CauDiTS: Causal Disentangled Domain Adaptation of Multivariate Time Series

Junxin Lu [1] **Shiliang Sun** [1 2]

Abstract

Unsupervised domain adaptation of multivariate time series aims to train a model to adapt its classification ability from a labeled source domain to an unlabeled target domain, where there are differences in the distribution between domains. Existing methods extract domain-invariant features directly via a shared feature extractor, neglecting the exploration of the underlying causal patterns, which undermines their reliability, especially in complex multivariate dynamic systems. To address this problem, we propose CauDiTS, an innovative framework for unsupervised domain adaptation of multivariate time series. CauDiTS adopts an adaptive rationale disentangler to disentangle domain-common causal rationales and domain-specific correlations from variable interrelationships. The stability of causal rationales across domains is vital for filtering domain-specific perturbations and facilitating the extraction of domain-invariant representations. Moreover, we promote the cross-domain consistency of intra-class causal rationales employing the learning strategies of causal prototype consistency and domain-intervention causality invariance. CauDiTS is evaluated on four benchmark datasets, demonstrating its effectiveness and outperforming state-of-the-art methods.

1. Introduction

Multivariate time series data play a crucial role in various domains, e.g., finance, healthcare, weather, and energy. By capturing systematic trends over dynamically changing variables, the performance of multiple applications, such as traffic volume forecasting (Bai et al., 2020; Chen et al., 2023), disease prediction (Deng et al., 2020; Purushotham et al., 2016a; Ozyurt et al., 2022), and climate monitoring (Li et al., 2021), are significantly improved. Unfortunately, domain shift is a common challenge for practical application of multivariate time series. Especially, the performance of a multivariate time series model degrades when trained and deployed across two domains with distinct data distributions and operating environments. For instance, due to the differences of latitude, topography, and land-sea distribution, weather prediction models relying on long-term multivariate time series from one location may struggle to generalize well to other locations.

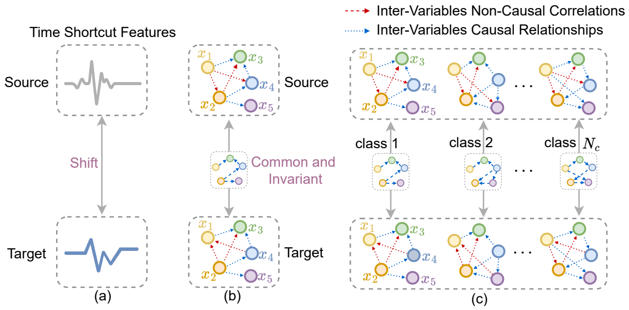


Figure 1: Across source→target domains, (a) time shortcut features show shift, (b) variable interrelationships involve non-causal correlations (--→) and causal relationships (--→). Causal relationships are domain-common and invariant, referred to as causal rationales. (c) Different classes of multivariate time series data possess distinct domain-common causal rationales.

Unsupervised domain adaptation (UDA) has been extensively investigated in vision and text tasks to alleviate the adverse impacts of domain shift (Sun & Saenko, 2016; Chen et al., 2020; Rahman et al., 2020; Zhu et al., 2020; Ganin et al., 2016). Existing UDA works for multivariate time series are comparatively few, which can be roughly categorized into two groups: (i) transferring UDA methods designed for vision (Singh, 2021; Long et al., 2015; Shu et al., 2018) and text tasks (Dai et al., 2008), or fine-tuning a pretrained model from the source domain to the target domain (Hu et al., 2020; Zhang et al., 2022; Yang & Hong, 2022); (ii) employing common backbone networks, such as recurrent neural networks (RNNs) (Purushotham et al., 2017), temporal convolutional networks (TCNs) (Bai et al., 2018),

[1] School of Computer Science and Technology, East China Normal University, Shanghai, China. [2] Department of Automation, Shanghai Jiao Tong University, Shanghai, China. Correspondence to: Shiliang Sun <slsun@cs.ecnu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

and long short-term memory (LSTM) networks (Graves & Graves, 2012) as feature extractors, coupled with adversarial training strategies (Purushotham et al., 2017; Wilson et al., 2020; Jin et al., 2022) or metric-based learning mechanisms (Liu & Xue, 2021; Cai et al., 2021), to alleviate cross-domain discrepancies in features. These two groups follow a common paradigm, aiming to discover a joint representation mapping space that is adapted to the source and target domains, facilitating the implicit extraction of domain-invariant representation.

The above-mentioned UDA works on multivariate time series have demonstrated promising performance but face a severe limitation. There exists intricate yet cross-domain stable interdependencies among variables, referred to as domain-common causal rationales. These rationales elucidate how variables generate specific class samples using invariant patterns of mutual interactions during distribution shifts, as illustrated in Figure 1(b). Previous methods prioritize extracting implicit domain-invariant representations rather than probing domain-common causal rationales. They typically align multivariate time series across domains without leveraging causal rationales to filter out domain-specific spurious information (e.g., background, biases and domain-specific styles), which can result in capturing coarse-grained shortcut features as domain-invariant representations, as depicted in Figure 1(a). Furthermore, it is worth noting that different classes of multivariate time series data exhibit distinct domain-common causal rationales, as shown in Figure 1(c). However, promoting model generalization to the unlabeled target domain by considering intra-class causality compactness remains unexplored in previous UDA works for multivariate time series.

These challenges motivate us to explore a more adaptive and robust model for UDA of multivariate time series. Different from existing works, we propose **CauDiTS**, a novel framework for **Cau**sal **Di**sentangled domain adaptation of multivariate **T**ime **S**eries. CauDiTS, grounded in a causal perspective, disentangles domain-invariant causal rationales and non-causal domain-specific correlations from variable interrelationships by an adaptive rationale disentangler. Specifically, we employ a learnable causal mask to disentangle the domain-common and domain-specific sub-graphs from the interdependencies among variables, as acquired by the variables interrelation attention module. The domain-common causal rationales play a crucial role in filtering out unstable domain-specific correlations, allowing the aggregation of domain-invariant representations from a causally augmented graph network. Incorporating non-causal domain-specific correlations facilitates effective domain discrimination in multivariate time series. Moreover, we introduce learning strategies of causal prototype consistency and domain intervention causality invariance to promote the cross-domain consistency and invariance of causal rationales. The main contributions of this paper are summarized as follows:

• We disentangle domain-common causal rationales and domain-specific non-causal correlations within the intricate variable interrelationships using an adaptive rationale disentangler. This enables CauDiTS to aggregate domain-invariant representations through causally augmented graph networks.

• We promote the cross-domain stability and invariance of intra-class causal rationales by proposing the learning strategies of causal prototypes consistency and domain intervention causality invariance.

• Experimental results on multiple datasets demonstrate that CauDiTS significantly outperforms the state-of-the-art baselines for UDA of multivariate time series.

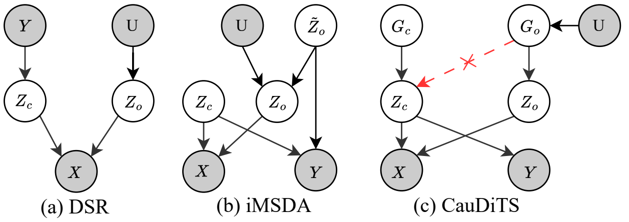
2. Related Work

Unsupervised Domain Adaptation for Time Series. In recent years, there has been a growing focus on unsupervised domain adaptation for time series classification. However, it is noteworthy that the works of the UDA for time series data are comparatively few compared to non-time series.

VRADA (Purushotham et al., 2017) is constructed on a variational recurrent neural network (Chung et al., 2015) and is trained adversarially to capture complex temporal relationships that are domain invariant. AdvSKM (Liu & Xue, 2021) performs time series domain matching by minimizing an extended version of maximum mean discrepancy (MMD) embedded in a hybrid spectral kernel network. CoDATS (Wilson et al., 2020) is based on adversarial training and employs a convolutional neural network with gradient reversal (Ganin et al., 2016) as a feature extractor. RAINCOAT (He et al., 2023) addresses feature and label shifts by combining and aligning time and frequency features, correcting misalignments, and detecting label shifts. CLUDA (Ozyurt et al., 2022) captures contextual representations between the source and target domains via customized nearest-neighbor contrastive learning, while preserving label information for prediction tasks.

The aforementioned methods implicitly extract domain-invariant features directly from a shared extractor, lacking domain-specific private information filtering, resulting in shortcut features. In contrast, causality-based CauDiTS disentangles causal rationales from non-causal domain-specific correlations. These causal rationales, which remain domain-invariant, are more trustworthy and stable than shortcut features for predicting cross-domain data labels.

Causality for Domain Adaptation. Assuming that the causality between features and classes is robust across domains, some works have explored how to utilize causal mechanisms to assist domain adaptation.



(a) DSR (b) iMSDA (c) CauDiTS

Figure 2: Different SCMs for cross-domain data generation. The gray nodes represent the observed variables, while the white nodes represent the unobserved variables. (a)-(b) are disentangled representation model for non-time series. (c) Our causal disentangled model for multivariate time series.

For non-time series data, domain adaptation methods use the structural causal model (SCM) (Pearl et al., 2016) to illustrate the data generation process, employing auto-encoder structures to reconstruct causal latent variables and domain latent variables (Cai et al., 2019; Kong et al., 2022; Wang et al., 2022; Yang et al., 2021a). For time series, SASA (Cai et al., 2021) exploits the stability of causality to propose a sparse associative structure alignment model for domain adaptation, where the alignment of associative structures guides knowledge transfer between domains. GCRL (Bagi et al., 2023) uses causality to improve the identifiability and robustness of existing out-of-distribution motion forecasting models. GCA (Li et al., 2023) addresses the tasks of cross-domain time-series forecasting and granger-causal structure learning (Granger, 1969) within the semi-supervised domain adaptation framework.

Despite their comparative performance, these methods fail to disentangle stable and comprehensive domain-common causal rationales, as well as domain-specific non-causal correlations with complementarity and discriminability from complex dependencies among variables. Furthermore, they neglect the discrepancy and consistency of inter-class and intra-class causality, rendering them unsuitable for cross-domain time series classification under distribution shift.

3. Causal Model on Domain Adaptation of Multivariate Time Series

Notations. When distinguishing between domains, variables are annotated with superscripts in the upper right corner. The **labeled** *i.i.d.* samples $\mathcal{D}^s = \{(X_i^s, Y_i^s)\}_{i=1}^{N_s}$ are drawn from the source domain distribution P^s, and the **unlabeled** *i.i.d.* samples $\mathcal{D}^t = \{X_i^t\}_{i=1}^{N_t}$ are drawn from the target domain distribution P^t. In both the source and target domains, each X_i is a multivariate time series denoted by $X_i = \{x_1, \dots, x_K\} \in \mathbb{R}^{K \times \tau_{max}}$, where K is the number of variables and τ_{max} is the sampling length from the observed time steps T. $x_i = \{x_{i,t}\}_{t=1}^{\tau_{max}} \in \mathbb{R}^{\tau_{max}}$ is a unit time series representing the observation of variable i from time step 1 to τ_{max}. During evaluation, we test the learned model on the **labeled** *i.i.d.* samples $\mathcal{D}^{test} = \{(X_i^t, Y_i^t)\}_{i=1}^{N_{test}} \sim P^t$ from the target domain.

Causal Model. We instantiate the causal mechanisms governing the generation of multivariate time series data across domains using structural causal model (SCM) (Pearl et al., 2016). DSR (Cai et al., 2019) and iMSDA (Kong et al., 2022), tailored for UDA of non-time series, are the closest works to CauDiTS, assuming that cross-domain data are generated from two latent representations, domain-invariant Z_c and domain-variant Z_o. The key difference is that CauDiTS explores a deeper disentanglement between domain-common causal rationales G_c and domain-specific non-causal correlations G_o which precede the generation of Z_c, as despited in Figure 2(c). This has the advantage to eliminate the perturbations of G_o on Z_c, thereby blocking spurious causal paths between G_o and class label Y. Specifically, we have seven variables: time series data X, class label Y, domain-invariant representation Z_c (i.e., content), domain-variant representation Z_o (i.e., background or noise), domain-common causal rationales G_c (i.e., commonsense laws), domain-specific non-causal correlations G_o (i.e., specific style), and a domain prior U. A directed link between two variables indicates a causal relationship between them. We describe the causal relationships between these variables below.

- $U \rightarrow G_o \rightarrow Z_o$: U governs the distribution of domain-specific G_o, where G_o guides the interaction of multivariate variables to generate Z_o.

- $Z_c \rightarrow X \leftarrow Z_o$: X is generated by domain-invariant Z_c and domain-variant Z_o.

- $G_o \dashrightarrow Z_c$: This is a spurious causal relationship, indicating that the extraction of Z_c is influenced by G_o. This spurious causal link is pruned away in CauDiTS, but is implicit in existing methods (Ozyurt et al., 2022; Kong et al., 2022; He et al., 2023). As discussed in Section 1, previous works ignore in-depth exploration of disentangling causal rationales G_c and non-causal correlations G_o, resulting in capturing coarse-grained shortcut features. The explanation from CauDiTS is that if G_c and G_o cannot be disentangled to eliminate the perturbations of G_o in the extraction of Z_c, then the link $G_o \rightarrow Z_c$ inherently exists. In other words, if $G_o \rightarrow Z_c$, then G_o is a confounder between Z_c and Z_o, opening a backdoor path $Z_o \leftarrow G_o \rightarrow Z_c \rightarrow Y$, introducing a spurious correlation between Z_o and Y for previous works. It is evident that G_o varies with domains, making $Z_c \rightarrow Y$ unstable. In the UDA setting, this path should be excluded to ensure that Z_c does not contain any mixture of domain-specific information.

- $G_c \rightarrow Z_c \rightarrow Y$: G_c is stable and shared across domains. According to G_c, the interaction of multivariate variables generates Z_c. Assuming that G_c and G_o are disentangled, we can identify Y, which depends solely on Z_c without perturbations from Z_o across domains U. Therefore, in contrast to existing methods (He et al., 2023; Zhang et al., 2013), assuming the marginal distributions of X are differences across domains, i.e., $P^s(X^s) \neq P^t(X^t)$, while the conditional distributions remain constant, i.e., $P^s(Y|X^s) = P^t(Y|X^t)$, we propose the following assumption.

Assumption 3.1. *Given G_c, we assume that the marginal distribution $P(X)$ and the conditional distribution $P(Y|X)$ all vary across domains, i.e., $P^s(X^s) \neq P^t(X^t)$ and $P^s(Y|X^s) \neq P^t(Y|X^t)$, while the causal rationales G_c and the conditional distribution $P(Y|Z_c, G_c)$ remain fixed.*

Assumption 3.1 raises Question (1): *How can G_c and G_o be disentangled from the complex interrelationships between the variables?*

Furthermore, we assume that each class has a unique causal rationale that remains invariant across domains. Consequently, as depicted in Figure 1(c), we instantiate the domain-invariant causal rationales for class i, $i \in N_c$, as a directed graph $G_{c_i} = (\mathcal{V}_{c_i}, \mathcal{E}_{c_i}) \in G_c$, where $\mathcal{V}_{c_i} = \{x_1, x_2, \ldots, x_K\}$ is the set of variables, and $\mathcal{E}_{c_i} = \{(j,k) : x_j \to x_k\}_{j,k=1}^{K}$ is a set of edges between variables x_j and x_k. Without loss of generality, we represent the causal relationships $\mathcal{E}_{c_i}$ using an adjacent matrix $A_i = \{a_{j,k}\}_{j,k=1}^{K} \in \mathbb{R}^{K \times K}$, where $a_{j,k} = 1$ indicates a directed edge from x_j to x_k, and $a_{j,k} = 0$ indicates no edge. Similarly, we instantiate domain-specific non-causal correlations as a directed graph $G_{o_i} = (\mathcal{V}_{o_i}, \mathcal{E}_{o_i}) \in G_o$, with $B_i = \{b_{j,k}\}_{j,k=1}^{K} \in \mathbb{R}^{K \times K}$. G_c and G_o collectively form the intricate variable interrelationships $G_{\mathcal{I}}$, and $\mathcal{E}_{c_i} \cap \mathcal{E}_{o_i} = \phi$ and $\mathcal{I}_i = A_i + B_i$. This raises Question (2): *How to ensure that each class has a disentangled and unique G_c while keeping cross-domain consistency?*

In causal research (Pearl, 2009; Pearl et al., 2016), in a SCM, every variable is causally influenced by its parent variables. For example, variable X and its parent variables $Pa(X) = \{Z_c, Z_o\}$ have a causal function f_X: $Pa(X) \to X$, if and only if the causal mechanism $X = f_X(Pa(X), \epsilon_X)$ establish, where $\epsilon_X \perp\!\!\!\perp Pa(X)$ is an exogenous noise of X, as shown in Figure 2(c). Consequently, in the ideal CauDiTS, denoting the optimal domain-common causal rationales for class i as $G^*_{c_i}$, we have Proposition 3.2 on the causal disentangled domain adaptation for multivariate time series. The proof is given in Appendix B.

Proposition 3.2. *Assuming Assumption 3.1 and Questions (1) and (2) are answered by CauDiTS, disentangle the optimal $G^*_{c_i}$ for class i from $G_{\mathcal{I}}$. Then, for $\forall u \in U$ and $\forall j \in N_c$,*

$$Y_i = f_Y(f_{Z_c}(G^*_{c_i}, \epsilon_{Z_c}), \epsilon_Y) \quad s.t. \quad Y_i \perp\!\!\!\perp G^u_{o_j} | G^*_{c_i}, \tag{1}$$

where $f_Y : Z_c \to Y$ and $f_{Z_c} : G_c \to Z_c$ are the invertible causal functions. $G^u_{o_j}$ is domain-specific non-causal correlations of class j from domain u.

Proposition 3.2 indicates that the extraction of $Z^*_{c_i} = f_{Z_c}(G^*_{c_i}, \epsilon_{Z_c})$ depends only on $G^*_{c_i}$ for each class i. This shields Y from the influence of domain-variant $G^u_{o_j}$ of different domains u, thereby preserving the stability of the causal relationship $G_c \to Z_c \to Y$ across domains.

4. Methodology

We propose CauDiTS, a causal disentangled unsupervised domain adaptation framework for multivariate time series. Specifically, Section 4.1 and Section 4.2 address the first and second questions aforementioned, respectively.

4.1. Adaptive Causal Rationale Disentanglement

In UDA for multivariate time series, only variables X, Y and U are observed during training, while G_c and G_o are unobserved. To tackle the problem of effectively disentangle G_c and G_o from $G_{\mathcal{I}}$, we introduce an adaptive causal rationale disentanglement strategy. For simplicity, we do not differentiate between the source and target domains in this section.

CauDiTS first employs an LSTM model (Graves & Graves, 2012) to learn the hidden status $H = \{h_1, h_2, \ldots, h_K\}$ of $X = \{x_1, x_2, \ldots, x_K\}$, as shown in Figure 3. For each variable $x_i = \{x_{i,1}, \ldots, x_{i,\tau_{max}}\} \in X$ over time steps τ_{max}, the LSTM updates the hidden state $h_{i,t}$ of element $x_{i,t}$ using the following recurrence:

$$h_{i,t} = \text{LSTM}(x_{i,t}, h_{i,t-1}) \in \mathbb{R}^{d_h}, i \in \{1, 2, \ldots, K\}, \tag{2}$$

where $h_{i,t-1}$ denotes the hidden state propagating from time step 1 to step $t-1$ of x_i. d_h is the dimension of $h_{i,t}$.

