific information, which can be formalized as follows:

$$\hat{Y} = \hat{C}(\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \cdots, \hat{\mathbf{z}}_t, \mathbf{u}), \tag{25}$$

which is optimized by labeled source and target data.

Based on this modification, we evaluate our method on PPG-DaLiA datasets in the semi-supervised domain adaptation setting. Experiment results are shown in 8. We can find that our method can still achieve the best performance in the semi-supervised setting.

6.5 Sensitivity Analysis

We performed a sensitivity analysis on the parameters α, β, γ, and δ in the loss function (Equation 24), focusing on the transfer directions from G to W and W to G in the Human Motion dataset. We set several values for each parameter, ranging from 10 to 0.001 and decreasing by a factor of 10 each time. The experimental results are shown in Figure 6. Notably, as the parameters decrease, the MSE exhibits a trend of initially decreasing and then increasing, which matches our expectations: when the parameters are large, the constraints on the corresponding modules are strongest, causing the model to over-focus on certain areas and leading to a decline in performance. Conversely, when these parameters are small, the corresponding modules receive insufficient attention, resulting in suboptimal performance.

6.6 Model Performance on Different Time Lags

To investigate the impact of time lag on model performance, we extended our framework to larger lag settings and conducted experiments with lag=2 and lag=3 across multiple transfer directions on the ETT, Human Motion, and PPG-DaLiA datasets. The results are summarized in Figure 8.

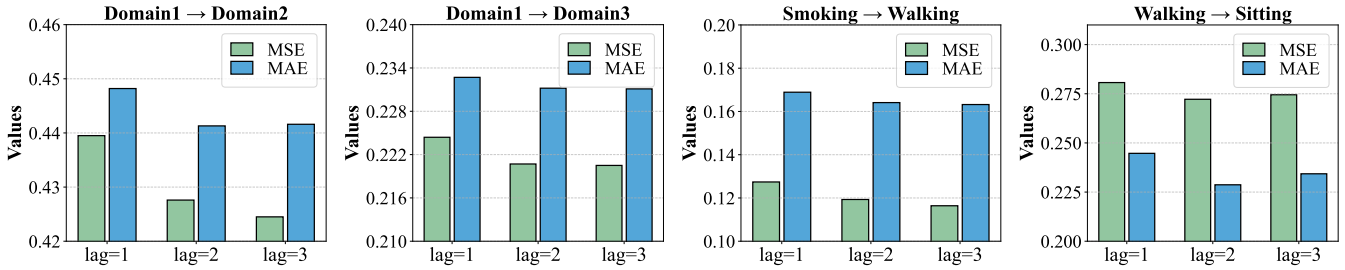
According to the experiment results, we observe a consistent trend: as the lag increases from 1 to 2, the performance improves significantly in terms of both MSE and MAE. This indicates that incorporating additional historical information helps the model capture more informative temporal dependencies. When the lag further increases to 3, the improvements become smaller and the performance tends to stabilize. This suggests that the model has already exploited sufficient past information by lag 3, and longer histories provide diminishing returns.

6.7 Model Efficiency

To validate the usability of our approach, we further provide the model computational implication comparison from the perspectives of training time, MSE and accuracy performance, and memory footprint, which are shown in Figure 9. Compared with other approaches for time series domain adaptation, the proposed LCA model achieves a significantly higher performance while maintaining competitive training time. Although our method requires more GPU memory than methods such as TA3N and GCA, it substantially outperforms them in terms of overall performance.