Adaptive Rationale Disentangler. We propose an adaptive rationale disentangler (ARAD) to capture $G_{\mathcal{I}}$ among variables, while simultaneously disentangling G_c and G_o. With only U, obtaining $G_{\mathcal{I}}$ is impossible. ARAD employs a reverse generation mechanism, using observed time series data X to reveal its underlying $G_{\mathcal{I}}$. Specifically, ARAD integrates a variables interrelation attention module to capture spatio-temporal interrelationships between variables that form $G_{\mathcal{I}}$. This involves calculating the attention coefficient $\bar{\varepsilon}_{ij,t:t+1}$, which reflects the dynamic interrelationships of x_i to x_j from time steps t to $t+1$, defined as:

$$\bar{\varepsilon}_{ij,t:t+1} = v^T \sigma(W_1 h_{i,t} + W_2 h_{j,t+1} + b_1) + b_2, \tag{3}$$

where W_1, $W_2 \in \mathbb{R}^{d_\varepsilon \times d_h}$, $b_1 \in \mathbb{R}^{d_\varepsilon}$, $v \in \mathbb{R}^{d_\varepsilon}$ and $b_2 \in \mathbb{R}$ are the trainable parameters. $\sigma(\cdot)$ is an activation function. A softmax function is applied to $\bar{\varepsilon}_{ij,t:t+1}$ to ensure that all attention coefficients between time steps t

to $t+1$ sum to 1, i.e., $\varepsilon_{ij,t:t+1} = \frac{\exp(\bar{\varepsilon}_{ij,t:t+1})}{\sum_{i,j=1}^{K}\exp(\bar{\varepsilon}_{ij,t:t+1})}$. Then, we aggregate the obtained attention coefficients over time steps τ_{max} into the interrelationship coefficient matrix $\mathcal{I} = \{\varepsilon_{ij}\}_{i,j=1}^{K} \in \mathbb{R}^{K\times K}$, where $\varepsilon_{ij} = \max(\varepsilon_{ij,1:2},\ldots,\varepsilon_{ij,t:t+1},\ldots\varepsilon_{ij,\tau_{max}-1:\tau_{max}})$.

ARAD treats the disentanglement of G_c and G_o as an edge selection problem. Some edges are domain-invariant and represent stable causal generation patterns for cross-domain samples, while others are domain-specific and class-irrelevant. We employ a binary mask as a disentangler, represented as $M = \{m_{ij}\}_{i,j=1}^{K} \in \{0,1\}^{K\times K}$, as shown in Figure 3. This helps us to disentangle $A \in \mathbb{R}^{K\times K}$ and $B \in \mathbb{R}^{K\times K}$ from $\mathcal{I}$ as follows:

$$\begin{aligned} &\bar{M} = \sigma(W_{\bar{M}}\mathcal{I} + b_{\bar{M}}), m_{i,j} = \mathbb{I}(\bar{m}_{i,j} > \alpha), \\ &A = \mathcal{I}\odot M, B = \mathcal{I}\odot(1-M) \end{aligned} \tag{4}$$

where $W_{\bar{M}} \in \mathbb{R}^{K\times K}$ and $b_{\bar{M}} \in \mathbb{R}$ are trainable parameters. $\bar{M} = \{\bar{m}_{ij}\}_{i,j=1}^{K}$ is the trainable mask of M. α is the hyperparameter for selecting domain-invariant edges. $\odot$ denotes element-wise multiplication.

After disentangling A and B, the next steps involve designing f_{Z_c} and f_Y to learn the domain-invariant representation Z_c and label Y. To achieve this, we use graph convolutional networks (GCNs), inspired by their success in node representation learning through the aggregated effect of neighborhood variables (Wu et al., 2020; Dai & Chen, 2022). We first use a causally augmented multi-layer graph network, denoted as GCN_c, to aggregate hidden states of parents for dependency encoding, thereby generating node representations $R_c = \{r_1,\ldots,r_t,\ldots,r_{\tau_{max}}\} \in \mathbb{R}^{\tau_{max}\times(K\times d_r)}$:

$$\begin{aligned} &r_t^l = W_5(\text{ReLU}(Ar_t^{l-1}W_3 + r_{t-1}^{l-1}W_4)) + b_5, \\ &r_t = W_6(r_t^1 \oplus r_t^2 \oplus \ldots \oplus r_t^L) + b_6, \end{aligned} \tag{5}$$

where $W_3 \in \mathbb{R}^{d_h\times d_h}$, $W_4 \in \mathbb{R}^{d_h\times d_h}$, $W_5 \in \mathbb{R}^{d_h\times d_h}$, $W_6 \in \mathbb{R}^{d_h\times d_r}$, $b_5 \in \mathbb{R}^{d_h}$ and $b_6 \in \mathbb{R}^{d_r}$ are the trainable parameters. r_t^l denotes the aggregated node representations in layer l, and $r_t^0 = h_t \in \mathbb{R}^{K\times d_h}$ is the hidden states of all variables at time t. L denotes the number of layers. $\oplus$ denotes the concatenation of representations. Then, R_c is passed to a domain-invariant feature extractor $F_c(\cdot)$ to extract $Z_c \in \mathbb{R}^{d_z}$. Furthermore, we employ a classifier $\Phi(\cdot)$ to map Z_c onto the probability distribution over classes. The process is formally defined as follows:

$$Z_c = F_c(R_c),\ Y_c = \Phi(Z_c). \tag{6}$$

Similarly, domain classification is performed employing another graph network, GCN_o, together with a domain-specific feature extractor $F_o(\cdot)$ and a domain discriminator $D(\cdot)$. More formally, the process is as follows:

$$R_o = GCN_o(B,H),\ Z_o = F_o(R_o),\ Y_o = D(Z_o). \tag{7}$$

Finally, we define a basic adaptation loss $\mathcal{L}_{bs}$ for CauDiTS, which integrates three losses: (1) the source domain classification loss $\mathcal{L}_{cls}$, (2) the target domain conditional entropy loss $\mathcal{L}_{ucls}$, and (3) the domain classification loss $\mathcal{L}_{dis}$, defined as:

$$\mathcal{L}_{bs} = \lambda_1\mathcal{L}_{cls} + \lambda_2\mathcal{L}_{ucls} + \lambda_3\mathcal{L}_{dis}, \tag{8}$$

where λ_1, λ_2 and λ_3 are the balancing coefficients regulating the importance of each loss. Further details of the three loss terms are given in Appendix C.1.

4.2. Cross-Domain Causal Rationales Consistency Learning

Based on Section 4.1, we disentangle G_c and G_o from $G_{\mathcal{I}}$, but we posit that each class has a unique and invariant domain-common causal rationales. Therefore, we propose learning strategies of causal prototype consistency and domain intervention causality invariance to answer the second question, i.e., to promote cross-domain consistency and invariance of intra-class causal rationales.

Following previous works (Ozyurt et al., 2022; Eldele et al., 2023), we apply time series augmentations to each multivariate time series X from the source and target domains (see Appendix D.1 for details). After the augmentations, two augmented views of X are obtained, denoted query $\tilde{X}$ and key $\hat{X}$.

Causal Prototype Consistency Loss. We can impose supervised graph consistency constraint on G_c using class labels (e.g., constraining $G_{c_{j,1}} = \cdots = G_{c_{j,N_j}}$, for class j). However, this constraint may negatively affect the quality of G_c by directly aligning suboptimal shared graphs during the initial training phase. Furthermore, in the UDA setting, the target domain data lacks class labels. To address this, we introduce a domain-common causal prototype $\mathbb{P}$ for each class and perform the causal prototype consistency loss $\mathcal{L}_{cpc}$ between the node representations R_c and $\mathbb{P}$. Concretely, we first randomly select N_r samples from $\mathcal{D}^s$ for each class, forming $\mathcal{D}^r$. Utilizing Equation (5), we compute the node representation set $\tilde{\mathcal{R}}_c^s = \{\tilde{\mathcal{R}}_{c_i}^s\}_{i=1}^{N_c\times N_r}$ of $\mathcal{D}^r$ in the query view of source domain. Then, we apply singular value decomposition (SVD) on $\tilde{\mathcal{R}}_c^s$ to obtain $\mathbb{P} = \{\mathbb{P}_j\}_{j=1}^{N_c} \in \mathbb{R}^{N_c\times K\times d_r}$ ($\mathbb{P}$ is updated after each training step):

$$\begin{aligned} &\mathbb{P}_j = \text{SVD}(\{\mathbb{I}_{Y_i^s==j}\tilde{\mathcal{R}}_{c_i}^s\}) \in \mathbb{R}^{K\times d_r}, i = 1,\ldots,N_c\times N_r, \\ &\text{s.t. } \frac{1}{K}\sum_{k=1}^{K}\mathbb{P}_{j,k}\mathbb{P}_{j,k}^T = \text{I}, \frac{1}{K}\sum_{k=1}^{K}\mathbb{P}_{j,k} = 0, \end{aligned} \tag{9}$$

where Y_i^s is the class label of $\tilde{\mathcal{R}}_{c_i}^s$. $\mathbb{P}_j$ is the causal prototype of the j-th class. $\mathbb{P}_{j,k}$ is the prototype representation of k-th variables belonging to class j. The indicator function $\mathbb{I}$ is equal to 1 if $Y_i^s == j$ and 0 otherwise. d_r is the dimension of $\mathbb{P}_{j,k}$. SVD helps to reduce noise and highlight potentially

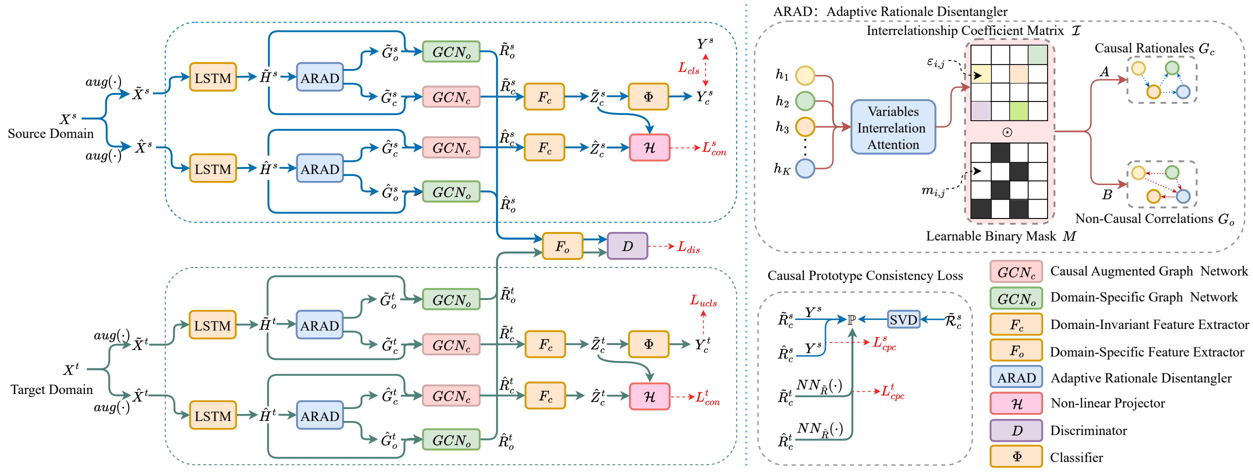


Figure 3: Overview of CauDiTS. CauDiTS introduces an adaptive rationale disentangler (ARAD) to disentangle domain-common causal rationales G_c and domain-specific non-causal correlations G_o from the complex variable interrelationships $G_{\mathcal{I}}$. According to G_c, CauDiTS efficiently extracts the domain-invariant representation Z_c using a causally augmented graph network GCN_c, which allows prediction of the class label Y_c independent of the domain-variant representation Z_o.

important features of the prototypes (Wu et al., 2021; Mo et al., 2023) .

In the source domain, the causal prototype consistency loss L_{cpc}^s is formulated as:

$$\mathcal{L}_{cpc}^s = \frac{1}{2N_c \cdot N_j} \sum_{j=1}^{N_c} \sum_{i=1}^{N_j} (\|\tilde{R}_{c_i}^s - \mathbb{P}_j\| + \|\hat{R}_{c_i}^s - \mathbb{P}_j\|), \quad (10)$$

where N_j denotes the number of node representations $\tilde{R}_{c_i}^s$ and $\hat{R}_{c_i}^s$ belonging to the j-th class. $\|\cdot,\cdot\|$ denotes the mean square error (MSE). In the target domain, where only unlabeled data are available, we define the causal prototype nearest-neighbor consistency loss $\mathcal{L}_{cpc}^t$ as follows:

$$\mathcal{L}_{cpc}^t = \frac{1}{2N_t} \sum_{i=1}^{N_t} (\|\tilde{R}_{c_i}^t - NN_{\tilde{R}}(\mathbb{P})\| + \|\hat{R}_{c_i}^t - NN_{\hat{R}}(\mathbb{P})\|), \quad (11)$$

where $NN_{\tilde{R}}(\cdot)$ and $NN_{\hat{R}}(\cdot)$ retrieve the nearest-neighbor of $\tilde{R}_{c_i}^t$ and $\hat{R}_{c_i}^t$ from $\mathbb{P}$, respectively. Therefore, we have the causal prototype consistency loss $\mathcal{L}_{cpc}$ as follows:

$$\mathcal{L}_{cpc} = \varphi_1 L_{cpc}^s + \varphi_2 L_{cpc}^t, \quad (12)$$

where the hyperparameters φ_1 and φ_2 control the contribution of each loss.

Invariant Causality of Domain Interventions. The causal prototype consistency loss ensures the intra-class comprehensive and consistent extraction of causal rationales of the same class across different domains. However, achieving the complete and invariant disentanglement of Z_c requires additional constraints. We emphasize the independence between Z_c and Z_o in the content of time series cross-domain interdependence. To achieve this, we employ backdoor adjustment (Pearl et al., 2016; Tian et al., 2006) to infer $P(Y|do(Z_c))$ by intervening on Z_c and conditioning on the confounding variable G_o. This breaks the link between Z_c and its pseudo-parent node G_o, effectively blocking the backdoor path $Z_o \leftarrow G_o \rightarrow Z_c \rightarrow Y$. The backdoor adjustment assumes that we have observed the confounder, i.e., $G_o = \{G_{o_j}^u\}$, where each $G_{o_j}^u$ is a randomly selected non-causal correlation within the domain $u \in U$. Formally, as shown in Appendix C.2, the deconfounded model for the graph in Figure 2(c) is:

$$P(Y|do(Z_c)) = \sum_{u}^{U} P(Y|Z_c = Z_{c_i}, Z_o = Z_{o_j}^u) P(G_o = G_{o_j}^u), \quad (13)$$

where the generated $Z_{c_i} = f_{Z_c}(G_{c_i}, \epsilon_{Z_c})$ and domain-variant representation $Z_{o_j}^u = f_{Z_o}(G_{o_j}^u, \epsilon_{Z_o})$ form the intervention pair $(Z_{c_i}, Z_{o_j}^u)$. $P(G_o = G_{o_j}^u) = 1/U$, assuming a uniform prior for the non-causal correlations. The detailed derivation is given in Appendix C.2. Therefore, Equation (13) can be reformulated as:

$$P(Y|do(Z_c)) = \frac{1}{U} \sum_{u \in U} P(Y|Z_c = Z_{c_i}, Z_o = Z_{o_j}^u). \quad (14)$$

Furthermore, we model P in Equation (14) as a softmax probability and adopt the supervised cross-entropy classification loss in the query view of the source domain. Maximizing the probability of $P(Y|do(Z_c))$ is equivalent to minimizing the supervised cross-entropy classification loss:

$$\mathcal{L}_{dcls} = \frac{1}{N_s \cdot U} \sum_{i,j=1, i \neq j}^{N_s} \sum_{u}^{U} \mathcal{J}(\Phi(\tilde{Z}_{c_i}^s \oplus \tilde{Z}_{o_j}^u), Y_i^s), \quad (15)$$

where $\mathcal{J}(\cdot)$ is the cross-entropy function. $\tilde{Z}^u_{o_j}$ is the domain-variant representation of the selected $\tilde{G}^u_{o_j}$ from the query view of domain u. Motivated by the re-weighting mechanism (Wang et al., 2021; Deng & Zhang, 2022), we reweight the probability distribution to characterize the perturbations of different $\tilde{Z}^u_{o_j}$ on the context distribution of $\tilde{Z}^s_{c_i}$:

$$\mathcal{L}_{dcls} = \frac{1}{N_s \cdot U} \sum_{i,j=1,i\neq j}^{N_s} \sum_{u}^{U} w^u_{ij} \mathcal{J}(\Phi(\tilde{Z}^s_{c_i} \oplus \tilde{Z}^u_{o_j}), Y^s_i), \quad (16)$$

where $w^u_{ij} = \frac{\sum_{k=1}^{K} (\tilde{r}^s_{c_{i,k}} - \bar{R}^s_{c_i})(\tilde{r}^u_{o_{j,k}} - \bar{R}^u_{o_j})}{\sqrt{\sum_{k=1}^{K} (\tilde{r}^s_{c_{i,k}} - \bar{R}^s_{c_i})^2 (\tilde{r}^u_{o_{j,k}} - \bar{R}^u_{o_j})^2}}$ is the Pearson correlation coefficient between the node representations $\tilde{R}^s_{c_i} = \{\tilde{r}^s_{c_{i,k}}\}^K_{k=1}$ and $\tilde{R}^u_{o_j} = \{\tilde{r}^u_{o_{j,k}}\}^K_{k=1}$ with respect to $\tilde{G}^s_{c_i}$ and $\tilde{G}^u_{o_j}$. $\bar{R}^s_{c_i}$ and $\bar{R}^u_{o_j}$ are the averages of all node representations in $\tilde{G}^s_{c_i}$ and $\tilde{G}^u_{o_j}$, respectively. It is noteworthy that when minimizing $\mathcal{L}_{dcls}$, the weight w^u_{ij} is encouraged to converge to 0. As $\mathcal{L}_{dcls}$ converges, $\tilde{R}^s_{c_i}$ and $\tilde{R}^u_{o_j}$ are statistically independent. This implies that G_c can be disentangled without influence from any G_o, thereby achieving independent extraction of the domain-invariant representation Z_c.

Intra-class Contrastive Learning. By emphasizing intra-class relationships within each domain, we can promote compactness of intra-class distributions in a unified space and discriminate different classes across domains. We apply intra-class contrastive learning to effectively reduce the disparity between causal representations Z_c of augmented in-domain views. Specifically, a non-linear projector $\mathcal{H}(\cdot)$ takes $\tilde{Z}_c$ and $\hat{Z}_c$ from the query and key views as inputs and generates the embeddings $\tilde{E}_c$ and $\hat{E}_c$. The in-domain intra-class embeddings contrastive losses of the source and target domains are formulated as:

$$\begin{aligned} \mathcal{L}^s_{con} &= -\frac{1}{N_s} \sum_{i=1}^{N_s} \log \frac{\exp(\tilde{E}^s_{c_i} \cdot \hat{E}^s_{c_i} / \tau_s)}{\exp(\tilde{E}^s_{c_i} \cdot \hat{E}^s_{c_i} / \tau) + \sum_{j=1}^{J} \exp(\tilde{E}^s_{c_i} \cdot \hat{E}^s_{c_j} / \tau_s)}, \\ \mathcal{L}^t_{con} &= -\frac{1}{N_t} \sum_{i=1}^{N_t} \log \frac{\exp(\mathrm{sim}(\tilde{E}^t_{c_i}, \hat{E}^t_{c_i} / \tau_t))}{\sum_{j=1}^{J} \mathbb{I}_{i\neq j} \exp(\mathrm{sim}(\tilde{E}^t_{c_i}, \hat{E}^t_{c_j}) / \tau_t)}, \end{aligned} \quad (17)$$

where $\cdot$ is the inner dot product. $\tau_s > 0$ and $\tau_t > 0$ are the temperature scaling parameters. J is the number of negative samples. The indicator function $\mathbb{I}_{i\neq j}$ is 0 if $i == k$ and 1 otherwise. sim$(\cdot)$ is the cosine similarity function.

4.3. The Overall Objective

In the training phase, the overall objective function of CauDiTS is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{bs} + \mathcal{L}_{cpc} + \gamma_1 \cdot \mathcal{L}_{dcls} + \gamma_2 \cdot \mathcal{L}^s_{con} + \gamma_3 \cdot \mathcal{L}^t_{con}, \quad (18)$$

where γ_1, γ_2 and γ_3 are the balancing hyperparameters. All hyperparameter values are determined by grid search. In the inference phase, we use only the query branch of the target domain in CauDiTS to predict the label of non-augmented multivariate time series. The detailed overview of CauDiTS is given in Algorithm 1 of Appendix C.3.

5. Experiments

5.1. Experimental Setup

Datasets. We consider four multivariate time-series benchmark datasets, namely WISDM (Kwapisz et al., 2011), HAR (Anguita et al., 2013) HHAR (Stisen et al., 2015), and Boiler (Cai et al., 2021). Further details of the datasets are provided in Appendix D.2. WISDM, HAR and HHAR are three human activity recognition datasets, and each participant is defined as an independent domain. Boiler is a fault dataset comprising sensor data from three separate boilers, with each boiler as a domain. Following CLUDA (Ozyurt et al., 2022), we split each dataset into training, validation and test with a ratio of 0.7, 0.15 and 0.15, respectively. We randomly select five source→target pairs of domains for the experiment.

Baselines. CauDiTS is compared with the following baselines: VRADA (Purushotham et al., 2016b), CoDATS (Wilson et al., 2020), AdvSKM (Liu & Xue, 2021), SASA (Cai et al., 2021), RAINCOAT (He et al., 2023), and CLUDA (Ozyurt et al., 2022), which are representative UDA methods for multivariate time series. Moreover, CAN (Kang et al., 2019), CDAN (Long et al., 2018), DDC (Tzeng et al., 2014), DeepCORAL (Sun & Saenko, 2016), DSAN (Zhu et al., 2020), HOMM (Chen et al., 2020) and MMDA (Rahman et al., 2020) are general UDA methods originally provided for non-time series data, which have shown outstanding performance when applied to time series data (Liu & Xue, 2021; Eldele et al., 2023; Ozyurt et al., 2022).

Evaluation. The reported performance of all baselines is derived from publicly available results or re-implemented using benchmarking suites (Ragab et al., 2022). For each selected source↦target pair in each dataset, the reported performance for each method consists of the mean predicted accuracy and the mean Macro-F1 score calculated over 10 random, independent initializations.

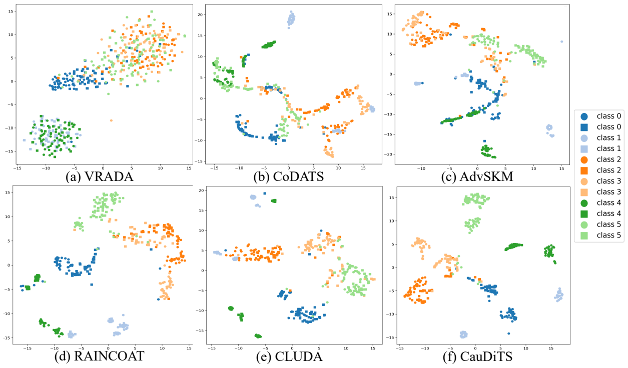
Implementation. The implementation details of CauDiTS and all baselines are described in Appendix F.

5.2. Results

Comparison with Baselines. In Table 1, we present the mean Macro-F1 scores for each selected source↦target pair across all datasets for each baseline method. The complete lists are shown in Table 5 (Macro-F1) and Table 6 (Accuracy) in Appendix G.1. CauDiTS outperforms the baselines on all benchmark datasets, achieving an average improvement of 11.81%. On the Boiler dataset, CauDiTS outperforms the second-best method CLUDA by 12.99% (0.680

Table 1: The results of selected source ↦ target pairs in four benchmark datasets in terms of mean Macro-F1 over 10 independent runs. The best results are shown in **bold**, and the second-best results are underlined.

Dataset	Source ↦ Target	VRADA	CoDATS	AdvSKM	CAN	CDAN	DDC	DeepCORAL	DSAN	HoMM	MMDA	SASA	RAINCOAT	CLUDA	CauDiTS	↑
Boiler	1 ↦ 2	0.505	0.487	0.519	0.468	0.522	0.542	0.542	0.468	0.514	0.512	0.508	0.552	0.559	**0.5798**	+3.72%
	1 ↦ 3	0.664	0.660	0.739	0.890	0.718	0.923	0.929	0.934	0.933	0.580	0.909	0.935	0.929	**0.9632**	+3.02%
	2 ↦ 1	0.487	0.495	0.585	0.497	0.499	0.563	0.523	0.490	0.519	0.500	0.494	0.591	0.570	**0.6381**	+7.96%
	2 ↦ 3	0.410	0.398	0.481	0.398	0.398	0.484	0.408	0.398	0.405	0.398	0.400	0.498	0.499	**0.7251**	+45.31%
	3 ↦ 1	0.496	0.640	0.844	0.621	0.425	0.778	0.805	0.476	0.809	0.631	0.810	0.807	0.847	**0.9354**	+10.44%
	Avg	0.512	0.536	0.633	0.575	0.552	0.685	0.641	0.553	0.636	0.524	0.624	0.677	0.680	**0.7683**	+12.99%
WISDM	12 ↦ 7	0.437	0.612	0.655	0.636	0.546	0.632	0.486	0.574	0.442	0.539	0.633	0.684	0.678	**0.7960**	+16.37%
	19 ↦ 2	0.615	0.403	0.460	0.327	0.312	0.459	0.501	0.428	0.522	0.306	0.496	0.432	0.458	**0.7557**	+22.88%
	2 ↦ 28	0.688	0.688	0.742	0.610	0.644	0.669	0.726	0.654	0.691	0.677	0.708	0.700	0.788	**0.7944**	+0.81%
	26 ↦ 2	0.517	0.598	0.463	0.362	0.404	0.414	0.618	0.424	0.519	0.453	0.689	0.472	0.701	**0.8098**	+15.39%
	7 ↦ 26	0.308	0.405	0.416	0.395	0.344	0.412	0.396	0.401	0.406	0.385	0.391	0.424	0.403	**0.5249**	+23.80%
	Avg	0.513	0.541	0.547	0.466	0.450	0.517	0.545	0.496	0.516	0.472	0.583	0.542	0.607	**0.7362**	+21.29%
HAR	15 ↦ 19	0.657	0.663	0.664	0.593	0.696	0.658	0.708	0.831	0.686	0.656	0.957	0.940	0.957	**0.9657**	+0.91%
	19 ↦ 25	0.737	0.381	0.359	0.640	0.768	0.360	0.535	0.754	0.397	0.348	0.560	0.956	0.932	**0.9831**	+2.83%
	20 ↦ 6	0.773	0.603	0.576	0.725	0.796	0.529	0.666	0.759	0.627	0.641	0.827	0.903	**1.000**	**1.0000**	+0.00%
	23 ↦ 13	0.696	0.440	0.436	0.410	0.660	0.447	0.616	0.606	0.549	0.527	0.709	0.778	0.762	**0.8308**	+6.79%
	13 ↦ 19	0.696	0.738	0.769	0.729	0.837	0.752	0.763	0.662	0.798	0.752	0.943	0.946	0.911	**0.9715**	+2.70%
	Avg	0.712	0.565	0.560	0.619	0.751	0.549	0.658	0.723	0.609	0585	0.799	0.905	0.892	**0.9502**	+4.99%
HHAR	2 ↦ 4	0.415	0.320	0.219	0.294	0.431	0.231	0.305	0.143	0.230	0.192	0.447	0.523	0.526	**0.6457**	+22.76%
	4 ↦ 0	0.243	0.222	0.163	0.165	0.273	0.175	0.249	0.116	0.179	0.162	0.346	0.284	0.352	**0.4296**	+22.05%
	5 ↦ 1	0.756	0.723	0.692	0.813	0.848	0.707	0.766	0.285	0.738	0.765	0.916	0.964	0.950	**0.9661**	+0.22%
	7 ↦ 1	0.583	0.528	0.338	0.524	0.412	0.280	0.483	0.278	0.461	0.367	0.814	0.825	0.875	**0.8806**	+0.64%
	7 ↦ 5	0.529	0.374	0.154	0.546	0.480	0.175	0.496	0.192	0.323	0.283	0.624	0.633	0.626	**0.6736**	+6.41%
	Avg	0.505	0.433	0.313	0.468	0.488	0.314	0.450	0.203	0.386	0.354	0.629	0.646	0.666	**0.7191**	+7.97%



(a) VRADA (b) CoDATS (c) AdvSKM (d) RAINCOAT (e) CLUDA (f) CauDiTS

Figure 4: The t-SNE visualization shows the domain-invariant representations learned on the HHAR 2↦4 pair. Circles represent the source domain, while squares represent the target domain.

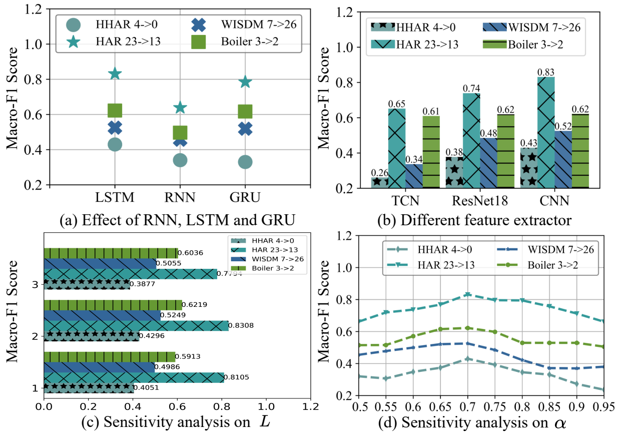
vs 0.7683) and achieves a remarkable 45.31% improvement on 2↦3. On the WISDM dataset, where the number of time series is limited, all baselines fail to maintain stable performance across different pairs. However, CauDiTS has more stable performance, achieving an improvement of 21.29% (0.607 vs 0.7362). On the HHAR dataset, CauDiTS achieves a remarkable average improvement of 7.97%, surpassing the best baseline accuracy of CLUDA (0.666 vs 0.7191). On the HAR dataset, CauDiTS outperforms RAINCOAT with a significant improvement of 4.99% (0.905 vs. 0.9502). This improvement is remarkable, especially considering the saturation performance of the existing method on the HAR dataset. Overall, the performances confirm that CauDiTS achieves significant Macro-F1 improvements, demonstrating its viability and effectiveness on all datasets.

In Figure 4, we visualize the domain-invariant representations extracted for baselines and CauDiTS on the HHAR 2↦4 pair. The t-SNE visualization of the representations shows that CauDiTS has a more compact intra-class distribution, while effectively separating clusters of different classes. This further demonstrates the effectiveness of CauDiTS and its ability to extract representations with higher levels of generalization and discrimination.

5.3. Ablation Study

Contribution of Each Component. We investigate the effectiveness of each component in CauDiTS. In Table 2, w/o UDA means that the model is trained on the source domain but tested on the target domain without any UDA strategy. Table 2 demonstrates a progressive increase in accuracy as more components are integrated, indicating that each loss serves as an essential component within CauDiTS. Remarkably, the introduction of the ARAD module resulted in a significant improvement in mean accuracy of approximately 8% across all datasets compared to using $\mathcal{L}_{bs}$ only. The full results are provided in Table 8 of Appendix G.1.

Ablation Studies of Model Architecture. Firstly, experiments are conducted to replace LSTM with RNN (Werbos,



(a) Effect of RNN, LSTM and GRU

(b) Different feature extractor

(c) Sensitivity analysis on L

(d) Sensitivity analysis on α

Figure 5: Results of ablation studies. (a)-(b) show the effects of different hidden state extractors and feature extractors, (c)-(d) show sensitivity analyses on the hyperparameters L and α.

1990) and GRU (Cho et al., 2014). As depicted in Figure 5(a), while there were some variations in Macro-F1, no significant differences are observed, and the model achieves the highest accuracy with LSTM. Subsequently, we investigate the performance of CauDiTS using different backbones, namely CNN(Ragab et al., 2022), TCN(Bai et al., 2018) and ResNet18(He et al., 2016), and present the results in Figure 5(b). The evaluations indicate that CNN outperforms the more complex TCN and ResNet18 on all datasets, which is why CauDiTS chose CNN. TCN has lower performance compared to other backbones attributed to the dilated convolution mechanism, which hinders its effective extraction of information from the augmented irregular time series data.

Sensitivity Analysis. The hyperparameter L represents the number of convolutional layers in GCN_c and GCN_o. Figure 5(c) illustrates the effect of different values of L on CauDiTS. CauDiTS exhibits robustness to variations in L, maintaining relatively consistent performance across all selected source$\mapsto$target pairs. Another crucial parameter is α, which governs the selection of edges is domain-invariant. Figure 5(d) demonstrates that optimal performance in CauDiTS is achieved when α is around 0.7, and its selection is sensitive to excessively high or low values. The full sensitivity analysis is reported in Appendix H.

6. Conclusion

In this paper, we propose CauDiTS, a framework for unsupervised domain adaptation of multivariate time series that disentangles domain-common causal rationales and non-causal domain-specific correlations. The adaptive rationale disentangler, equipped with a learnable causal mask, effectively filters out unstable domain-specific correlations, enabling the aggregation of domain-invariant representations. CauDiTS outperforms state-of-the-art baselines and provides insights into domain-common causal rationales often overlooked by existing works. Extensive experiments on benchmark datasets demonstrate its superiority.

Table 2: Ablation studies for each component in CauDiTS. We report the average prediction accuracy for the target domain across all datasets over 10 independent runs.

w/o UDA	$\mathcal{L}_{bs}$	ARAD	$\mathcal{L}_{con}$	$\mathcal{L}_{cpc}$	$\mathcal{L}_{dcls}$	Boiler	WISDM	HAR	HHAR
✓						0.6941	0.6420	0.6462	0.5135
	✓					0.7832	0.6800	0.7159	0.5551
	✓	✓				0.8510	0.7396	0.8462	0.6325
	✓	✓	✓			0.8812	0.7761	0.9137	0.7142
	✓	✓	✓	✓		0.8962	0.7999	0.9390	0.7539
	✓	✓	✓	✓	✓	0.9358	0.8375	0.9731	0.8001

Acknowledgements

This work was supported by National Natural Science Foundation of China under Project 62076096, the Open Research Fund of KLATASDS-MOE, and the Fundamental Research Funds for the Central Universities.

Impact Statement

Regarding the ethical aspects, our work does not introduce any inherent ethical concerns or issues. The focus of this work is to improve the robustness and accuracy of the unsupervised domain adaptation model for multivariate time series, which has positive implications for various applications, including healthcare, finance, weather, energy, and industrial processes.

The potential broader impact of our work lies in a more adaptive and robust unsupervised domain adaptation model for multivariate time series. Specifically, we propose CauDiTS, a novel framework for causally disentangled domain adaptation of multivariate time series. Based on a causal perspective, CauDiTS achieves a deeper disentanglement between domain-common causal rationales and domain-specific non-causal correlations before extracting domain-invariant representations. It is noteworthy that domain-common causal rationales play a crucial role in filtering out unstable domain-specific correlations, allowing the aggregation of domain-invariant representations from a causally augmented graph network. Incorporating non-causal domain-specific correlations facilitates effective domain discrimination in multivariate time series. Another significant contribution is our assumption that each class possesses unique and domain-invariant causal rationales. We further consider the consistency of intra-class causal rationales to promote the model's ability to generalize to unlabeled target domains. This aspect, the consistency of intra-class causal rationales, remains unexplored in previous works on unsupervised domain adaptation for multivariate time series.

In summary, our work aims to make a positive contribution to the field of unsupervised domain adaptation for multivariate time series, with potential applications to complex multivariate time series dynamic systems. We remain committed to promoting ethical practices in the development and deployment of AI technologies, ensuring responsible use of machine learning.

References

Adib, R., Griffin, P., Ahamed, S. I., and Adibuzzaman, M. A causally formulated hazard ratio estimation through backdoor adjustment on structural causal model. In *Proceedings of the 5th Machine Learning for Healthcare Conference*, volume 126 of *Proceedings of Machine Learning Research*, pp. 376–396. PMLR, 07–08 Aug 2020.

Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J. L., et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3, 2013.

Bagi, S. S. G., Gharaee, Z., Schulte, O., and Crowley, M. Generative causal representation learning for out-of-distribution motion forecasting. *arXiv preprint arXiv:2302.08635*, 2023.

Bai, L., Yao, L., Li, C., Wang, X., and Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

Cai, R., Li, Z., Wei, P., Qiao, J., Zhang, K., and Hao, Z. Learning disentangled semantic representation for domain adaptation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2060–2066, 2019.

Cai, R., Chen, J., Li, Z., Chen, W., Zhang, K., Ye, J., Li, Z., Yang, X., and Zhang, Z. Time series domain adaptation via sparse associative structure alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6859–6867, 2021.

Chen, C., Fu, Z., Chen, Z., Jin, S., Cheng, Z., Jin, X., and Hua, X.-S. Homm: Higher-order moment matching for unsupervised domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3422–3429, 2020.

Chen, L., Chen, D., Shang, Z., Wu, B., Zheng, C., Wen, B., and Zhang, W. Multi-scale adaptive graph neural network for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Cheng, Y., Yang, R., Xiao, T., Li, Z., Suo, J., He, K., and Dai, Q. Cuts: Neural causal discovery from irregular time-series data. In *The Eleventh International Conference on Learning Representations*.

Cheng, Y., Yang, R., Xiao, T., Li, Z., Suo, J., He, K., and Dai, Q. Cuts: Neural causal discovery from irregular time-series data. *arXiv preprint arXiv:2302.07458*, 2023.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

Dai, E. and Chen, J. Graph-augmented normalizing flows for anomaly detection of multiple time series. *arXiv preprint arXiv:2202.07857*, 2022.

Dai, W., Chen, Y., Xue, G.-R., Yang, Q., and Yu, Y. Translated learning: Transfer learning across different feature spaces. *Advances in neural information processing systems*, 21, 2008.

Deng, S., Wang, S., Rangwala, H., Wang, L., and Ning, Y. Cola-gnn: Cross-location attention based graph neural networks for long-term ili prediction. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 245–254, 2020.

Deng, X. and Zhang, Z. Deep causal metric learning. In *International Conference on Machine Learning*, pp. 4993–5006. PMLR, 2022.

Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C.-K., and Li, X. Contrastive domain adaptation for time-series via temporal mixup. *IEEE Transactions on Artificial Intelligence*, pp. 1–10, 2023. doi: 10.1109/TAI.2023.3293473.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V.

Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

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

Graves, A. and Graves, A. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Hasan, U., Hossain, E., and Gani, M. O. A survey on causal discovery methods for temporal and non-temporal data. *arXiv preprint arXiv:2303.15027*, 2023.

He, G., Liu, X., Fan, F., and You, J. Classification-aware semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 964–965, 2020.

He, H., Queen, O., Koker, T., Cuevas, C., Tsiligkaridis, T., and Zitnik, M. Domain adaptation for time series under feature and label shifts. In *International Conference on Machine Learning*, 2023.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hu, H., Tang, M., and Bai, C. Datsing: Data augmented time series forecasting with adversarial domain adaptation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2061–2064, 2020.

Hyvärinen, A., Zhang, K., Shimizu, S., and Hoyer, P. O. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(5), 2010.

Jin, X., Park, Y., Maddix, D., Wang, H., and Wang, Y. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, pp. 10280–10297. PMLR, 2022.

Kang, G., Jiang, L., Yang, Y., and Hauptmann, A. G. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4893–4902, 2019.

Kong, L., Xie, S., Yao, W., Zheng, Y., Chen, G., Stojanov, P., Akinwande, V., and Zhang, K. Partial disentanglement for domain adaptation. In *International Conference on Machine Learning*, pp. 11455–11472. PMLR, 2022.

Kwapisz, J. R., Weiss, G. M., and Moore, S. A. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

Lee, S. and Honavar, V. Towards robust relational causal discovery. In *Uncertainty in Artificial Intelligence*, pp. 345–355. PMLR, 2020.

Li, Y., Lang, J., Ji, L., Zhong, J., Wang, Z., Guo, Y., and He, S. Weather forecasting using ensemble of spatial-temporal attention network and multi-layer perceptron. *Asia-Pacific Journal of Atmospheric Sciences*, 57:533–546, 2021.

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

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

Liu, X., Yoo, C., Xing, F., Oh, H., El Fakhri, G., Kang, J.-W., Woo, J., et al. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Transactions on Signal and Information Processing*, 11 (1), 2022.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Löwe, S., Madras, D., Zemel, R., and Welling, M. Amortized causal discovery: Learning to infer causal graphs from time-series data. In *Conference on Causal Learning and Reasoning*, pp. 509–525. PMLR, 2022.

Marcinkevičs, R. and Vogt, J. E. Interpretable models for granger causality using self-explaining neural networks. In *International Conference on Learning Representations*, 2021.