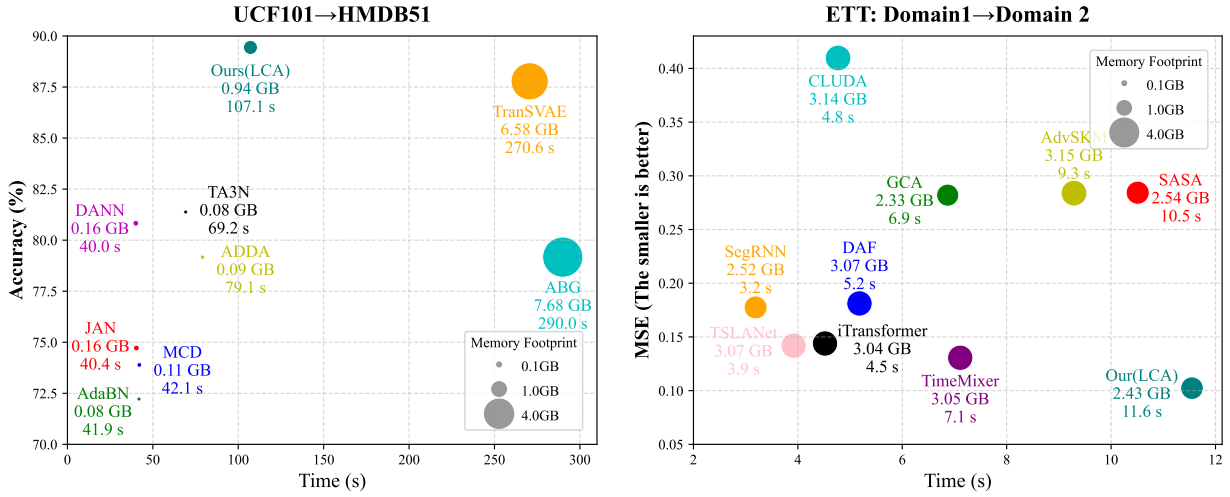
7 Conclusion

In this paper, we propose a Latent Causality Alignment (**LCA**) model for time series domain adaptation. By identifying low-dimensional latent variables and reconstructing their causal structures with sparsity constraints, we effectively transfer domain knowledge through aligning the causal relationships among latent variables across domains. The proposed method not only addresses the challenges of modeling latent causal mechanisms in high-dimensional time series data but also guarantees the uniqueness of latent causal structures, providing improved performance on



(a) Dataset ETT (b) Dataset PEMS (c) Dataset Human Motion (d) Dataset PPG-DaLiA

Figure 8: Performance comparison with different lag settings (lag=1,2,3) across different datasets. Results are reported on (a) ETT with transfer directions $1 \rightarrow 2$, (b) PEMS with transfer directions $1 \rightarrow 3$, (c) Human Motion (Walking) with transfer direction $Smoking \rightarrow Walking$, and (d) PPG-DaLiA (Sitting) with transfer direction $Walking \rightarrow Sitting$.



(a) Dataset HMDB-UCF (b) Dataset ETT

Figure 9: Model efficiency comparison of different methods on the datasets HMDB-UCF and ETT. Training time and memory footprint are recorded with the same batch size.

domain-adaptive time series classification and forecasting tasks. However, our method assumes sufficient changes in historical information for latent variable identification. Exploring how to relax this assumption and extend our framework to more complex scenarios would be a promising direction for future work.

REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[2] K. Zhang, M. Gong, and B. Schölkopf, "Multi-source domain adaptation: A causal view," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[3] R. Cai, Z. Li, P. Wei, J. Qiao, K. Zhang, and Z. Hao, "Learning disentangled semantic representation for domain adaptation," in *IJCAI: proceedings of the conference*, vol. 2019. NIH Public Access, 2019, p. 2060.

[4] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation." *Journal of Machine Learning Research*, vol. 8, no. 5, 2007.

[5] Z. Lipton, Y.-X. Wang, and A. Smola, "Detecting and correcting for label shift with black box predictors," in *International conference on machine learning*. PMLR, 2018, pp. 3122–3130.

[6] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *International Conference on Machine Learning*. PMLR, 2013, pp. 819–827.

[7] P. R. d. O. da Costa, A. Akçay, Y. Zhang, and U. Kaymak, "Remaining useful lifetime prediction via deep domain adaptation," *Reliability Engineering & System Safety*, vol. 195, p. 106682, 2020.