Mo, Y., Lei, Y., Shen, J., Shi, X., Shen, H. T., and Zhu, X. Disentangled multiplex graph representation learning. In *International Conference on Machine Learning*, pp. 24983–25005. PMLR, 2023.

Ozyurt, Y., Feuerriegel, S., and Zhang, C. Contrastive learning for unsupervised domain adaptation of time series. In *The Eleventh International Conference on Learning Representations*, 2022.

Pearl, J. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Pearl, J. *Causality*. Cambridge university press, 2009.

Pearl, J., Glymour, M., and Jewell, N. P. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Purushotham, S., Carvalho, W., Nilanon, T., and Liu, Y. Variational adversarial deep domain adaptation for health care time series analysis. 2016a.

Purushotham, S., Carvalho, W., Nilanon, T., and Liu, Y. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations*, 2016b.

Purushotham, S., Carvalho, W., Nilanon, T., and Liu, Y. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations*, 2017.

Ragab, M., Eldele, E., Tan, W. L., Foo, C.-S., Chen, Z., Wu, M., Kwoh, C.-K., and Li, X. Adatime: A benchmarking suite for domain adaptation on time series data. *arXiv preprint arXiv:2203.08321*, 2022.

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

Schmidt, P., Reiss, A., Duerichen, R., Marberger, C., and Van Laerhoven, K. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM international conference on multimodal interaction*, pp. 400–408, 2018.

Shu, R., Bui, H. H., Narui, H., and Ermon, S. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

Singh, A. Clda: Contrastive learning for semi-supervised domain adaptation. *ArXiv*, abs/2107.00085, 2021.

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

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Tank, A., Covert, I., Foti, N., Shojaie, A., and Fox, E. B. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2021.

Tian, J., Kang, C., and Pearl, J. A characterization of interventional distributions in semi-markovian causal models. In *AAAI*, pp. 1239–1244, 2006.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

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

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Wang, T., Zhou, C., Sun, Q., and Zhang, H. Causal attention for unbiased visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3091–3100, 2021.

Wang, X., Saxon, M., Li, J., Zhang, H., Zhang, K., and Wang, W. Y. Causal balancing for domain generalization. In *The Eleventh International Conference on Learning Representations*, 2022.

Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Wilson, G., Doppa, J. R., and Cook, D. J. 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*, pp. 1768–1778, 2020.

Wu, A., Zhao, S., Deng, C., and Liu, W. Generalized and discriminative few-shot object detection via svd-dictionary enhancement. *Advances in Neural Information Processing Systems*, 34:6353–6364, 2021.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

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

Yang, L. and Hong, S. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*, pp. 25038–25054. PMLR, 2022.

Yang, S., Yu, K., Cao, F., Liu, L., Wang, H., and Li, J. Learning causal representations for robust domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 2021a.

Yang, X., Zhang, H., and Cai, J. Deconfounded image captioning: A causal retrospect. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021b.

Yue, Z., Zhang, H., Sun, Q., and Hua, X.-S. Interventional few-shot learning. *Advances in neural information processing systems*, 33:2734–2746, 2020.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. Domain adaptation under target and conditional shift. In *International conference on machine learning*, pp. 819–827. PMLR, 2013.

Zhang, X., Zhao, Z., Tsiligkaridis, T., and Zitnik, M. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022.

Zhang, Z., Wang, M., Huang, Y., and Nehorai, A. Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3437–3445, 2018.

Zhu, Y., Zhuang, F., Wang, J., Ke, G., Chen, J., Bian, J., Xiong, H., and He, Q. Deep subdomain adaptation network for image classification. *IEEE transactions on neural networks and learning systems*, 32(4):1713–1722, 2020.

Appendix

A. Related Work

A.1. Unsupervised Domain Adaptation of Non-Time Series.

Unsupervised Domain Adaptation (UDA) aims to learn a model capable of transferring knowledge from a labeled source domain to a heterogeneous and unlabeled target domain. UDA solutions mainly involve statistical divergence alignment mechanisms (Kang et al., 2019; Chen et al., 2020; Zhu et al., 2020; Sun & Saenko, 2016; Rahman et al., 2020) and adversarial learning (Tzeng et al., 2017; Long et al., 2018; He et al., 2020; Shu et al., 2018; Xie et al., 2018).

Statistical divergence alignment mechanisms focus on learning domain-invariant representations by minimizing domain discrepancies in the latent fusion feature space using various divergence measures (e.g., Maximum Mean Discrepancy (MMD) (Borgwardt et al., 2006)). DDC (Tzeng et al., 2014) introduces an adaptation layer and an additional domain confusion loss based on MMD to learn a semantically meaningful and domain invariant representation. CAN (Kang et al., 2019) estimates the target domain label with clustering while minimizing the contrastive domain discrepancy. HoMM (Chen et al., 2020) is a higher-order moment matching method that can perform arbitrary-order moment matching and is extended to reproduce kernel hilbert spaces (Zhang et al., 2018).

Adversarial learning, which uses a discriminator to adaptively guide the minimization of feature-level cross-domain divergence, helps to extract domain-invariant representations (Liu et al., 2022). CDAN (Long et al., 2018) integrates conditional adversarial learning into domain adaptation, guided by classifier predictions and innovative conditioning strategies. MSTN (Xie et al., 2018) proposes a moving semantic transfer network to learn semantic representations for target samples by aligning labeled source centroids and pseudo-labeled target centroids. Dirt-T (Shu et al., 2018) proposes a decision boundary iterative refinement training with a teacher model that combines domain adversarial training with a penalty term that punishes and minimizes the violation of the cluster assumption.

The above methods are not tailored for time series data and do not explicitly account for the temporal characteristics inherent in time series. However, with appropriate adjustments, some works such as CDAN (Long et al., 2018), DeepCORAL (Sun & Saenko, 2016), DSAN (Zhu et al., 2020), HOMM (Chen et al., 2020), and MMDA(Rahman et al., 2020) have been successfully adapted to time series data, resulting in promising performance.

A.2. Granger Causality Learning for Time Series.

Granger causality is initially introduced by Granger (Granger, 1969), who proposed to analyze the temporal causal relationships by testing the help of one time-series on predicting another time-series. Granger causality, initially based on a linear model, explores the causal structure by fitting vector auto-regressive (VAR) models (Hyvärinen et al., 2010).

With the rapid progresses and wide applications of neural networks (NNs), Granger causality inference methods have increasingly leveraged the expressive power offered by NNs (Tank et al., 2021; Cheng et al.; Löwe et al., 2022; Marcinkevičs & Vogt, 2021). Neural-GC (Tank et al., 2021) infers Granger causality directly from component-wise NNs by enforcing sparse input layers. ACD (Löwe et al., 2022) proposes an amortized causal discovery framework for time series, inferring causal relationships across samples with different underlying causal graphs but shared dynamics. CUTS (Cheng et al.) presents an iterative framework to jointly impute the irregular time series and discover Granger causal graphs. GVAR (Marcinkevičs & Vogt, 2021) is a framework for inferring multivariate Granger causality under nonlinear dynamics based on auto-regressive modeling with self-explanatory NNs. These neural-based Granger causality discovery algorithms aim to capture nonlinear lagged and instantaneous causal relationships between variables in time series data.

B. Proofs of Proposition 3.2

Proposition 3.1. *Assuming Assumption 3.1 and Questions (1) and (2) are answered by CauDiTS, disentangle the optimal $G^*_{c_i}$ for class i from $G_{\mathcal{I}}$. Then, for $\forall u \in U$ and $\forall j \in N_c$,*

$$Y_i = f_Y(f_{Z_c}(G^*_{c_i}, \epsilon_{Z_c}), \epsilon_Y) \quad s.t. \quad Y_i \perp\!\!\!\perp G^u_{o_j} | G^*_{c_i}, \tag{B.1}$$

where $f_Y : Z_c \to Y$ and $f_{Z_c} : G_c \to Z_c$ are the invertible causal functions. $G^u_{o_j}$ is domain-specific non-causal correlations of class j from domain u.

Proof. First, it is a fact that, the class label Y of cross-domain multivariate time series of the same class are consistent. Second, we assume that the causal rationales of the same class are cross-domain invariant and consistent, which means that the causal rationales of different multivariate time series of the same class are fully consistent, i.e., $G^u_{c_i} = G^{u'}_{c_i}, (u \neq u' \in U)$. Therefore, under the same causal function f_Y and f_{Z_c}, the domain-invariant representation Z_{c_i} from different cross-domain multivariate time series of class i is perfectly aligned, i.e., $Z^u_{c_i} = Z^{u'}_{c_i}$. Assuming that the optimal causal functions f_Y and f_{Z_c} have been found, then we have:

$$f_Y(f_{Z_c}(G^u_{c_i}, \epsilon_{Z_c}), \epsilon_Y) = f_Y(f_{Z_c}(G^{u'}_{c_i}, \epsilon_{Z_c}), \epsilon_Y). \tag{B.2}$$

$$f_{Z_c}(G^u_{c_i}, \epsilon_{Z_c}) = f_{Z_c}(G^{u'}_{c_i}, \epsilon_{Z_c}). \tag{B.3}$$

We use $f^{(-1)}_{Z_c}$ as the inverted function of f_{Z_c}. Denote the optimal domain-invariant representation as $Z^*_{c_i} = f_{Z_c}(G^*_{c_i}, \epsilon_{Z_c})$ for class i. In this paper, the optimal domain-invariant representation $Z^*_{c_i}$ and domain-variant representation $Z^u_{o_j} = f_{Z_o}(G^u_{o_j}, \epsilon_{Z_o})$ are statistically independent. According to the invertibility of f_{Z_c} and the independence between $Z^*_{c_i}$ and $Z^u_{o_j}$, we can reformulate Equation (B.3) as:

$$f^{(-1)}_{Z_c}\left(\begin{bmatrix} Z^*_{c_i} \\ Z^u_{o_j} \end{bmatrix}\right) = f^{(-1)}_{Z_c}\left(\begin{bmatrix} Z^*_{c_i} \\ Z^{u'}_{o_j} \end{bmatrix}\right). \tag{B.4}$$

Based on Equation (B.4), to demonstrate that under the optimal domain-invariant rationales G^*_c, function $f^{(-1)}_{Z_c}$ can fully extract domain-common complete and invariant information, we only need to prove that $f^{(-1)}_{Z_c}$ is a function of $Z^*_{c_i}$ but not the function of $Z^u_{o_j}$. To do this, we compute the Jacobian matrix of $f^{(-1)}_Z$ to analyze the first-order partial derivatives of $f^{(-1)}_{Z_c}$ and $f^{(-1)}_{Z_o}$ w.r.t $Z^*_{c_i}$ and $Z^u_{o_j}$. The Jacobian of $f^{(-1)}_Z$ can be denoted as:

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}, \tag{B.5}$$

where $J_{11} \in \mathbb{R}^{d_z \times d_z}$, $J_{12} \in \mathbb{R}^{d_z \times d_z}$, $J_{21} \in \mathbb{R}^{d_z \times d_z}$ and $J_{22} \in \mathbb{R}^{d_z \times d_z}$ are Jacobian matrices, and elements of them are formulated as:

$$\begin{aligned} [J_{11}]_{m,n} &= \frac{\partial \left[f^{(-1)}_{Z_c}(\mathcal{Z})\right]_m}{\partial (Z^*_{c_i})_n}, \quad & [J_{12}]_{m,k} &= \frac{\partial \left[f^{(-1)}_{Z_c}(\mathcal{Z})\right]_m}{\partial (Z^u_{o_j})_k}, \\ [J_{21}]_{k,m} &= \frac{\partial \left[f^{(-1)}_{Z_o}(\mathcal{Z})\right]_k}{\partial (Z^*_{c_i})_m}, \quad & [J_{22}]_{k,l} &= \frac{\partial \left[f^{(-1)}_{Z_o}(\mathcal{Z})\right]_k}{\partial (Z^u_{o_j})_l}, \end{aligned} \tag{B.6}$$

where $\mathcal{Z} = [Z^*_{c_i}, Z^u_{o_j}]^T$, $m, n, k, l \in [1, d_z]$. Based on Equation (B.6), we only need to prove that J_{12} is a zero matrix, and the determinant of J_{11} is a non-zero matrix. This would demonstrates that the matrix composed of all partial derivatives of $f^{(-1)}_{Z_c}$ w.r.t $Z^*_{c_i}$ is full rank while any partial derivatives of $f^{(-1)}_{Z_c}$ w.r.t $Z^u_{o_j}$ is zero.

Furthermore, Equation (B.4) is hold for all classes in the latent domain-invariant representation space, then for any fixed $\bar{Z}^*_{c_i}$ and $\bar{Z}^u_{o_j}$, for all $Z^{u'}_{o_j}$, we have:

$$f^{(-1)}_{Z_c}\left(\begin{bmatrix} \bar{Z}^*_{c_i} \\ \bar{Z}^u_{o_j} \end{bmatrix}\right) = f^{(-1)}_{Z_c}\left(\begin{bmatrix} \bar{Z}^*_{c_i} \\ Z^{u'}_{o_j} \end{bmatrix}\right), \tag{B.7}$$

Then we calculate the partial derivatives of Equation (B.7), and we have: $J_{12}|_{\bar{Z}_{c_i}, Z^{u'}_{o_j}} = J_{12}|_{\bar{Z}_{c_i}, \bar{Z}^{u}_{o_j}}$. Under the chain rules, and by considering derivatives with respect to constants, we can extend this to:

$$J_{12}|_{\bar{Z}_{c_i}, \bar{Z}^{u}_{o_j}} = \left(J_{f^{(-1)}_{Z_c}} |_{\bar{Z}_{c_i}, \bar{Z}^{u}_{o_j}} \right) \begin{bmatrix} 0_{d_z \times d_z} \\ 0_{d_z \times d_z} \end{bmatrix} = 0_{d_z \times d_z}, \tag{B.8}$$

where $J_{f^{(-1)}_{Z_c}} \in \mathbb{R}^{d_z \times 2d_z}$ is the Jacobian of $f^{(-1)}_{Z_c}$. Equation (B.8) holds for any fixed $\bar{Z}^*_{c_i}$ and $\bar{Z}^u_{o_j}$, and consequently, the same derivation is holds for all $Z^*_{c_i}$ and $Z^u_{o_j}$. Therefore, J_{12} is an all-zero matrix and the learned causal function $f^{(-1)}_{Z_c}$ is not a function of $Z^u_{o_j}$. Based on the aforementioned derivation, we can reformulate Equation (B.5) as:

$$J = \begin{bmatrix} J_{11} & 0_{d_z \times d_z} \\ J_{21} & J_{22} \end{bmatrix}, \tag{B.9}$$

Furthermore, we have the following equation:

$$\det(J) = \det(J_{11})\det(J_{22}) \neq 0. \tag{B.10}$$

In a nutshell, we have $\det(J_{11}) \neq 0$ and $\det(J_{22}) \neq 0$. Therefore, J_{11} is a non-zero matrix, and $f^{(-1)}_{Z_c}$ is only a function of $Z^*_{c_i}$, i.e., for $\forall u \in U$, we have $Y_i = f_Y(f_{Z_c}(G^*_{c_i}, \epsilon_{Z_c}), \epsilon_Y)$ s.t. $Y_i \perp\!\!\!\perp G^u_{o_j} | G^*_{c_i}$ for class i. □

C. Method Details

C.1. The Basic Adaption Loss

CauDiTS is trained using a domain adversarial learning strategy with a basic adaptation loss, $\mathcal{L}_{bs}$, which comprises three distinct losses: (1) the supervised classification loss $\mathcal{L}_{cls}$ of the source domain, (2) the unsupervised conditional entropy loss $\mathcal{L}_{ucls}$ of the target domain, (3) the domain classification loss $\mathcal{L}_{dis}$, denoted as:

$$\mathcal{L}_{bs} = \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{ucls} + \lambda_3 \mathcal{L}_{dis}, \tag{C.1}$$

where λ_1, λ_2 and λ_3 are balancing coefficients adjusting the importance of each component.

More formally, the three losses are defined as follows:

$$\mathcal{L}_{cls} = \frac{1}{N_s} \sum_{N_s} \mathcal{J}(\Phi(F_c(\tilde{R}^s_c)), Y^s) = \frac{1}{N_s} \sum_{N_s} \mathcal{J}(\tilde{Y}^s_c, Y^s), \tag{C.2}$$

$$\mathcal{L}_{ucls} = -\mathbb{E}_{\tilde{X}^t \in \mathbf{X}^t} [(\tilde{Y}^t_c)^T \log \tilde{Y}^t_c], \tag{C.3}$$

where $\mathcal{J}$ is the cross-entropy function. $\tilde{R}^s_c$ is the node representation of query view from source domain. $\tilde{Y}^t_c$ is the predicted label of multivariate time from query view of target domain. The domain classification loss $\mathcal{L}_{dis}$ is achieved by the gradient reversal layer $\mathcal{R}(\cdot)$ (Ozyurt et al., 2022) between the domain-specific feature extractor $F_o(\cdot)$ and discriminator $D(\cdot)$, formulated as:

$$\mathcal{L}_{dis} = \frac{1}{N_s} \sum_{N_s} \mathcal{J}(D(\mathcal{R}(F_o(R^s_o))), u = s) + \frac{1}{N_t} \sum_{N_t} \mathcal{J}(D(\mathcal{R}(F_o(R^t_o))), u = t), \tag{C.4}$$

where $\mathcal{R}(Z_o) = Z_o$ and $\frac{d\mathcal{R}}{dZ_o} = -\mathrm{I}$. $u = s$ and $u = t$ represent the domain labels from the source and target domains, respectively. R^s_o and R^t_o signify that multivariate time series samples can come from either the query view or the key view.

In the ablation study on the contribution of the components of CauDiTS (see Section 5.3), omitting the ARAD module results in changes to the definitions of $\mathcal{L}_{cls}$ and $\mathcal{L}_{dis}$. Specifically, the strategy of Adaptive Causal Rationale Disentanglement (see Section 4.1) is deactivate. The multivariate time series X is fed directly to the domain-invariant feature extractor $F_c(\cdot)$ and classifier $\Phi(\cdot)$ for class prediction. As a result, $\mathcal{L}_{cls}$ will be refined as follows:

$$\mathcal{L}_{cls} = \frac{1}{N_s} \sum_{i}^{N_s} \mathcal{J}(\Phi(F_c(\tilde{X}^s_i)), \tilde{Y}^s_i), \tag{C.5}$$

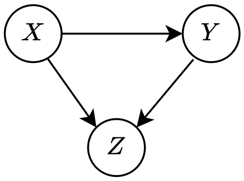
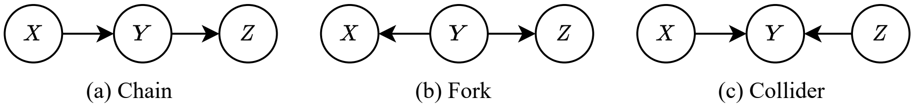


Figure 6: A causal graph.



(a) Chain

(b) Fork

(c) Collider

Figure 7: The three basic causal structures.

Furthermore, the adversarial training loss can be reformulated as:

$$\mathcal{L}_{dis} = \frac{1}{N_s}\sum_{i}^{N_s} \mathcal{J}(D(\mathcal{R}(F_o(\tilde{X}_i^s))), u = s) + \frac{1}{N_t}\sum_{i}^{N_t} \mathcal{J}(D(\mathcal{R}(F_o(\tilde{X}_i^t))), u = t), \tag{C.6}$$

C.2. Invariant Causality of Domain Interventions for CauDiTS

Causal Graphical Models: In causal research, a causal graph model or causal graph is typically a directed acyclic graph (DAG) G representing the joint probability distribution P over variables $X = \{x_1, x_2, \cdots, x_n\}$, where P is Markov with respect to G (Pearl, 1995; Hasan et al., 2023). The directed acyclic graph $G = (V, \mathcal{E})$ consists of nodes V and directed edges $\mathcal{E}$, where nodes V represent feature the observed value of variables X, and directed edges $\mathcal{E}$ represent conditional dependency relationships between variables X. Conditional dependency relationships are defined as causal relationships, where the arrows of directed edges represent the causal direction between nodes. The joint distribution P can be factorized as follows:

$$P(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} P(x_i | Pa(x_i)), \tag{C.7}$$

where $Pa(x_i)$ represents the parents of x_i in G. Typically, causal graphs use directed edges (→) from causes to effects to encode causal relationships between variables. As shown in Figure 6, X is a cause of Y and Z (i.e., Y←X→Z), and Y is also a cause of Z (i.e., Y→Z).

Causal graph models constructed on the causal relationships between variables generally include three known basic structures, namely chain, fork, and collider, as shown in Figure 7. The three basic causal structures and their metaphorical dependencies are as follows:

- Chain: As depicted in Figure 7(a), the causal path $X \rightarrow Y \rightarrow Z$ is a chain structure where X has a directed edge to Y and Y has an edge to Z. Here, X causes Y and Y causes Z, and Y is called a mediator.
- Fork: As depicted in Figure 7(b), the causal path $X \leftarrow Y \rightarrow Z$ is a fork structure where Y is the common parent of X and Z. In a fork structure, Y is called a confounder.
- Collider: As depicted in Figure 7(c), the causal path $X \rightarrow Y \leftarrow Z$ is a collider structure where one variable (Y) is a common child of the other two variables (X and Z), which are non adjacent. Y is called a collider.

Causal Assumptions: Often, the available data provide only partial insights into the underlying causal mechanisms. Consequently, it is imperative to make certain priors or assumptions about the structure of the causal relationships to facilitate causality learning (Lee & Honavar, 2020; Hasan et al., 2023). The following are the prevalent assumptions commonly adopted by causality learning community.

- **Causal Markovian Condition.** Each variable is independent of all its non-descendants, conditional on its parents.

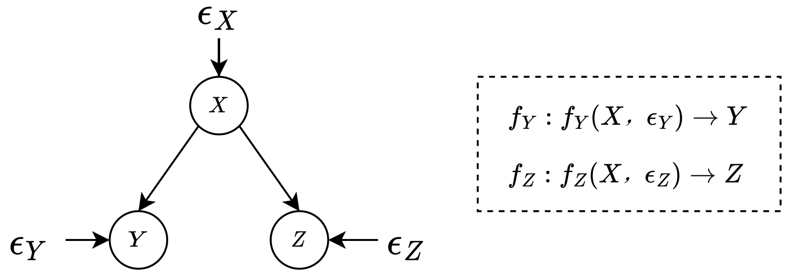

ϵ_X
X
ϵ_Y
Y
Z
ϵ_Z
$f_Y : f_Y(X, \epsilon_Y) \to Y$
$f_Z : f_Z(X, \epsilon_Z) \to Z$

Figure 8: A SCM with causal graph G and the corresponding causal functions f_Y and f_Z.

- **Causal Sufficiency.** The causal sufficiency assumption states that all the common causes of all variables are observed and there are no latent/hidden/unobserved confounders. This assumption is important for a variety of literature.

- **Causal Faithfulness.** The causal graph represents exactly the distributional independence relations implied by d-separation. Consequence: Any independence relations in the data are caused by the underlying structure of the graph that generated it, rather than from some random coincidence which narrows down the scope of possible causal graphs.

- **Acyclicity.** A graph must be acyclic in order to be a causal graph. According to the acyclicity condition, there can be no directed paths starting from a node and ending back to itself. This resembles the structure of a directed acyclic graph (DAG).

Structural Causal Model: A structural causal model (SCM) is a specific type of causal graph model used to formalize structural knowledge about the data generation process. SCMs are widely used in causal analysis, inference, and causal discovery tasks because they can represent the latent causal relationships within data. The formal definition of SCM is as follows:

Definition C.1. (Structural Causal Model) *Pearl (Pearl, 2009): A structural causal model is a 4-tuple $M = \{U, V, F, P(U)\}$, where:*

i. U is a set $\{\epsilon_1, \epsilon_2, \cdots, \epsilon_n\}$ of background variables, also called exogenous noise. These variables cannot be observed or intervened, but can influence other variables in the model.

ii. V is a set $\{v_1, v_2, \cdots, v_n\}$ of endogenous variables that are observable, and are determined by variables in the model, i.e., variables in $U \cup V$.

iii. F is a set of causal functions $\{f_1, f_2, \cdots, f_n\}$ such that each f_i assigns a value to the corresponding $V_i \in V$, $V_i \leftarrow f_i(Pa(V_i), \epsilon_i)$, for $i = 1, 2, \cdots, n$.

iv. $P(\epsilon)$ is a joint probability distribution over exogenous variables U.

Each SCM M is associated with a causal graph model G (DAG) and a set of causal functions f_i. In the SCM, each f_i assigns a value to the corresponding $V_i \in V$, $V_i \leftarrow f_i(Pa(V_i), \epsilon_i)$, for $i = 1, 2, \cdots, n$, representing the causal relationships between V_i and $Pa(V_i)$. As shown in Figure 8, the variable X is a direct cause of Y and Z because X appears in the functions that assign values to Y and Z, respectively. That is, if a variable Y or Z is a child of another variable X, then X is a direct cause of Y or Z. In Figure 8, ϵ_X, ϵ_Y and ϵ_Z are the exogenous noises; X, Y and Z are the endogenous variables; f_Y and f_Z are the causal functions assigned to the corresponding variables. An exogenous variable is characterized by the fact that it is either unobserved or unmeasured and, importantly, it cannot be a descendant of any other variable. Each endogenous variable is a descendant of at least one exogenous variable.

Do-operator: The do-operator, also referred to as do-intervention, represents a form of intervention in a causal model that involves fixing the observed values of one or more variables to specific values (Pearl, 1995; Yue et al., 2020). When an intervention is performed on an endogenous variable X in a SCM, setting its value to x is denoted as $do(X = x)$. Fixing the value of $X = x$ removes the edges to X, allowing for the simulation of experimental conditions to assess the changes in causal effects produced by the causal model when the node is intervened upon.

The difference between $do(X = x)$ and $X = x$ is quite clear. In the former, the external imposition fixes the value of the variable as a constant, whereas in the latter, the value is naturally observed in the data. This difference is also evident when evaluating causal effects, for example, by comparing $P(Y = y|do(X = x))$ and $P(Y = y|X = x)$. In terms of probability distributions, the former reflects the overall distribution of Y when each individual in the population sets the value of X to x, while the latter reflects the overall distribution of Y among individuals when the value of X is naturally x.

When intervening on X, i.e., $do(X = x)$, and calculating the causal effects on other variables, it is necessary to further transform expressions involving the do-operator into general probability expressions that can be calculated. This process is called do-calculus.

Do-calculus: Do-calculus aims to convert expressions constrained by the do-operator into expressions without the do-operator. Expressions without the do-operator can be estimated directly from observational data without the need for experimental intervention. Judea Pearl (Pearl, 1995; 2009) proposes a set of rules for the do-calculus, which includes three rules for transforming conditional probability expressions with the do-operator:

For a causal graph G, let X, Y, Z and W be the disjoint set of endogenous variables. $G_{\bar{X}}$ is used to represent the manipulated graph, where all incoming arrows to the node X are removed. Similarly, $G_{\underline{X}}$ represents the graph where outgoing arrows are removed from node X (Yang et al., 2021b; Yue et al., 2020).

Rule 1: permitting the addition or deletion of observations:

$$P(Y = y|do(X = x), Z = z, W = w) = P(Y = y|do(X = x), W = w), \quad \text{if}(Y \perp\!\!\!\perp Z|X, W)_{G_{\bar{X}}}. \tag{C.8}$$

In this case, the variable set Z blocks all paths from W to Y and all arrows leading to X have been removed.

Rule 2: permitting the replacement of an intervention by an observation, or vice verse:

$$P(Y = y|do(X = x), do(Z = z), W = w) = P(Y = y|do(X = x), Z = z, W = w), \quad \text{if}(Y \perp\!\!\!\perp Z|X, W)_{G_{\bar{X}\underline{Z}}}. \tag{C.9}$$

In this case, Z satisfies the backdoor criterion.

Rule 3: permitting the deletion or addition of interventions:

$$P(Y = y|do(X = x), do(Z = z), W = w) = P(Y = y|do(X = x), W = w), \quad \text{if}(Y \perp\!\!\!\perp Z|X, W)_{G_{\bar{X}\overline{Z(W)}}}, \tag{C.10}$$

where $Z(W)$ is the set of nodes in Z that are not ancestors of any W in $G_{\bar{X}}$. In this case, Z satisfies the backdoor criterion.

Backdoor Criterion: The backdoor criterion is applied while the confounder is observable. In fact, Rule 2 of the three rules in do-calculus is a generalization of the backdoor criterion. Judea Pearl (Pearl, 1995) proved that if there exists a set of variables Z that satisfies the backdoor criterion with respect to (X, Y), then the causal effect from X to Y is identifiable. The formal definition of the backdoor criterion is as follows:

Definition C.2. (Backdoor Criterion) (Pearl, 1995; Adib et al., 2020) *Given an ordered pair of variables (X, Y) in a directed acyclic graph G, a set of variables Z satisfies the backdoor criterion relative to (X, Y) if no node in Z is a descendant of X, and Z blocks every path between X and Y that contains an arrow into X.*

The corresponding backdoor adjustment strategy for the backdoor criterion is as follows:

Definition C.3. (Backdoor Adjustment) *If a set of variables Z satisfies the backdoor criterion relative to (X, Y), then the causal effect of X on Y is identifiable and is given by the formula: $P(y|do(X = x)) = \sum_z P(y|X = x, Z = z)P(Z = z)$.*

Derivation: Utilizing the three rules of the do-calculus and the backdoor criterion described above, we now illustrate the derivation process of the backdoor adjustment for $P(Y|do(Z_c = Z_{c_i}))$ as described in Section 4.2:

$$P(Y|do(Z_c)) = \sum_u^U P(Y|do(Z_c = Z_{c_i}), G_o = G^u_{o_j})P(G_o = G^u_{o_j}|do(Z_c = Z_{c_i})), \tag{C.11}$$

$$= \sum_u^U P(Y|do(Z_c = Z_{c_i}), G_o = G^u_{o_j})P(G_o = G^u_{o_j}), \tag{C.12}$$

$$= \sum_u^U P(Y|Z_c = Z_{c_i}, G_o = G^u_{o_j})P(G_o = G^u_{o_j}), \tag{C.13}$$

$$= \sum_u^U \sum_{Z^u_{o_j}} P(Y|Z_c = Z_{c_i}, G_o = G^u_{o_j}, Z_o = Z^u_{o_j})P(Z_o = Z^u_{o_j}|G_o = G^u_{o_j})P(G_o = G^u_{o_j}) \tag{C.14}$$

$$= \sum_u^U P(Y|Z_c = Z_{c_i}, Z_o = f_{Z_o}(G^u_{o_j}, \epsilon_{Z_o}))P(G_o = G^u_{o_j}), \tag{C.15}$$

where Equation (C.11) and Equation (C.14) follow the law of total probability. Equation (C.12) uses Rule 3 given $G_o \perp\!\!\!\perp Z_c$ in $G_{\overline{Z_c}}$. Equation (C.13) uses Rule 2 to change the intervention term to observation as $(Y \perp\!\!\!\perp Z_c|G_o)$ in $G_{\underline{Z_c}}$. In our SCM (as shown in Figure 2(c)), Z_o takes a deterministic value given by the causal function $Z_o \leftarrow f_{Z_o}(G_o, \epsilon_{Z_o})$. Therefore, the summation over all values of Z_o in Equation (C.14) is reduced to a single probability measure in Equation (C.15). Meanwhile, Equation (C.15) uses Rule 1 to delete G_o from the observation as $(Y \perp\!\!\!\perp G_o|Z_c)$ in $G_{\overline{Z_c}}$.

C.3. Detailed Overview of CauDiTS

The pseudocode for CauDiTS is presented in Algorithm 1.

D. Dataset Detail

D.1. Augmentations for Data

For each input multivariate time series, we apply time series augmentations in CauDiTS, following the methods proposed by CLUDA (Ozyurt et al., 2022) and CoTMix(Eldele et al., 2023). The augmentations are described below:

History Crop: We randomly mask out at minimum 20% (10% - 40%) of the initial time series with a probability of 50% (20% - 50%).

Historical Cutout: A random time window of 15% (5% - 20%) of the time series is masked with a probability of 50% (20% - 70%).

Channel Dropout: Each channel is masked independently with a probability of 10% (5% - 30%).

Gaussian Noise: Gaussian noise is applied independently to each measurement with a standard deviation of 0.1 (0.05 - 0.2).

We sequentially apply these augmentations twice to each multivariate time series. As a result, we have two augmented views of the same multivariate time series in CauDiTS.

D.2. Benchmark Datasets

We conduct experiments on four real-world multivariate time-series datasets: Boiler (Cai et al., 2021), WISDM (Kwapisz et al., 2011), HAR (Anguita et al., 2013) and HHAR (Stisen et al., 2015). These datasets are widely recognized as benchmark datasets in existing work on domain adaptation of multivariate time series. WISDM and HAR are two small-scale datasets, while HHAR and Boiler are two large-scale datasets. Summary statistics for all datasets are provided in Table 3, and a more detailed description of each dataset is given below:

WISDM: WISDM contains data from 36 participants collected using a 3-axis accelerometer sensor at 20 Hz. We use non-overlapping segments of 128 time steps to predict each participant's activity type. The dataset encompasses six activity

Table 3: Summary of the adopted datasets. (D: domains, M: channels, L: series length, C: classes.)

Dataset	D	M	L	C	Train samples	Val samples	Test samples
WISDM	36	3	128	6	4731	1274	1287
HAR	30	9	128	6	7194	1542	1563
HHAR	9	3	128	6	10336	2214	2222
Boiler	3	20	36	2	160734	134074	134074

types: walking, walking upstairs, walking downstairs, standing, sitting, and lying down. However, this dataset is more challenging due to a class-imbalance problem, where data from different participants may have only a small sample of subclasses from all classes.

HHAR: HHAR contains data from 9 participants collected using a 3-axis accelerometer sensor at 50 Hz. We use non-overlapping segments of 128 time steps to predict each participant's activity type. The dataset encompasses six activity types: biking, sitting, standing, walking, walking upstairs, and walking downstairs.

HAR: HAR contains data from three sensors, namely, 3-axis accelerometer, 3-axis gyroscope, and 3-axis body acceleration, worn by 30 participants. The data is collected at 50Hz, and non-overlapping segments of 128 time steps are employed to predict the type of activity of each participant. The dataset includes six types of activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying down.

Boiler: Boiler dataset records sensor data from three boilers spanning from March 24, 2014, to November 30, 2016. Each boiler is treated as an individual domain, with the sensor data mainly comprising the status of the blowdown valves for each boiler, including any associated mechanical faults. However, in routine operations, fault data is typically sparse. Consequently, this dataset demonstrates a class sample imbalance compared to other datasets, presenting increased research challenges.

We employ a dataset partitioning strategy consistent with CLUDA (Ozyurt et al., 2022), where each dataset is partitioned into three distinct subsets: training, validation, and testing. This partitioning follows a 70:15:15 ratio, with the datasets being mutually exclusive. The subsets serve specific purposes: the training set is used for model training, the validation set for parameter tuning and optimal model selection, and the test set to evaluate the performance of the trained model.

E. Baselines

We compare CauDiTS to the following methods:

VRADA (Purushotham et al., 2016b): VRADA is based on a variational recurrent neural network (VRNN) and trains adversarially to capture complex temporal relationships that are domain-invariant. It is the first attempt to capture and transfer latent temporal dependencies in multivariate time-series data.

CoDATS (Wilson et al., 2020): CoDATS, based on weak unsupervised adversarial training, employs the convolutional neural layer with gradient reversal as a feature extractor.

AdvSKM (Liu & Xue, 2021): AdvSKM performs time series domain matching by minimizing an extended version of the maximum mean discrepancy (MMD) (Tzeng et al., 2014) embedded in a hybrid spectral kernel network.

CAN (Kang et al., 2019): CAN optimizes a novel metric that explicitly models both intra-class and inter-class domain discrepancies, implementing an alternating update strategy for end-to-end training.

CDAN (Long et al., 2018): CDAN is a principled framework that conditions the adversarial adaptation models on the discriminative information conveyed in the classifier predictions.

DDC (Tzeng et al., 2014): DDC proposes a CNN with an adaptation layer and a confusion loss to obtain a semantically meaningful and domain-invariant representation, using a domain confusion metric for model selection.

DeepCORAL (Sun & Saenko, 2016): DeepCORAL learns a nonlinear transformation that aligns correlations of layer activations in deep neural networks of the source and target distributions.

DSAN (Zhu et al., 2020): DSAN learns a transfer network by aligning the relevant subdomain distributions of domain-specific layer activations across different domains based on a local maximum mean discrepancy (LMMD).

HoMM (Chen et al., 2020): HoMM is a higher-order moment matching method that achieves fine-grained domain alignment through arbitrary-order moment matching. By leveraging higher-order statistics, it effectively approximates complex non-gaussian distributions.

MMDA (Rahman et al., 2020): MMDA fits the second-order statistics (covariances) as well as the maximum mean discrepancy of the source and target data with a two-stream convolutional neural network.

SASA (Cai et al., 2021): SASA exploits the stability of causality to introduce a sparse associative structure alignment model for domain adaptation. The alignment of associative structures serves as a guiding mechanism for knowledge transfer between domains.

RAINCOAT (He et al., 2023): RAINCOAT is a domain adaptation method for time series that addresses feature and label shifts by combining and aligning features in time and frequency space, correcting for misalignment, and detecting label shifts.

CLUDA (Ozyurt et al., 2022): CLUDA proposes a novel framework for unsupervised domain adaptation (UDA) of time series through contrastive learning. It is the first work to acquire domain-invariant contextual representations in UDA of multivariate time series.

VRADA, CoDATS, AdvSKM, SASA, RAINCOAT and CLUDA are representative UDA methods tailored for multivariate time series. MMDA, HOMM, DSAN, DeepCORAL, DDC, CDAN and CAN are originally proposed for domain adaptation for non-time series but have been successfully adapted to time series with excellent performance (Liu & Xue, 2021; Cai et al., 2021; Eldele et al., 2023).

F. Implementation

The training, validation, and testing of all baselines and CauDiTS are conducted on identical dataset splits using an NVIDIA GeForce GTX 4090 with 24GB GPU memory and the PyTorch framework.

Table 4: Ranges of hyperparameter tuning.