[8] S. Purushotham, W. Carvalho, T. Nilanon, and Y. Liu, "Variational recurrent adversarial deep domain adaptation," in *International Conference on Learning Representations*, 2022, pp. 295–309.

[9] G. Wilson, J. R. Doppa, and D. J. Cook, "Calda: Improving multi-source time series domain adaptation with contrastive adversarial learning," *IEEE transactions on pattern analysis and machine intelligence*, 2023.

[10] H. He, O. Queen, T. Koker, C. Cuevas, T. Tsiligkaridis, and M. Zitnik, "Domain adaptation for time series under feature and label shifts," in *International Conference on Machine Learning*. PMLR, 2023, pp. 12 746–12 774.

[11] H. Hoyez, B. Mirbach, J. Rambach, C. Schockaert, and D. Stricker, "Which time series domain shifts can neural networks adapt to?"

[12] R. Cai, J. Chen, Z. Li, W. Chen, K. Zhang, J. Ye, Z. Li, X. Yang, and Z. Zhang, "Time series domain adaptation via sparse associative structure alignment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6859–6867.

[13] Z. Li, R. Cai, J. Chen, Y. Yan, W. Chen, K. Zhang, and J. Ye, "Time-series domain adaptation via sparse associative structure alignment: Learning invariance and variance," *Neural Networks*, vol. 180, p. 106659, 2024.

[14] Y. Wang, Y. Xu, J. Yang, Z. Chen, M. Wu, X. Li, and L. Xie, "Sensor alignment for multivariate time-series unsupervised domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 10 253–10 261.

[15] Y. Wang, Y. Xu, J. Yang, M. Wu, X. Li, L. Xie, and Z. Chen,

"Sea++: Multi-graph-based higher-order sensor alignment for multivariate time-series unsupervised domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, 2024.

[16] Z. Li, R. Cai, T. Z. Fu, Z. Hao, and K. Zhang, "Transferable time-series forecasting under causal conditional shift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[17] L. Kong, S. Xie, W. Yao, Y. Zheng, G. Chen, P. Stojanov, V. Akinwande, and K. Zhang, "Partial disentanglement for domain adaptation," in *International conference on machine learning*. PMLR, 2022, pp. 11 455–11 472.

[18] C. Shui, Z. Li, J. Li, C. Gagné, C. X. Ling, and B. Wang, "Aggregating from multiple target-shifted sources," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 18–24 Jul 2021, pp. 9638–9648.

[19] P. Stojanov, Z. Li, M. Gong, R. Cai, J. Carbonell, and K. Zhang, "Domain adaptation with invariant representation learning: What transformations to learn?" *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[20] J. Wen, N. Zheng, J. Yuan, Z. Gong, and C. Chen, "Bayesian uncertainty matching for unsupervised domain adaptation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 3849–3855.

[21] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," *Advances in neural information processing systems*, vol. 29, 2016.

[22] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*. PMLR, 2017, pp. 2208–2217.

[23] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, 2014.

[24] C. Chen, Z. Chen, B. Jiang, and X. Jin, "Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3296–3303.

[25] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 627–636.

[26] G. Kang, L. Jiang, Y. Wei, Y. Yang, and A. Hauptmann, "Contrastive adaptation network for single-and multi-source domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 4, pp. 1793–1804, 2020.

[27] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *International conference on machine learning*. PMLR, 2018, pp. 5423–5432.

[28] R. Shu, H. Bui, H. Narui, and S. Ermon, "A DIRT-t approach to unsupervised domain adaptation," in *International Conference on Learning Representations*, 2018, pp. 3602–3620.

[29] R. Turrisi, R. Flamary, A. Rakotomamonjy, and M. Pontil, "Multi-source domain adaptation via weighted joint distributions optimal transport," in *Uncertainty in artificial intelligence*. PMLR, 2022, pp. 1970–1980.

[30] S. Chen, M. Harandi, X. Jin, and X. Yang, "Domain adaptation by joint distribution invariant projections," *IEEE transactions on image processing*, vol. 29, pp. 8264–8277, 2020.