Method	Hyperparameter	Tuning Range
	Learning Rate	$1{\cdot}10^{-4} \sim 1{\cdot}10^{-1}$
	Weight Decay	$1{\cdot}10^{-4}$, $1{\cdot}10^{-3}$, $1{\cdot}10^{-2}$
	BatchNormalization	True, False
	Dropout	0.1, 0.2, 0.3, 0.4, 0.5
VRADA (Purushotham et al., 2016b)	VRNN hidden dim	32, 64, 128
	VRNN latent dim	32, 64, 128
	VRNN num. layers	1, 2, 3
	Discriminator hidden dim.	64, 128, 256
	Weight discriminator loss	0.1, 0.5, 1
	Weight KL divergence	0.1, 0.5, 1
	Weight neg. log-likelihood	0.1, 0.5, 1
CoDATS (Wilson et al., 2020)	Discriminator hidden dim	64, 128, 256
	Weight discriminator loss	0.1, 0.5, 1
AdvSKM (Liu & Xue, 2021)	Spectral kernel hidden dim	32, 64, 128
	Spectral kernel output dim	32, 64, 128
	Spectral kernel type	Linear, Gaussian
	Num. kernel (if Gaussian)	3, 5, 7
	Weight MMD loss	0.1, 0.5, 1
CAN (Kang et al., 2019)	Kernel type	Linear Gaussian
	Num. kernel (if Gaussian)	1, 3, 5, 7
	Num. iterations k-means clustering (each loop)	1, 3,5
	Sampling type	Random, Class-aware

Method	Hyperparameter	Tuning Range
	Weight MMD loss	0.1, 0.5, 1
CDAN (Long et al., 2018)	Discriminator hidden dim	64, 128, 256
	Multiplier discriminator update	0.1, 1, 10
	Weight discriminator loss	0.1, 0.5, 1
	Weight conditional entropy loss	0.1, 0.5, 1
DDC (Tzeng et al., 2014)	Kernel type	Linear, Gaussian
	Num. kernel (if Gaussian)	1, 3, 5, 7
	Weight MMD loss	0.1, 0.5, 1
Deep-Coral (Sun & Saenko, 2016)	Weight Coral Loss	0.1, 0.3, 0.5, 1
DSAN (Zhu et al., 2020)	Kernel multiplier	1, 2, 3
	Num. kernel	3, 5, 7
	Weight domain loss	0.1, 0.5, 1
HOMM (Chen et al., 2020)	Moment Order	1,2,3
	Weight domain discrepancy loss	0.1, 0.5, 1
	Weight discriminative clustering loss	0.1, 0.5, 1
MMDA (Rahman et al., 2020)	Kernel type	Linear, Gaussian
	Num. kernel (if Gaussian)	1, 3, 5, 7
	Weight MMD loss	0.1, 0.5, 1
	Weight CORAL loss	0.1, 0.5, 1
	Weight Entropy loss	0.1, 0.5, 1
SASA (Cai et al., 2021)	LSTM Hidden Dim	4, 8, 12
	Num. Segments	4, 8, 12, 24
	Segments Lengths	3, 6, 12, 24
	Weight Intra-Attention Loss	$[1{\cdot}10^{-1}, 1{\cdot}10^{0}]$
	Weight Inter-Attention Loss	$[1{\cdot}10^{-1}, 1{\cdot}10^{0}]$
RAINCOAT (He et al., 2023)	Fourier frequency mode	10, 64, 200
	Regularization term in Sinkhorn divergence	$1{\cdot}10^{-3}$
	λ_1, λ_2, λ_3	$0 \sim 1$
CLUDA (Ozyurt et al., 2022)	Momentum	0.9, 0.95, 0.99
	Queue size	24576, 49152, 98304
	Discriminator hidden dim.	64, 128, 256
	Projector hidden dim.	64, 128, 256
	λ_{dis}	0.1, 0.5, 1
	λ_{CL}	0.05, 0.1, 0.2
	λ_{NNCL}	0.05, 0.1, 0.2
CauDiTS	Discriminator hidden dim	32, 64, 128
	Weight (λ_3) discriminator loss	0.1, 0.5, 1
	λ_1	0.75, 0.8, 0.85, 0.9, 0.95, 1.0
	λ_2	0.0001, 0.001, 0.01, 0.1, 1.0
	φ_1	0.1, 0.2, 0.4, 0.6, 0.8, 1.0
	φ_2	0.001, 0.01, 0.1
	γ_1	0.01, 0.1, 0.5, 1.0
	γ_2	0.01, 0.1, 1.0
	γ_3	0.0001, 0.001, 0.01, 0.1, 1.0
	Projector $\mathcal{H}$ hidden dim	32, 64, 128
	L	1, 2, 3
	α	0.5, 0.6, 0.7, 0.8, 0.9
	N_r	25, 50, 100, 150, 200, 300, 400, 500

F.1. Details on Neural Networks of CauDiTS

We employ a 1D-convolutional neural network (CNN) for the feature extractors $F_c(\cdot)$ and $F_o(\cdot)$, similar to RAINCOAT (He et al., 2023) and AdaTime (Ragab et al., 2022). The architecture of the 1D CNN is structured into three blocks, each consisting of a sequence of components: a 1D convolutional layer, followed by a 1D batch normalization layer, a rectified linear unit (ReLU) function to introduce non-linearity, and finally a 1D max-pooling layer. Corresponding ablation experiments were conducted regarding the adoption of different feature extractors, with the results presented in Figure 5(b). The evaluations indicates that CNN outperformes the more complex TCN and ResNet18 across all datasets, leading to the selection of CNN for CauDiTS. TCN is found to exhibit lower performance, which is attributed to the dilated convolution mechanism hindering its effective extraction of information from the augmented irregular time-series data.

The classifier $C(\cdot)$ employs a multi-layer perceptron. The LSTM model used is a single-layer unidirectional Long Short-Term Memory (LSTM) (Graves & Graves, 2012) with a dropout rate of 0.2. $GCN_c(\cdot)$ is a causal augmented multi-layer graph neural networks. The non-linear projector $\mathcal{H}(\cdot)$ is a non-linear hidden layer with 64 units.

F.2. Details of Training, Validation and Testing

In this subsection, we provide a detailed description of the hyperparameter selection, tuning, and best model selection for CauDiTS and all baselines. We train each method for a maximum of 15000 training steps with a batch size of 32 for WISDM, HAR, and HHAR, and 128 for Boiler. All methods are optimized using an Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The sliding window τ_{max} is set to 128 for WISDM, HAR and HHAR, and 36 for Boiler.

We adopt uniform strategies for hyperparameter tuning, early stopping, and model selection, similar to CLUDA (Ozyurt et al., 2022). Specifically, we individually tune hyperparameters for each method using grid search on source validation datasets. The hyperparameter search ranges are derived from the released original implementations or benchmark suites (Ozyurt et al., 2022; Ragab et al., 2022). Table 4 lists the tuning ranges of all hyperparameters for all methods. The parameters that appear in all methods are listed in the first rows of Table 4. The search range for the learning rates is from 1×10^{-4} to 1×10^{-1}.

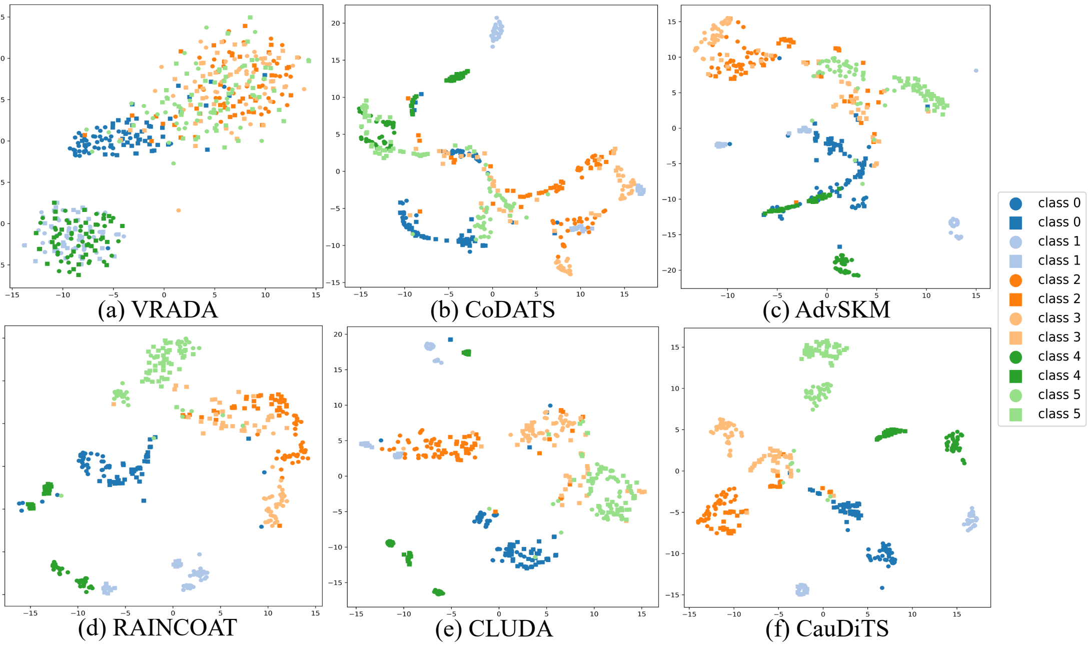
For early stopping, we rely on the validation loss and Macro-F1 scores from labeled source domain samples, excluding the target domain data. The best model is selected based on the highest performance in metrics such as accuracy and Macro-F1 scores on the source domain validation set. In our experimental, it is important to highlight that the tuning of model parameters, the early stopping of training, and the selection of the optimal model do not incorporate any unlabeled or labeled multivariate time-series data from the target domain. This configuration is chosen to better reflect real-world applications. After model selection, we report the prediction results on the labeled test samples from the target domain that were not seen during training and validation (parameter and model selection). We use the same setup for all baselines in this paper to ensure a fair comparison. To capture the variability in test performance for each method, we perform each experiment over 10 independent runs.

The transparency and reproducibility of our experiments with CauDiTS is achieved by providing these implementation details and hyperparameter tuning ranges.

G. More Experiments

G.1. Full Results Compared to Baselines

The comprehensive comparison results between CauDiTS and the baselines, in terms of Macro-F1 and accuracy across all datasets, are presented in Tables 5 and 6, respectively. Table 5 displays the Macro-F1 scores, while Table 6 shows the accuracy scores. In both tables, the reported values are the average performance after 10 independent runs of 10 randomly selected source→target pairs from each dataset. Notably, the Boilers dataset, with only three domains, yields a total of 6 different source→target pairs. Upon analyzing Table 5, it manifests the conspicuous superiority of CauDiTS over all baselines is mainfested, securing optimal results in 35 out of 36 source→target pairs across all datasets. In terms of accuracy, CauDiTS consistently outperforms the baseline methods, achieving optimal results in 33 out of 36 cross-domain pairs. This underscores the effectiveness of CauDiTS in significantly improving the benchmark Macro-F1 and the accuracy of unsupervised domain adaptation for multivariate time series.


(a) VRADA (b) CoDATS (c) AdvSKM

(d) RAINCOAT (e) CLUDA (f) CauDiTS

Figure 9: The t-SNE visualization depicts the representations learned on the HHAR 2↦4 pair. Circle markers represent the source samples, while squares represent the target.

G.2. t-SNE Visualizations of The Learned Representations

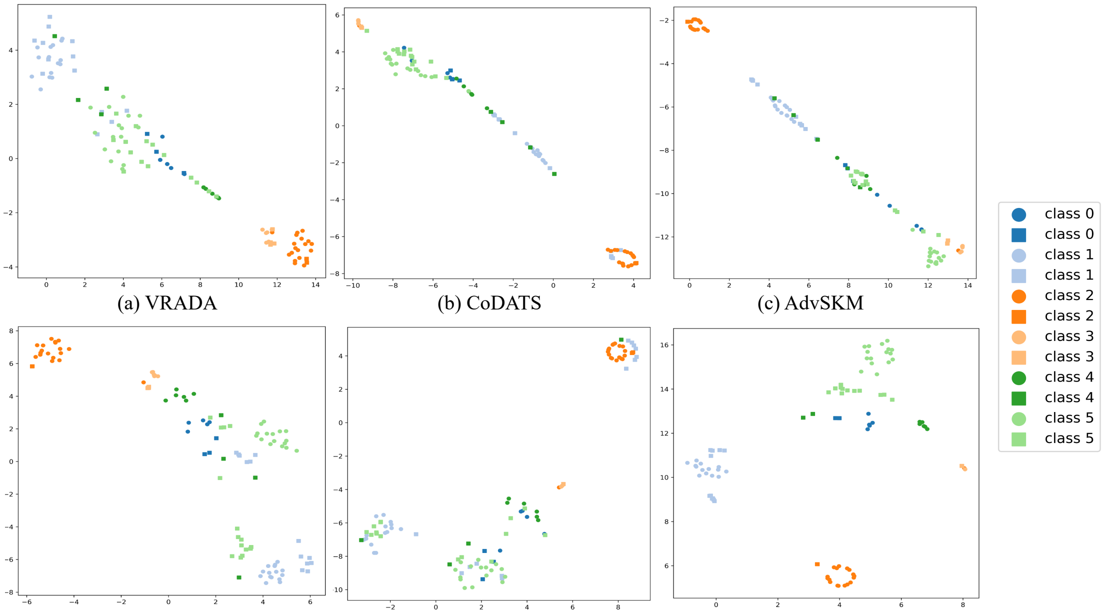
In this section, we visualize the representations learned by VRADA, CoDATS, AdvSKM, RAINCOAT, CLUDA and CauDiTS using the t-SNE tool (Van der Maaten & Hinton, 2008). Figures 9 and 10 present visual representations of the learned source domain and target domain feature distributions for each method on HHAR 2→4 and WISDM 19→2, respectively. Circular markers represent representations of source domain samples, while square markers represent representations of target domain samples. Different colors of circular and square markers correspond to representations of different classes. The t-SNE visualizations provide an intuitive observation of the differences in sample distribution between the source domain and the target domain. This allows an assessment of whether each method can effectively promote the compactness of the intra-class distribution and promote the discriminability of the inter-class distribution in a cross-domain setting.

The t-SNE visualizations in Figures 9 and 10 reveal a clear drift between the source and target domains. Adaption from the source 19 to the target 2 in the WISDM dataset proves challenging due to the limited traininng samples for each class and the initial low discriminability between classes. In Figure 10, the visualizations indicate severe clustering overlap for all baselines, whereas CauDiTS displays clear inter-class boundaries, resulting in more compact clusters. Overall, compared to other methods, CauDiTS effectively consolidates representations belonging to the same class across different domains and separates representations of different classes. While some overlap still exists, CauDiTS demonstrates a significant improvement over other methods.

G.3. Visualization of The Disentangled Causal Rationales

In the visualizations of the adjacency matrix of disentangled causal rationales in Figure 11 (WISDM dataset, task 26→2) and Figure 12 (HAR dataset, task 15→19), deeper colors signify more robust interdependencies. These visualizations suggest causal relationship between variables within the domain adaptation of multivariate time.

Emphasizing that the disentangled causal rationales obtained in CauDiTS do not represent the true summary causal graph between variables, as highlighted in causal discovery literature (Hasan et al., 2023; Löwe et al., 2022; Cheng et al., 2023). In this paper, the causal rationales are regarded as potential weak causal relationships, and no constraints or optimizations have been imposed on the authenticity and identifiability of causal rationales with respect to the true summary causal graph. Our focus is not on revealing the true causal structure between variables but rather on disentangling domain-invariant and



(a) VRADA (b) CoDATS (c) AdvSKM

(d) RAINCOAT (e) CLUDA (f) CauDiTS

Figure 10: The t-SNE visualization depicts the representations learned on the WISDM 19↦2 pair. Circle markers represent the source samples, while squares represent the target.

domain-variant components from the inter-variable interrelationships. The disentangled domain-invariant components are defined as causal rationales, representing a potential form of weak causality that is cross-domain invariant and stable. These disentangled causal rationales play a crucial role in guiding variables through the dimensions of time for information transfer and representation aggregation, ultimately yielding domain-invariant representations. Our future work will involve a more in-depth analysis and the imposition of constraints on the authenticity and identifiability of causal rationales. This is aimed at promoting the validity of CauDiTS, particularly in more intricate multivariate dynamic systems.

In the domain adaptation tasks, the adjacency matrices of the disentangled causal rationales obtained by CauDiTS are shown in Figures 11 and 12. Differences in the relationships between variables are evident, with deeper colors indicating stronger causal relationships. We observe that the causal rationales for the same classes in different domains maintain an essentially consistent structure, with variations in the strength of the relationships between variables. However, these variations are not pronounced, reflecting, to some extent, the ability of CauDiTS to ensure consistency of causal rationales across domains. Furthermore, there are also differences in whether causal relationships exist between variables and the strength of relationships between variables for causal rationales in different classes. Overall, however, the differences in causal rationales across classes are not obvious, suggesting that distinguishing causal rationales across classes is very challenging. This anomaly highlights the problem with existing works, which neglects to explore the discriminability and differences in causal rationales across classes. Such negligence is erroneous, as even subtle differences in causal rationales can lead to significant error accumulation over time due to the ongoing process of information transfer and representational aggregation between variables.

G.4. Runtime and Model Parameters

We conducted comparisons of runtime, inference time, and model parameter sizes between CauDiTS and baseline methods for unsupervised domain adaptation of multivariate time series. Baseline methods in unsupervised domain adaptation for time series data include primarily VRADA, CoDATS, AdvSKM, SASA, RAINCOAT, and CLUDA. In order to provide a fair and comprehensive evaluation, we report the average runtime per 100 training steps for each method, taking into account that the specific methods and application contexts may influence the overall runtime.

In this evaluation, we specifically considered the HHAR dataset, a large-scale dataset used in our experiments. The average runtime for each method (per 100 training steps) is presented in the first row of Table 7. The second row of Table 7 reports the logarithmically transformed inference time for each method. It can be seen that while our training time is significantly

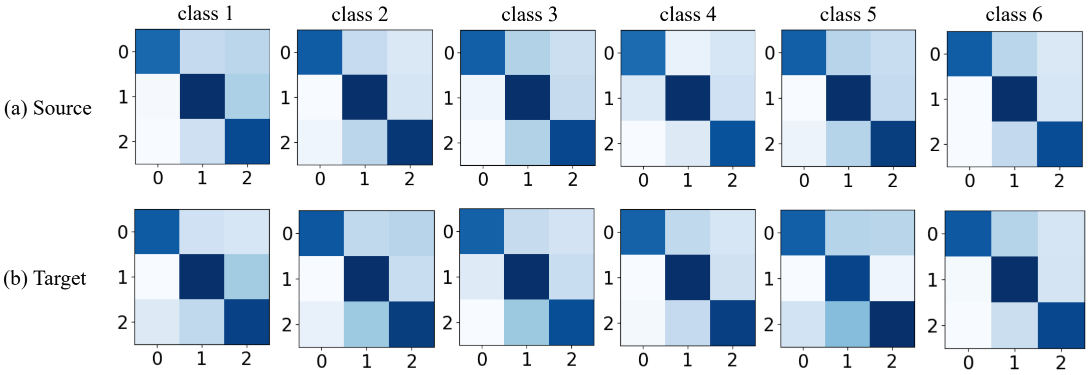


Figure 11: The visualizations of causal rationales adjacent matrix for the 26→2 in the WISDM. The deeper the color, the stronger the causal relationship.

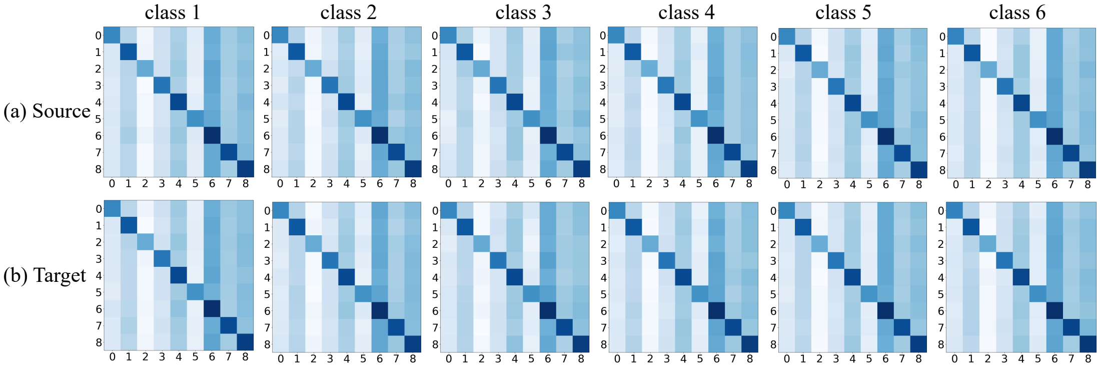


Figure 12: The visualizations of causal rationales adjacent matrix for 15→19 in the HAR. The deeper the color, the stronger the causal relationship.

longer than that of the other methods, our inference time remains comparable to or even superior to most of the baselines. CauDiTS exhibits superior log_flops (see Equation G.1) compared to the baselines due to its use of a structurally simple CNN as the domain-invariant representation extractor. In contrast, other methods rely on a relatively complex TCN structure to achieve satisfactory performance. In addition, during testing, CauDiTS selectively activates the query branch in the target domain and deactivates the remaining branches. This significantly reduces the time consumption.

$$\text{log\_flops} = \log\left(\frac{1}{\text{Inference time}}\right). \tag{G.1}$$

Furthermore, we conducted a comparative analysis of the CauDiTS parameter size against the baselines. The results, depicted in Table 7, indicate that CauDiTS shows a marginal increase in model parameter size but achieves a significant improvement in accuracy and Macro-F1, as shown in Table 5 and Table 6.