[31] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," *Advances in neural information processing systems*, vol. 30, 2017.

[32] M. Roberts, P. Mani, S. Garg, and Z. Lipton, "Unsupervised learning under latent label shift," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 763–18 778, 2022.

[33] J. Wen, R. Greiner, and D. Schuurmans, "Domain aggregation networks for multi-source domain adaptation," in *International conference on machine learning*. PMLR, 2020, pp. 10 214–10 224.

[34] Z. Li, R. Cai, G. Chen, B. Sun, Z. Hao, and K. Zhang, "Subspace identification for multi-source domain adaptation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[35] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *Advances in neural information processing systems*, vol. 28, 2015.

[36] X. Jin, Y. Park, D. Maddix, H. Wang, and Y. Wang, "Domain adaptation for time series forecasting via attention sharing," in *International Conference on Machine Learning*. PMLR, 2022, pp. 10 280–10 297.

[37] Y. Luo, Z. Huang, Z. Wang, Z. Zhang, and M. Baktashmotlagh, "Adversarial bipartite graph learning for video domain adaptation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 19–27.

[38] A. Sahoo, R. Shah, R. Panda, K. Saenko, and A. Das, "Contrast and mix: Temporal contrastive video domain adaptation with background mixing," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 386–23 400, 2021.

[39] P. Chen, Y. Gao, and A. J. Ma, "Multi-level attentive adversarial learning with temporal dilation for unsupervised video domain adaptation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1259–1268.

[40] V. G. T. Da Costa, G. Zara, P. Rota, T. Oliveira-Santos, N. Sebe, V. Murino, and E. Ricci, "Dual-head contrastive domain adaptation for video action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1181–1190.

[41] P. Wei, L. Kong, X. Qu, Y. Ren, Z. Xu, J. Jiang, and X. Yin, "Unsupervised video domain adaptation for action recognition: A disentanglement perspective," *Advances in Neural Information Processing Systems*, vol. 36, pp. 17 623–17 642, 2023.

[42] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. B. Fox, "Neural granger causality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[43] C. Diks and V. Panchenko, "A new statistic and practical guidelines for nonparametric granger causality testing," *Journal of Economic Dynamics and Control*, vol. 30, no. 9-10, pp. 1647–1669, 2006.

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

[45] R. Marcinkevics and J. E. Vogt, "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, pp. 11 489–11 511.

[46] S. Löwe, D. Madras, R. Z. Shilling, and M. Welling, "Amortized causal discovery: Learning to infer causal graphs from time-series data," in *1st Conference on Causal Learning and Reasoning, CLeaR 2022, Sequoia Conference Center, Eureka, CA, USA, 11-13 April, 2022*.

[47] C.-M. Lin, C. Chang, W.-Y. Wang, K.-D. Wang, and W.-C. Peng, "Root cause analysis in microservice using neural granger causal discovery," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 206–213.

[48] C. Gong, D. Yao, C. Zhang, W. Li, J. Bi, L. Du, and J. Wang, "Causal discovery from temporal data," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 5803–5804.

[49] A. C. Lozano, N. Abe, Y. Liu, and S. Rosset, "Grouped graphical granger modeling methods for temporal causal modeling," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 577–586.

[50] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.

[51] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 68, no. 1, pp. 49–67, 2006.

[52] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[53] Y. Cheng, R. Yang, T. Xiao, Z. Li, J. Suo, K. He, and Q. Dai, "CUTS: Neural causal discovery from irregular time-series data," in *The Eleventh International Conference on Learning Representations*, 2023, pp. 23 538–23 561.

[54] Y. Cheng, L. Li, T. Xiao, Z. Li, J. Suo, K. He, and Q. Dai, "Cuts+: High-dimensional causal discovery from irregular time-series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 10, 2024, pp. 11 525–11 533.

[55] G. Rajendran, S. Buchholz, B. Aragam, B. Schölkopf, and P. K. Ravikumar, "From causal to concept-based representation learning," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[56] A. Mansouri, J. Hartford, Y. Zhang, and Y. Bengio, "Object centric architectures enable efficient causal representation learning," in *The Twelfth International Conference on Learning Representations*, 2024, pp. 42 385–42 408.

[57] L. Wendong, A. Kekić, J. von Kügelgen, S. Buchholz, M. Besserve, L. Gresele, and B. Schölkopf, "Causal component analysis," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[58] D. Yao, D. Xu, S. Lachapelle, S. Magliacane, P. Taslakian, G. Martius, J. von Kügelgen, and F. Locatello, "Multi-view causal representation learning with partial observability," in *The Twelfth International Conference on Learning Representations*, 2024, pp. 48 111–48 142.

[59] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[60] Y. Liu, A. Alahi, C. Russell, M. Horn, D. Zietlow, B. Schölkopf, and F. Locatello, "Causal triplet: An open challenge for intervention-centric causal representation learning," in *2nd Conference on Causal Learning and Reasoning (CLeaR)*, 2023.

[61] L. Gresele, P. K. Rubenstein, A. Mehrjou, F. Locatello, and B. Schölkopf, "The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 217–227.

[62] P. Comon, "Independent component analysis, a new concept?" *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[63] A. Hyvärinen, "Independent component analysis: recent advances," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20110534, 2013.

[64] T.-W. Lee and T.-W. Lee, *Independent component analysis*. Springer, 1998.

[65] K. Zhang and L. Chan, "Kernel-based nonlinear independent component analysis," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2007, pp. 301–308.

[66] Y. Zheng, I. Ng, and K. Zhang, "On the identifiability of nonlinear ica: Sparsity and beyond," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 411–16 422, 2022.

[67] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural networks*, vol. 12, no. 3, pp. 429–439, 1999.

[68] A. Hyvärinen, I. Khemakhem, and R. Monti, "Identifiability of latent-variable and structural-equation models: from linear to nonlinear," *Annals of the Institute of Statistical Mathematics*, vol. 76, no. 1, pp. 1–33, 2024.

[69] I. Khemakhem, R. Monti, D. Kingma, and A. Hyvarinen, "Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 768–12 778, 2020.

[70] Z. Li, Z. Xu, R. Cai, Z. Yang, Y. Yan, Z. Hao, G. Chen, and K. Zhang, "Identifying semantic component for robust molecular property prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[71] A. Hyvarinen and H. Morioka, "Nonlinear ica of temporally dependent stationary sources," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 460–469.

[72] S. Xie, L. Kong, M. Gong, and K. Zhang, "Multi-domain image generation and translation with identifiability guarantees," in *The Eleventh International Conference on Learning Representations*, 2022, pp. 7155–7189.

[73] L. Kong, B. Huang, F. Xie, E. Xing, Y. Chi, and K. Zhang, "Identification of nonlinear latent hierarchical models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 2010–2032, 2023.

[74] H. Yan, L. Kong, L. Gui, Y. Chi, E. Xing, Y. He, and K. Zhang, "Counterfactual generation with identifiability guarantees," in *37th International Conference on Neural Information Processing Systems, NeurIPS 2023*, 2023.

[75] I. Ng, Y. Zheng, X. Dong, and K. Zhang, "On the identifiability of sparse ica without assuming non-gaussianity," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[76] S. Lachapelle, P. Rodriguez, Y. Sharma, K. E. Everett, R. Le Priol, A. Lacoste, and S. Lacoste-Julien, "Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica," in *Conference on Causal Learning and Reasoning*. PMLR, 2022, pp. 428–484.

[77] D. Xu, D. Yao, S. Lachapelle, P. Taslakian, J. von Kügelgen, F. Locatello, and S. Magliacane, "A sparsity principle for partially observable causal representation learning," in *Forty-first International Conference on Machine Learning*, 2024.