H. Ablation Study and Sensitivity Analysis

Contribution of Each Component. We conducted an ablation study of the model components using the benchmark datasets WISDM, HAR, HHAR and Boiler. The model is composed of difference elements, including w/o UDA, also known as source-only, where the model is trained exclusively on time-series samples from the source domain and then used for inference on samples from the target domain after training. $\mathcal{L}_{bs}$ denotes the basic loss for domain adaptation of time series and includes three widely used loss terms: supervised classification loss ($\mathcal{L}_{cls}$) on the source domain, unsupervised conditional entropy loss ($\mathcal{L}_{ucls}$) on the target domain, and domain classification loss ($\mathcal{L}_{dis}$). ARAD is proposed as an adaptive causal rationale disentanglement strategy, designed to effectively disentangle domain-specific

Table 5: The results of selected source ↦ target pairs in four benchmark datasets in terms of mean Macro-F1 over 10 independent runs. The best results are shown in **bold**, and the second-best results are underlined.

Dataset	Source ↦ Target	VRADA	CoDATS	AdvSKM	CAN	CDAN	DDC	DeepCORAL	DSAN	HoMM	MMDA	SASA	RAINCOAT	CLUDA	CauDiTS	↑
Boiler	1 ↦ 2	0.505	0.487	0.519	0.468	0.522	0.542	0.542	0.468	0.514	0.512	0.508	0.552	0.559	**0.5798**	+3.72%
	1 ↦ 3	0.664	0.660	0.739	0.890	0.718	0.923	0.929	0.934	0.933	0.580	0.909	0.935	0.929	**0.9632**	+3.02%
	2 ↦ 1	0.487	0.495	0.585	0.497	0.499	0.563	0.523	0.490	0.519	0.500	0.494	0.591	0.570	**0.6381**	+7.96%
	2 ↦ 3	0.410	0.398	0.481	0.398	0.398	0.484	0.408	0.398	0.405	0.398	0.400	0.498	0.499	**0.7251**	+45.31%
	3 ↦ 1	0.496	0.640	0.844	0.621	0.425	0.778	0.805	0.476	0.809	0.631	0.810	0.807	0.847	**0.9354**	+10.44%
	3 ↦ 2	0.491	0.417	0.557	0.499	0.434	0.561	0.557	0.496	0.489	0.499	0.537	0.523	0.535	**0.6219**	+10.86%
	Avg	0.509	0.525	0.621	0.562	0.599	0.642	0.627	0.544	0.612	0.520	0.610	0.651	0.657	**0.7439**	+13.23%
WISDM	12 ↦ 19	0.410	0.456	0.510	0.508	0.298	0.396	0.317	0.518	0.281	0.233	0.246	0.392	0.532	**0.5758**	+8.23%
	12 ↦ 7	0.437	0.612	0.655	0.636	0.546	0.632	0.486	0.574	0.442	0.539	0.633	0.684	0.678	**0.7960**	+16.37%
	18 ↦ 20	0.578	0.427	0.348	0.389	0.600	0.383	0.379	0.268	0.421	0.280	0.665	0.672	0.673	**0.7002**	+4.04%
	19 ↦ 2	0.615	0.403	0.460	0.327	0.312	0.459	0.501	0.428	0.522	0.306	0.496	0.432	0.458	**0.7557**	+22.88%
	2 ↦ 28	0.688	0.688	0.742	0.610	0.644	0.669	0.726	0.654	0.691	0.677	0.708	0.700	0.788	**0.7944**	+0.81%
	26 ↦ 2	0.517	0.598	0.463	0.362	0.404	0.414	0.618	0.424	0.519	0.453	0.689	0.472	0.701	**0.8098**	+15.52%
	28 ↦ 2	0.473	0.492	0.484	0.412	0.400	0.484	0.495	0.451	0.511	0.430	0.701	0.623	0.710	**0.7499**	+5.62%
	28 ↦ 20	0.672	0.578	0.557	0.655	0.605	0.571	0.620	0.615	0.699	0.537	0.835	0.849	0.703	**0.8512**	+0.26%
	7 ↦ 2	0.399	0.494	0.476	0.490	0.543	0.496	0.490	0.481	0.494	0.459	0.558	0.611	0.576	**0.6608**	+8.15%
	7 ↦ 26	0.308	0.405	0.416	0.395	0.344	0.412	0.396	0.401	0.406	0.385	0.391	0.424	0.403	**0.5249**	+23.80%
	Avg	0.510	0.515	0.511	0.479	0.469	0.492	0.503	0.482	0.498	0.430	0.592	0.586	0.622	**0.7219**	+16.06%
HAR	15 ↦ 19	0.657	0.663	0.664	0.593	0.696	0.658	0.708	0.831	0.686	0.656	0.957	0.940	0.957	**0.9657**	+0.91%
	18 ↦ 21	0.668	0.428	0.445	0.434	0.718	0.427	0.539	0.458	0.486	0.440	**1.000**	**1.000**	0.923	**1.0000**	+0.00%
	19 ↦ 25	0.737	0.381	0.359	0.640	0.768	0.360	0.535	0.754	0.397	0.348	0.560	0.956	0.932	**0.9831**	+2.83%
	19 ↦ 27	0.723	0.643	0.652	0.723	0.752	0.683	0.689	0.852	0.650	0.684	0.936	0.992	0.996	**1.0000**	+0.40%
	20 ↦ 6	0.773	0.603	0.576	0.725	0.796	0.529	0.666	0.759	0.627	0.641	0.827	0.903	**1.000**	**1.0000**	+0.00%
	23 ↦ 13	0.696	0.440	0.436	0.410	0.660	0.447	0.616	0.606	0.549	0.527	0.709	0.778	0.762	**0.8308**	+6.79%
	24 ↦ 22	0.749	0.714	0.726	0.772	0.756	0.710	0.647	0.726	0.768	0.722	**1.000**	0.765	0.983	**1.0000**	+0.00%
	25 ↦ 24	0.782	0.516	0.503	0.702	0.765	0.527	0.625	0.873	0.538	0.641	0.989	0.988	0.992	**0.9947**	+0.27%
	3 ↦ 20	0.671	0.853	0.847	0.549	0.769	0.852	0.828	0.757	0.860	0.784	0.961	0.869	0.968	**0.9833**	+1.58%
	13 ↦ 19	0.696	0.738	0.769	0.729	0.837	0.752	0.763	0.662	0.798	0.752	0.943	0.946	0.911	**0.9715**	+2.70%
	Avg	0.715	0.598	0.598	0.628	0.752	0.595	0.662	0.728	0.636	0.619	0.888	0.914	0.942	**0.9729**	+3.28%
HHAR	0 ↦ 2	0.536	0.598	0.628	0.667	0.611	0.605	0.569	0.205	0.627	0.612	0.699	0.627	0.710	**0.7204**	+1.46%
	1 ↦ 6	0.702	0.696	0.662	0.621	0.727	0.678	0.725	0.696	0.726	0.693	0.830	0.858	0.858	**0.8816**	+2.75%
	2 ↦ 4	0.415	0.320	0.219	0.294	0.431	0.231	0.305	0.143	0.230	0.192	0.447	0.523	0.526	**0.6457**	+22.76%
	4 ↦ 0	0.243	0.222	0.163	0.165	0.273	0.175	0.249	0.116	0.179	0.162	0.346	0.284	0.352	**0.4296**	+22.05%
	4 ↦ 1	0.545	0.469	0.466	0.523	0.667	0.456	0.461	0.488	0.607	0.517	0.775	0.799	0.751	**0.8209**	+2.74%
	5 ↦ 1	0.756	0.723	0.692	0.813	0.848	0.707	0.766	0.285	0.738	0.765	0.916	0.964	0.950	**0.9661**	+0.22%
	7 ↦ 1	0.583	0.528	0.338	0.524	0.412	0.280	0.483	0.278	0.461	0.367	0.814	0.825	0.875	**0.8806**	+0.64%
	7 ↦ 5	0.529	0.374	0.154	0.546	0.480	0.175	0.496	0.192	0.323	0.283	0.624	0.633	0.626	**0.6736**	+6.32%
	8 ↦ 3	0.818	0.734	0.692	0.845	0.943	0.719	0.872	0.564	0.836	0.936	0.939	0.955	0.944	**0.9615**	+0.68%
	8 ↦ 4	0.715	0.539	0.580	0.596	0.710	0.550	0.578	0.434	0.606	0.636	0.855	0.644	**0.891**	0.8645	-2.97%
	Avg	0.584	0.520	0.459	0.559	0.610	0.458	0.550	0.340	0.533	0.516	0.725	0.711	0.748	**0.7845**	+4.88%

Table 6: The results of selected source ↦ target pairs in four benchmark datasets in terms of mean accuracy over 10 independent runs. The best results are shown in **bold**, and the second-best results are underlined.

Dataset	Source ↦ Target	VRADA	CoDATS	AdvSKM	CAN	CDAN	DDC	DeepCORAL	DSAN	HoMM	MMDA	SASA	RAINCOAT	CLUDA	CauDiTS	↑
Boiler	1 ↦ 2	0.889	0.892	0.921	0.826	0.946	0.942	0.926	0.811	0.902	0.947	0.950	0.946	0.927	**0.9672**	+1.81%
	1 ↦ 3	0.766	0.764	0.744	0.897	0.794	0.929	0.944	0.942	0.938	0.725	0.923	0.931	0.889	**0.9675**	+2.49%
	2 ↦ 1	0.897	0.899	0.899	0.902	0.905	0.884	0.891	0.907	0.885	0.905	0.901	0.907	0.905	**0.9131**	+0.67%
	2 ↦ 3	0.654	0.661	0.688	0.661	0.661	0.628	0.664	0.661	0.663	0.661	0.662	0.661	0.661	**0.7997**	+16.24%
	3 ↦ 1	0.862	0.768	0.955	0.742	0.596	0.941	0.947	0.910	0.948	0.947	0.936	0.932	0.939	**0.9788**	+2.49%
	3 ↦ 2	0.930	0.704	0.982	0.906	0.987	0.984	0.981	0.986	0.943	0.985	0.942	0.944	0.942	**0.9883**	+0.13%
	Avg	0.833	0.781	0.865	0.822	0.815	0.885	0.892	0.869	0.880	0.862	0.886	0.887	0.877	**0.9358**	+5.50%
WISDM	12 ↦ 19	0.558	0.633	0.639	0.594	0.488	0.564	0.433	0.639	0.415	0.358	0.411	0.618	0.694	**0.7813**	+12.58%
	12 ↦ 7	0.708	0.721	0.742	0.588	0.771	0.692	0.592	0.625	0.546	0.679	0.750	0.785	0.792	**0.8500**	+7.32%
	18 ↦ 20	0.571	0.634	0.390	0.439	0.771	0.390	0.380	0.366	0.429	0.380	0.776	0.783	0.780	**0.8063**	+2.98%
	19 ↦ 2	0.644	0.395	0.434	0.322	0.346	0.459	0.473	0.366	0.488	0.385	0.575	0.361	0.561	**0.9063**	+40.73%
	2 ↦ 28	0.729	0.809	0.809	0.760	0.813	0.782	0.827	0.773	0.787	0.813	0.809	0.798	0.849	**0.8625**	+1.59%
	26 ↦ 2	0.683	0.727	0.620	0.580	0.615	0.600	0.737	0.605	0.702	0.634	0.852	0.595	0.863	**0.9125**	+5.73%
	28 ↦ 2	0.688	0.717	0.707	0.561	0.580	0.702	0.649	0.673	0.644	0.668	0.725	0.566	0.741	**0.7938**	+7.13%
	28 ↦ 20	0.741	0.741	0.707	0.673	0.776	0.727	0.737	0.746	0.790	0.722	0.887	**0.890**	0.820	0.8813	-0.98%
	7 ↦ 2	0.605	0.610	0.610	0.571	0.649	0.620	0.624	0.620	0.605	0.605	0.706	0.759	0.712	**0.7938**	+4.58%
	7 ↦ 26	0.693	0.702	0.702	0.717	0.722	0.717	0.683	0.698	0.698	0.712	0.703	0.729	0.727	**0.7875**	+8.02%
	Avg	0.662	0.669	0.636	0.580	0.653	0.625	0.613	0.611	0.610	0.596	0.719	0.688	0.754	**0.8375**	+11.07%
HAR	15 ↦ 19	0.756	0.733	0.741	0.685	0.759	0.733	0.759	0.874	0.748	0.726	0.953	0.950	**0.967**	0.9625	-0.47%
	18 ↦ 21	0.794	0.552	0.555	0.552	0.803	0.548	0.610	0.558	0.581	0.555	**1.000**	**1.000**	0.910	**1.0000**	+0.00%
	19 ↦ 25	0.768	0.468	0.452	0.661	0.771	0.455	0.590	0.774	0.487	0.448	0.575	0.963	0.932	**0.9875**	+2.54%
	19 ↦ 27	0.793	0.709	0.723	0.782	0.807	0.747	0.744	0.891	0.726	0.754	0.950	0.991	0.996	**1.0000**	+0.40%
	20 ↦ 6	0.808	0.661	0.641	0.747	0.820	0.608	0.686	0.784	0.673	0.694	0.853	0.919	**1.000**	**1.0000**	+0.00%
	23 ↦ 13	0.736	0.504	0.504	0.476	0.700	0.504	0.668	0.628	0.604	0.572	0.722	0.803	0.788	**0.8375**	+4.30%
	24 ↦ 22	0.837	0.820	0.833	0.820	0.837	0.808	0.743	0.808	0.853	0.829	**1.000**	0.844	0.988	**1.0000**	+0.00%
	25 ↦ 24	0.817	0.583	0.566	0.721	0.790	0.593	0.648	0.883	0.607	0.666	0.988	0.991	0.993	**0.9938**	+0.08%
	3 ↦ 20	0.752	0.874	0.878	0.652	0.815	0.885	0.848	0.804	0.874	0.815	0.965	0.938	0.967	**0.9813**	+1.48%
	13 ↦ 19	0.752	0.793	0.807	0.785	0.841	0.800	0.793	0.726	0.815	0.800	0.938	0.941	0.904	**0.9688**	+2.95%
	Avg	0.781	0.670	0.670	0.688	0.794	0.668	0.709	0.773	0.697	0.686	0.894	0.934	0.944	**0.9731**	+3.08%
HHAR	0 ↦ 2	0.593	0.650	0.681	0.721	0.676	0.659	0.618	0.292	0.680	0.671	0.710	0.689	0.726	**0.7777**	+7.12%
	1 ↦ 6	0.690	0.686	0.652	0.619	0.717	0.672	0.712	0.689	0.725	0.686	0.839	0.847	0.855	**0.8723**	+2.02%
	2 ↦ 4	0.476	0.381	0.291	0.391	0.472	0.304	0.332	0.229	0.332	0.238	0.537	0.582	0.585	**0.7241**	+23.78%
	4 ↦ 0	0.263	0.229	0.203	0.194	0.262	0.216	0.259	0.193	0.193	0.205	0.349	0.291	0.353	**0.4479**	+26.88%
	4 ↦ 1	0.558	0.501	0.494	0.549	0.690	0.502	0.482	0.504	0.628	0.551	0.814	0.823	0.774	**0.8281**	+0.62%
	5 ↦ 1	0.775	0.761	0.737	0.829	0.857	0.744	0.787	0.407	0.784	0.790	0.917	0.963	0.948	**0.9648**	+0.19%
	7 ↦ 1	0.575	0.551	0.426	0.534	0.413	0.378	0.511	0.366	0.496	0.415	0.867	0.830	0.875	**0.8866**	+1.32%
	7 ↦ 5	0.523	0.380	0.192	0.592	0.492	0.229	0.489	0.233	0.328	0.320	0.617	0.646	0.636	**0.6781**	+4.97%
	8 ↦ 3	0.813	0.766	0.748	0.860	0.942	0.763	0.869	0.602	0.844	0.934	0.945	0.954	0.942	**0.9598**	+0.61%
	8 ↦ 4	0.720	0.601	0.650	0.660	0.712	0.629	0.618	0.516	0.658	0.701	0.880	0.660	**0.896**	0.8616	-3.84%
	Avg	0.599	0.551	0.508	0.595	0.623	0.510	0.568	0.403	0.567	0.551	0.748	0.729	0.759	**0.8001**	+5.42%

Table 7: The training time, inference time and parameter size of each method.

Method Index	VRADA	CoDATS	AdvSKM	SASA	RAINCOAT	CLUDA	CauDiTS
Training Time (s)	4.025	4.807	5.375	1.382	1.921	7.256	18.056
Inference Time (log_flops)	5.521	4.402	4.407	7.528	7.501	4.348	6.716
Parameters (M)	0.416	0.621	0.876	0.948	1.205	1.239	2.195

non-causal correlations and domain-common causal rationales from complex inter-variable interrelationships. Furthermore, $\mathcal{L}_{con}$ represents the in-domain intra-class embeddings contrastive loss. Additionally, $\mathcal{L}_{cpc}$ and $\mathcal{L}_{dcls}$ are novel losses specific to CauDiTS, encompassing causal prototype consistency loss and domain intervention causally invariant classification loss. We present the average prediction accuracy for each dataset, computed over 10 randomly selected source→target pairs across 10 independent replicate experiments. The results are detailed in Table 8.

Table 8 illustrates a positive correlation between the increasing number of integrated components and the gradual improvement in model accuracy. This observation underscores the crucial role of each loss term as an indispensable components within CauDiTS, contributing significantly to the overall improvement of model accuracy. Particularly noteworthy is the discernible impact of incorporating the ARAD module, resulting in a significant average accuracy improvement of approximately 8% across all datasets, compared to exclusive reliance on $\mathcal{L}_{bs}$. Furthermore, the introduction of the loss terms $\mathcal{L}_{cpc}$ and $\mathcal{L}_{dcls}$ also demonstrates a noteworthy increase in model accuracy.

Analysis of Different Number of N_r. When calculating the causal prototype $\mathbb{P}$, N_r represents the number of multivariate time series data randomly sampled for each class from the dataset $\mathcal{D}^s$. The choice of N_r in this paper should strike a balance, neither too high nor too low. A low value may result in a lack of representativeness for the calculated causal prototypes, while a high value introduces additional sampling noise and increases computational complexity. Therefore, we investigate the impact of different values of N_r on the model accuracy, and the results are presented in Table 9. The experiments are primarily conducted on the HHAR and Boiler datasets, as these are two large-scale datasets. WISDM and HAR are two small-scale datasets where the number of time series samples allocated to each class in each domain is even much smaller than the batch size. Therefore, we do not impose N_r value restrictions on the WISDM and HAR datasets when calculating causal prototypes $\mathbb{P}$.

For the Boiler dataset, we set the range of N_r to be between 150 and 500, while for the HHAR dataset, the search range is between 25 and 200. Analyzing the results from Table 9, it becomes evident that both datasets perform optimally when N_r is chosen at the mid-point of the search range. Additionally, we observe that smaller values of N_r generally yield better results overall.

Robustness Analysis for Irregular Multivariate Time Series. Most existing methods assume a regular and structured nature of the input data. However, their effectiveness can be significantly reduced in the presence of missing random variables, random data, and non-uniform sampling frequencies in multivariate time series data. To validate the robustness of the domain adaptation of CauDiTS to complex and noisy multivariate time series data, experiments are performed on complex irregular time series data. Specifically, CauDiTS incorporates random data augmentation during model training, generating two augmented views (details are shown in D.1). This augmentation strategy inherently covers scenarios with missing random variable, random data, and uneven sampling frequencies. Consequently, we further investigate the domain adaptation capability of CauDiTS for irregular multivariate time series data and compare it with baselines, as shown in Table 10. In this experiment, we systematically apply random data augmentation to the test samples from the target domain. The subsequent evaluation involves measuring the mean prediction accuracy and the mean Macro-F1 scores of the trained models across all methods for the augmented time series samples.

I. More Results on Other Dataset

We conduct an additional experiment using the large-scale Wearable Stress and Affect Detection (WESAD) dataset (Schmidt et al., 2018). WESAD is a dataset tailored for wearable stress and affect detectiion, incorporating physiological and motion data collected from both wrist-worn (RespiBAN) and chest-worn (Empatica E4) devices during a controlled lab study involving 15 subjects. The dataset comprises 63,000,000 samples. The RespiBAN device provides the following sensor data: electrocardiogram (ECG), electrodermal activity (EDA), electromyogram (EMG), respiration, body temperature, and three-axis acceleration. All signals are sampled at 700 Hz. The Empatica E4 device provides the following sensor data: blood volume pulse (BVP, 64 Hz), electrodermal activity (EDA, 4 Hz), body temperature (4 Hz), and three-axis acceleration (32 Hz).