[78] K. Zhang, S. Xie, I. Ng, and Y. Zheng, "Causal representation learning from multiple distributions: a general setting," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 60 057–60 075.

[79] Z. Li, Y. Shen, K. Zheng, R. Cai, X. Song, M. Gong, G. Chen, and K. Zhang, "On the identification of temporal causal representation with instantaneous dependence," in *The Thirteenth International Conference on Learning Representations*, 2025, pp. 3733–3772.

[80] W. Yao, G. Chen, and K. Zhang, "Temporally disentangled representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 492–26 503, 2022.

[81] J. Lin, "Factorizing multivariate function classes," *Advances in neural information processing systems*, vol. 10, 1997.

[82] K. Zhang and A. Hyvärinen, "On the identifiability of the post-nonlinear causal model," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 647–655.

[83] Q. Liu and H. Xue, "Adversarial spectral kernel matching for unsupervised time series domain adaptation." in *IJCAI*, 2021, pp. 2744–2750.

[84] Y. Ozyurt, S. Feuerriegel, and C. Zhang, "Contrastive learning for unsupervised domain adaptation of time series," in *The Eleventh International Conference on Learning Representations*, 2023, pp. 24 597–24 636.

[85] S. Lin, W. Lin, W. Wu, F. Zhao, R. Mo, and H. Zhang, "Segrnn: Segment recurrent neural network for long-term time series forecasting," *arXiv preprint arXiv:2308.11200*, 2023.

[86] E. Eldele, M. Ragab, Z. Chen, M. Wu, and X. Li, "Tslanet: Rethinking transformers for time series representation learning," in *International Conference on Machine Learning*, 2024.

[87] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. ZHOU, "Timemixer: Decomposable multiscale mixing for time series forecasting," in *International Conference on Learning Representations (ICLR)*, 2024, pp. 4166–4192.

[88] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," in *The Twelfth International Conference on Learning Representations*, 2024, pp. 4004–4028.

[89] D. Kinga, J. B. Adam *et al.*, "A method for stochastic optimization," in *International conference on learning representations (ICLR)*, vol. 5, no. 6. California;, 2015, pp. 1–11.

[90] M. Ragab, E. Eldele, W. L. Tan, C.-S. Foo, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Adatime: A benchmarking suite for domain adaptation on time series data," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 8, pp. 1–18, 2023.

[91] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.

[92] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition," in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, 2015, pp. 127–140.

[93] G. Wilson, J. R. Doppa, and D. J. Cook, "Multi-source deep domain adaptation with weak supervision for time-series sensor data," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 1768–1778.

[94] E. Eldele, M. Ragab, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Contrastive domain adaptation for time-series via temporal mixup," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 3, pp. 1185–1194, 2023.

[95] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of machine learning research*, vol. 17, no. 59, pp. 1–35, 2016.

[96] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," *Domain adaptation in computer vision applications*, pp. 153–171, 2017.

[97] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "On minimum discrepancy estimation for deep domain adaptation," *Domain Adaptation for Visual Understanding*, pp. 81–94, 2020.

[98] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[99] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognition*, vol. 80, pp. 109–117, 2018.

[100] M.-H. Chen, Z. Kira, G. AlRegib, J. Yoo, R. Chen, and J. Zheng, "Temporal attentive alignment for large-scale video domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6321–6330.
[101] J. Choi, G. Sharma, S. Schulter, and J.-B. Huang, "Shuffle and attend: Video domain adaptation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 2020, pp. 678–695.
[102] B. Pan, Z. Cao, E. Adeli, and J. C. Niebles, "Adversarial cross-domain action recognition with co-attention," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 815–11 822.
[103] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

Ruichu Cai (M'17) is currently a Professor in the School of Computer Science and the Director of the Data Mining and Information Retrieval Laboratory at Guangdong University of Technology. He received his B.S. degree in Applied Mathematics and his Ph.D. degree in Computer Science from South China University of Technology in 2005 and 2010, respectively.