For our experiment, We utilize the sensor modalities of chest-worn devices (RespiBAN) and split the samples into three parts: training, testing and validation sets, with a ratio of 70:15:15. The window size τ_{max} for WESAD is set to 200, with a step size of 100, resulting in a 50% overlap between two multivariate time series.

WESAD is a complex, large-scale dataset with relatively low performance across all baselines. However, CauDiTS

Table 8: Ablation studies of each component of CauDiTS. The mean accuracy of selected source→target pairs from each dataset was reported over 10 independent runs.

Dataset	Source ↦ Target	Mean Accuracy (%)					
Component	w/o UDA	✓					
	$\mathcal{L}_{bs}$		✓	✓	✓	✓	✓
	ARAD			✓	✓	✓	✓
	$\mathcal{L}_{con}$				✓	✓	✓
	$\mathcal{L}_{cpc}$					✓	✓
	$\mathcal{L}_{dcls}$						✓
Boiler	1 ↦ 2	0.7616	0.8198	0.8745	0.9181	0.9256	0.9672
	1 ↦ 3	0.7499	0.8640	0.8719	0.9330	0.9346	0.9675
	2 ↦ 1	0.6956	0.8266	0.8677	0.8867	0.9098	0.9131
	2 ↦ 3	0.4433	0.4964	0.6510	0.6610	0.6745	0.7997
	3 ↦ 1	0.7668	0.8408	0.9474	0.9654	0.9674	0.9788
	3 ↦ 2	0.7472	0.8518	0.8932	0.9231	0.9653	0.9883
	Avg	0.6941	0.7832	0.8510	0.8812	0.8962	0.9358
WISDM	12 ↦ 19	0.7438	0.7438	0.7519	0.7594	0.7687	0.7813
	12 ↦ 7	0.6375	0.7188	0.7250	0.7344	0.7438	0.8500
	18 ↦ 20	0.5437	0.6500	0.6844	0.7513	0.7750	0.8063
	19 ↦ 2	0.4438	0.4625	0.7250	0.8187	0.8750	0.9063
	2 ↦ 28	0.7375	0.7438	0.7875	0.7875	0.8125	0.8625
	26 ↦ 2	0.6813	0.7813	0.8094	0.8438	0.8875	0.9125
	28 ↦ 2	0.7013	0.7187	0.7437	0.7550	0.7813	0.7938
	28 ↦ 20	0.6250	0.6629	0.6938	0.8000	0.8205	0.8813
	7 ↦ 2	0.6875	0.6750	0.7406	0.7675	0.7812	0.7938
	7 ↦ 26	0.6188	0.6437	0.7344	0.7438	0.7531	0.7875
	Avg	0.6420	0.6800	0.7396	0.7761	0.7999	0.8375
HAR	15 ↦ 19	0.7123	0.7681	0.8245	0.8687	0.9375	0.9625
	18 ↦ 21	0.5688	0.6687	0.8719	0.9288	0.9625	1.0000
	19 ↦ 25	0.4687	0.5813	0.8469	0.9375	0.9428	0.9875
	19 ↦ 27	0.7625	0.8438	0.8938	0.9687	0.9844	1.0000
	20 ↦ 6	0.6875	0.7062	0.8563	0.9313	0.9687	1.0000
	23 ↦ 13	0.4500	0.5233	0.6781	0.7500	0.7688	0.8375
	24 ↦ 22	0.8000	0.8135	0.8968	1.0000	1.0000	1.0000
	25 ↦ 24	0.5813	0.7413	0.9000	0.9541	0.9625	0.9938
	3 ↦ 20	0.7187	0.7188	0.8593	0.9038	0.9500	0.9813
	13 ↦ 19	0.7125	0.7937	0.8345	0.8938	0.9125	0.9688
	Avg	0.6462	0.7159	0.8462	0.9137	0.9390	0.9731
HHAR	0 ↦ 2	0.6469	0.6607	0.7067	0.6875	0.6991	0.7777
	1 ↦ 6	0.6393	0.6875	0.7674	0.7759	0.8571	0.8723
	2 ↦ 4	0.3665	0.4107	0.5915	0.6839	0.6897	0.7241
	4 ↦ 0	0.1830	0.1902	0.2710	0.3804	0.4018	0.4479
	4 ↦ 1	0.5672	0.6469	0.6813	0.7344	0.7563	0.8281
	5 ↦ 1	0.7262	0.7609	0.8008	0.8633	0.9094	0.9648
	7 ↦ 1	0.4016	0.5203	0.6054	0.8102	0.8203	0.8866
	7 ↦ 5	0.2992	0.3468	0.4566	0.5898	0.6679	0.6781
	8 ↦ 3	0.6728	0.6705	0.7491	0.8225	0.9250	0.9598
	8 ↦ 4	0.6321	0.6563	0.6955	0.7937	0.8125	0.8616
	Avg	0.5135	0.5551	0.6325	0.7142	0.7539	0.8001

Table 9: Analysis of the impact of N_r. Report the mean Macro-F1 over 5 random initializations.

Dataset	Source ↦ Target	150	200	300	400	500
Boiler	1 ↦ 2	0.5715	0.5804	0.5798	0.5523	0.5362
	2 ↦ 1	0.6117	0.6279	0.6384	0.6244	0.6240
	2 ↦ 3	0.7199	0.7311	0.7248	0.7211	0.7153
	3 ↦ 2	0.6210	0.6218	0.6197	0.6143	0.6094
	Avg	0.6310	0.6403	0.6407	0.6280	0.6212
Datasets	Source ↦ Target	25	50	100	150	200
HHAR	0 ↦ 2	0.7054	0.7146	0.7208	0.6534	0.6753
	2 ↦ 4	0.5907	0.6270	0.6772	0.6352	0.6189
	4 ↦ 0	0.4188	0.4248	0.4779	0.4123	0.3561
	4 ↦ 1	0.8143	0.8229	0.8244	0.8127	0.8031
	7 ↦ 5	0.6386	0.6435	0.6736	0.6694	0.6632
	Avg	0.6336	0.6466	0.6748	0.6366	0.6233

Table 10: Robustness analysis for irregular multivariate time series. Report the mean Macro-F1 over 5 random initializations.

Dataset	Source ↦ Target	VRADA	CoDATS	AdvSKM	SASA	RAINCOAT	CLUDA	CauDiTS
Boiler	1 ↦ 2	0.3940	0.3941	0.4144	0.4874	0.5117	0.5155	**0.5304**
	2 ↦ 1	0.4873	0.4697	0.5093	0.4039	0.5487	**0.5969**	0.5764
	2 ↦ 3	0.4096	0.3822	0.4037	0.3369	0.4953	0.4000	**0.5979**
	3 ↦ 1	0.4003	0.5814	0.7450	0.6350	0.6474	0.7097	**0.7730**
	3 ↦ 2	0.3937	0.3044	0.4198	0.4872	0.4664	0.4096	**0.5182**
	Avg	0.4170	0.4264	0.4984	0.4701	0.5339	0.5263	**0.5992**
WISDM	12 ↦ 19	0.3113	0.1722	0.2963	0.2358	0.3961	0.4803	**0.5230**
	12 ↦ 7	0.3064	0.3695	0.4133	0.5761	0.6783	0.4839	**0.6967**
	19 ↦2	0.3764	0.3750	0.3208	0.4847	0.4525	0.3683	**0.5384**
	26 ↦ 2	0.4484	0.5085	0.3674	0.5505	0.4478	0.5894	**0.6733**
	28 ↦ 2	0.5675	0.5026	0.3530	0.6007	0.6018	0.5751	**0.7194**
	Avg	0.4020	0.3856	0.3502	0.4896	0.5153	0.4994	**0.6302**
HAR	15 ↦ 19	0.4514	0.6285	0.5973	0.7968	0.7869	0.7712	**0.9111**
	19 ↦ 25	0.4868	0.3026	0.2970	0.5206	0.8139	**0.8833**	0.8103
	23 ↦ 13	0.4778	0.3344	0.3885	0.6490	0.4944	0.6800	**0.7360**
	25 ↦ 24	0.5240	0.3627	0.3109	0.8067	0.7103	0.9634	**0.9800**
	3 ↦ 20	0.5038	0.6719	0.4900	0.7190	0.8255	0.6524	**0.8609**
	Avg	0.4888	0.4600	0.4167	0.6984	0.7262	0.7901	**0.8597**
HHAR	0 ↦ 2	0.4898	0.4197	0.4767	0.6453	0.5357	0.6479	**0.6686**
	2 ↦ 4	0.3594	0.1528	0.1703	0.4305	0.4866	0.4518	**0.5085**
	4 ↦ 0	0.1588	0.1593	0.1569	0.3144	0.2673	0.2798	**0.3422**
	4 ↦ 1	0.5112	0.3606	0.2615	0.6944	0.7271	0.6158	**0.7643**
	7 ↦ 5	0.5097	0.2602	0.1400	0.6069	0.5746	0.6037	**0.6521**
	Avg	0.4076	0.2705	0.2411	0.5383	0.5183	0.5198	**0.5871**

Table 11: Robustness analysis for irregular multivariate time series. Report the mean accuracy over 5 random initializations.

Dataset	Source ↦ Target	VRADA	CoDATS	AdvSKM	SASA	RAINCOAT	CLUDA	CauDiTS
Boiler	1 ↦ 2	0.7984	0.7134	0.7326	0.8517	0.8414	0.8417	**0.8556**
	2 ↦ 1	0.6954	0.7355	0.7679	0.8934	0.8365	0.9045	**0.9099**
	2 ↦ 3	0.6532	0.6579	0.6593	0.5935	0.6505	0.6606	**0.6610**
	3 ↦ 1	0.8600	0.7138	0.9294	0.8438	0.9097	0.9133	**0.9419**
	3 ↦ 2	0.8267	0.6596	0.8569	0.9007	0.8983	0.9029	**0.9156**
	Avg	0.7667	0.6960	0.7892	0.8166	0.8273	0.8446	**0.8568**
WISDM	12 ↦ 19	0.4688	0.2500	0.4062	0.3594	0.5909	0.6406	**0.7250**
	12 ↦ 7	0.6500	0.6812	0.5938	0.6969	0.7625	0.6250	**0.7813**
	19 ↦2	0.5312	0.3312	0.3438	0.5531	0.4049	0.4375	**0.6563**
	26 ↦ 2	0.5875	0.6500	0.5938	0.8344	0.5756	0.8125	**0.8556**
	28 ↦ 2	0.6500	0.6188	0.5938	0.5625	0.5122	0.6312	**0.7375**
	Avg	0.5775	0.5062	0.5063	0.6013	0.5692	0.6294	**0.7511**
HAR	15 ↦ 19	0.4688	0.6562	0.6562	0.7844	0.7938	0.8438	**0.9063**
	19 ↦ 25	0.5312	0.3750	0.3438	0.5343	0.8250	**0.8750**	0.8500
	23 ↦ 13	0.5625	0.4625	0.4625	0.6656	0.5375	0.7125	**0.7438**
	25 ↦ 24	0.5938	0.4375	0.4062	0.8094	0.7500	0.9688	**0.9814**
	3 ↦ 20	0.6250	0.7500	0.5938	0.7438	0.8625	0.7188	**0.8750**
	Avg	0.5563	0.5362	0.4925	0.7075	0.7538	0.8238	**0.8713**
HHAR	0 ↦ 2	0.5357	0.4420	0.5179	0.6705	0.5911	0.6589	**0.6786**
	2 ↦ 4	0.4446	0.1473	0.2812	0.4098	0.5313	0.5027	**0.5938**
	4 ↦ 0	0.1089	0.1562	0.1723	0.3214	0.2759	0.2946	**0.3330**
	4 ↦ 1	0.3491	0.3594	0.3125	0.7211	0.7344	0.6562	**0.7734**
	7 ↦ 5	0.2536	0.2461	0.1758	0.5976	0.5195	0.4992	**0.5484**
	Avg	0.3384	0.2702	0.2919	0.5441	0.5304	0.5223	**0.5854**

demonstrates a significant improvement over other methods and achieves optimal performance, as shown in Table 12.

Table 12: The results of selected source ↦ target pairs on WESAD dataset in terms of mean accuracy (Acc) and mean Macro-F1 (MF1) over 5 independent runs. The best results are shown in **bold**, and the second-best results are underlined.

	VRADA		CoDATS		AdvSKM		SASA		RAINCOAT		CLUDA		CauDiTS	
Method	Acc	MF1	Acc	MF1	Acc	MF1	Acc	MF1	Acc	MF1	Acc	MF1	Acc	MF1
2 ↦ 3	0.3613	0.1588	0.4104	0.2168	0.2637	0.1884	0.4251	0.2743	0.4842	0.3056	0.3815	0.3046	**0.4915**	**0.3650**
3 ↦ 8	0.3633	0.1995	0.2655	0.1109	0.2634	0.1063	0.2628	0.1048	0.3388	0.1973	0.3201	0.1644	**0.4889**	**0.3871**
4 ↦ 10	0.3418	0.1838	0.2708	0.1171	0.2790	0.1272	0.2725	0.1189	0.6704	0.5291	0.5475	0.4194	**0.7675**	**0.6843**
11 ↦ 5	0.3529	0.1548	0.8551	0.8224	0.6647	0.5425	0.8458	0.8183	0.8271	0.7733	0.8276	0.8100	**0.8818**	**0.8346**
6 ↦ 17	0.3522	0.1475	0.6315	0.5056	0.7583	0.6129	0.7147	0.5659	0.6957	0.5255	0.7751	0.5893	**0.7974**	**0.6347**
Avg	0.3543	0.1689	0.4867	0.3546	0.4458	0.3155	0.5042	0.3764	0.6032	0.4662	0.6504	0.5375	**0.6854**	**0.5811**

Algorithm 1 Training and inference of CauDiTS

```
Input: dataset $\mathcal{D}^s$ and $\mathcal{D}^t$; epochs E;
Initialization: parameter $\Theta_H$ for LSTM, parameters $\Theta_{ARAD} = \{W_1, W_2, b_1, b_2, v, W_{\bar{M}}, b_{\bar{M}}\}$ for ARAD, parameters $\Theta_{R_c} = \{W_3, W_4, W_5, b_5, W_6, b_6\}$ for $GCN_c(\cdot)$, parameters $\Theta_{R_o}$ for $GCN_o(\cdot)$, parameter $\Theta_{Z_c}$ and $\Theta_{Z_o}$ for $F_c(\cdot)$ and $F_o(\cdot)$, parameter $\Theta_{Y_c}$ for $\Phi(\cdot)$, parameter $\Theta_{Y_o}$ for $D(\cdot)$, parameter $\Theta_E$ for $\mathcal{H}(\cdot)$.

Training:
for $i \leftarrow 1$ to E do
  best_loss $= 1000000$
  Compute: causal prototype $\mathbb{P}$ using Equation (9)
  while $\mathcal{D}^t$ not exhausted do
    Sample $X^s$ and $Y^s$ from $\mathcal{D}^s$, $X^t$ from $\mathcal{D}^t$
    Multivariate time series augmentation for $X^s$ and $X^t$.
    $\{\tilde{X}^s, \hat{X}^s\} \leftarrow aug(X^s)$          // Source domain
    $\{\tilde{X}^t, \hat{X}^t\} \leftarrow aug(X^t)$          // Target domain
    Update hidden state. (use Equation (2))
    $\tilde{H}^s =$ LSTM$(\tilde{X}^s)$, $\hat{H}^s =$ LSTM$(\hat{X}^s)$          // Source domain
    $\tilde{H}^t =$ LSTM$(\tilde{X}^t)$, $\hat{H}^t =$ LSTM$(\hat{X}^t)$          // Target domain
    Calculate attention coefficient. (use Equation (3))
    $\tilde{\mathcal{I}}^s = Atten(\tilde{H}^s), \hat{\mathcal{I}}^s = Atten(\hat{H}^s)$          // Source domain
    $\tilde{\mathcal{I}}^t = Atten(\tilde{H}^t), \hat{\mathcal{I}}^t = Atten(\hat{H}^t)$          // Target domain
    Disentangle causal rationales. (use Equation (4))
    $\tilde{A}^s, \tilde{B}^s = ARAD(\tilde{\mathcal{I}}^s), \hat{A}^s, \hat{B}^s = ARAD(\hat{\mathcal{I}}^s)$          // Source domain
    $\tilde{A}^t, \tilde{B}^t = ARAD(\tilde{\mathcal{I}}^t), \hat{A}^t, \hat{B}^t = ARAD(\hat{\mathcal{I}}^t)$          // Target domain
    Generate node representations. (use Equation (5))
    $\tilde{R}^s_c = GCNc(\tilde{A}^s, \tilde{H}^s)$ and $\hat{R}^s_c = GCNc(\hat{A}^s, \hat{H}^s)$, $\tilde{R}^s_o = GCNo(\tilde{B}^s, \tilde{H}^s)$ and $\hat{R}^s_o = GCNc(\hat{B}^s, \hat{H}^s)$          // Source domain
    $\tilde{R}^t_c = GCNc(\tilde{A}^t, \tilde{H}^t)$ and $\hat{R}^t_c = GCNc(\hat{A}^t, \hat{H}^t)$, $\tilde{R}^t_o = GCNo(\tilde{B}^t, \tilde{H}^t)$ and $\hat{R}^t_o = GCNc(\hat{B}^t, \hat{H}^t)$          // Target domain
    Extract domain-invariant representation
    $\tilde{Z}^s_c = F_c(\tilde{R}^s_c)$          // Source domain
    $\tilde{Z}^t_c = F_c(\tilde{R}^t_c)$          // Target domain
    Predict class label
    $\tilde{Y}^s_c = \Phi(\tilde{Z}^s_c)$          // Source domain
    $\tilde{Y}^t_c = \Phi(\tilde{Z}^t_c)$          // Target domain
    Extract domain-specific representation
    $\tilde{Z}^s_o = F_o(\tilde{R}^s_o)$, $\hat{Z}^s_o = F_o(\hat{R}^s_o)$          // Source domain
    $\tilde{Z}^t_o = F_o(\tilde{R}^t_o)$, $\hat{Z}^t_o = F_o(\hat{R}^t_o)$          // Target domain
    Predict domain label
    $\tilde{Y}^s_o = D(\tilde{Z}^s_o)$, $\hat{Y}^s_o = D(\hat{Z}^s_o)$          // Source domain
    $\tilde{Y}^t_o = D(\tilde{Z}^t_o)$, $\hat{Y}^t_o = D(\hat{Z}^t_o)$          // Target domain

    Compute: $\mathcal{L}_{bs}$ using Equation (8), $\mathcal{L}_{con}$ using Equation (17), $\mathcal{L}_{cpc}$ using Equation (12), $\mathcal{L}_{dcls}$ using Equation (16)
    $\mathcal{L} = \mathcal{L}_{bs} + \mathcal{L}_{cpc} + \gamma_1 \cdot \mathcal{L}_{dcls} + \gamma_2 \cdot \mathcal{L}^s_{con} + \gamma_3 \cdot \mathcal{L}^t_{con}$
    Update $\Theta_H, \Theta_{ARAD}, \Theta_{P_c}, \Theta_{P_o}, \Theta_{Z_c}, \Theta_{Z_o}, \Theta_{Y_c}, \Theta_{Y_o}$ and $\Theta_E$ with $\bigtriangledown \mathcal{L}$.
    if $\mathcal{L} <$ best_loss then
      Update causal prototype $\mathbb{P}$ using Equation (9)
      best_loss = $\mathcal{L}$
    end if
  end while
end for


Inference:
while $\mathcal{D}^{test}$ not exhausted do
  Sample $X^t$ from $\mathcal{D}^{test}$
  $\tilde{H}^t =$ LSTM$(\tilde{X}^t)$
  $\tilde{\mathcal{I}}^t = Atten(\tilde{H}^t)$
  $\tilde{A}^t, \tilde{B}^t = ARAD(\tilde{\mathcal{I}}^t)$
  $\tilde{R}^t_c = GCNc(\tilde{A}^t, \tilde{H}^t)$
  $\tilde{Z}^t_c = F_c(\tilde{R}^t_c)$
  $\tilde{Y}^t_c = \Phi(\tilde{Z}^t_c)$
end while
```