His research interests include causality, deep learning, and their applications. He has received several honors and awards, including the National Science Fund for Excellent Young Scholars and the Guangdong Natural Science Award. He serves as an Associate Editor for TPAMI and an Action Editor for Neural Networks. He has also served as an Area Chair for ICML (2022–present), NeurIPS (2022–present), and ICLR (2024–present). He is a Senior Member of both the CCF and IEEE.

Junxian Huang received the BS degree in computer science from the Guangdong University of Technology,Guangzhou, China, in 2020. He is currently working toward a Master's degree with the School of Computer Science, Guangdong University of Technology.His research interests cover various topics, including machine learning on time series analysis, transfer learning, and their applications.

Zhenhui Yang graduated from Guangdong University of Technology in Guangzhou, China in 2022 with a bachelor's degree in network engineering. He is currently pursuing a master's degree in Computer Science at Guangdong University of Technology. His research interests cover a variety of different topics, including time series data prediction, transfer learning, and their applications.

Zijian Li received a B.S. degree in computer science from the Guangdong University of Technology, Guangzhou, China, in 2017. He is currently a Ph.D. student at the School of Computer Science, Guangdong University of Technology. He was a research intern at the Advanced Digital Sciences Center, Illinois at Singapore Pte in 2018.3-2018-6, and a visiting student at Nanyang Technological University in 2019.9-2019.12. His research interests cover a variety of different topics including causality-inspired machine learning, transfer learning, and their applications.

Emadeldeen Eldele is currently an assistant professor at the Department of Computer Science, Khalifa University, Abu Dhabi, UAE. He received his B.Sc. and M.Sc. degrees in Computer Engineering from the Faculty of Engineering, Tanta University, Egypt, in 2012 and 2018, respectively. He completed his Ph.D. in Computer Science and Engineering at Nanyang Technological University (NTU), Singapore, in 2023. Previously, he was a research Scientist at the Centre for Frontier AI Research (CFAR) jointly with the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. In 2023, he was awarded the 3rd Prize in the IEEE Engineering in Medicine and Biology Prize Paper Award, recognizing his contributions to the field. His research focuses on enhancing the robustness of deep learning models to tackle challenges such as domain shifts and label scarcity. His broader research interests encompass self-supervised learning, domain adaptation, and the analysis of time-series data.

Min Wu is currently a Principal Scientist at Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore. He received his Ph.D. degree in Computer Science from Nanyang Technological University (NTU), Singapore, in 2011 and B.E. degree in Computer Science from University of Science and Technology of China (USTC) in 2006. He received the best paper awards in EMBS Society 2023, IEEE ICIEA 2022, IEEE SmartCity 2022, InCoB 2016 and DASFAA 2015. He also won the CVPR UG2+ challenge in 2021 and the IJCAI competition on repeated buyers prediction in 2015. He has been serving as an Associate Editor for journals like Neurocomputing, Neural Networks and IEEE Transactions on Cognitive and Developmental Systems, as well as area chairs of leading machine learning and data mining conferences, such as ICLR, NeurIPS, AAAI, KDD, etc. His current research interests focus on AI for time series data, graph data, and biological and healthcare data. He is a senior member of the IEEE.

Fuchun Sun (IEEE Fellow) received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 1997. He is currently a Professor with the Department of Computer Science and Technology and President of Academic Committee of the Department, Tsinghua University, deputy director of State Key Lab. of Intelligent Technology and Systems, Beijing, China. His research interests include artificial intelligence, intelligent control and robotics, information sensing and processing in artificial cognitive systems, etc. He was recognized as a Distinguished Young Scholar in 2006 by the Natural Science Foundation of China. He serves as Editor-in-Chief of International Journal on Cognitive Computation and Systems, and an Associate Editor for a series of international journals including IEEE Transactions on Cognitive and Developmental Systems, IEEE Transactions on Fuzzy Systems, and IEEE Transactions on Systems, Man and Cybernetics: Systems.