Disentangling Domain and General Representations for Time Series Classification

Youmin Chen¹, Xinyu Yan¹, Yang Yang¹*, Jianfeng Zhang², Jing Zhang³, Lujia Pan² and Juren Li¹
¹Zhejiang University
²Huawei Noah’s Ark Lab
³Renmin University of China
{youminchen, yangya, jrl}@zju.edu.cn, {zhangjianfeng3, panlujia}@huawei.com,
zhang-jing@ruc.edu.cn, yan512118546@gmail.com

Abstract

Modeling time series data has become a very attractive research topic due to its wide application, such as human activity recognition, financial forecasting and sensor-based automatic system monitoring. Recently deep learning models have shown great advances in modeling the time series data, but they heavily depend on a large amount of labeled data. To avoid costly labeling, this paper explores domain adaptation from a labeled source domain to the unlabeled target domain on time series data. To achieve the goal, we propose a disentangled representation learning framework named CADT to disentangle the domain-invariant features from the domain-specific ones. Particularly, CADT is injected with a novel class-wise hypersphere loss to improve the generalization of the classifier from the source domain to the target domain. Intuitively, it restricts the source data of the same class within the same hypersphere and minimizes the radius of it, which in turn enlarges the margin between different classes and makes the decision boundary of both domains easier. We further devise several kinds of domain-preserving data augmentation methods to better capture the domain-specific patterns. Extensive experiments on two public datasets and three real-world applications demonstrate the effectiveness of the proposed model against several state-of-the-art baselines.

1 Introduction

Time series data is prevalent in real-world applications like human activity recognition, financial forecasting, and manufacturing sensor monitoring. Recently, deep learning models such as long short-term memory networks (LSTM) [Hochreiter and Schmidhuber, 1997] and 1-dimensional convolution networks (Conv1D) [Bai et al., 2018] have shown promising advances in time series classification and forecasting.

Using the mentioned deep learning models effectively typically requires a large amount of labeled data. Labeling time series data demands more manual effort from experts due to the consecutive property of time series data [Lin and Jung, 2017; Wang et al., 2018; Kim and Jeong, 2021]. Fig. 1(a) demonstrates the challenge of identifying electricity theft, highlighting the extensive effort required for labeling. Detecting such behavior often demands prolonged on-site investigations by domain experts. Applying models trained on labeled data from one city to others is complex due to variations in electricity consumption patterns influenced by economic status, weather conditions, and electricity prices, leading to potential misclassification or failure to detect electricity thieves.

In view of the great labeling cost and the significant domain gap of time series data, we propose to study unsupervised domain adaptation (UDA) for time series data. UDA aims to perform the task on the unlabeled target domain by leveraging the labeled data of a source domain. Compared to existing pretraining methods, UDA does not require labeled target domain datasets for fine-tuning. Additionally, compared to zero-shot pretraining, UDA can better leverage the correlation between the source and target domains, thereby achieving improved performance. Back to our example, if the model can mitigate the distribution divergence between the source domain and target domain. Intuitively, electricity thieves share some common characteristics no matter in which city they live. For example, their overall electric power consumption is higher than that of normal residents which can be treated as domain-invariant patterns across domains. However, thieves of different areas have their specific electricity consumption patterns and theft modes which are domain-specific patterns. Thus, how to capture the domain-invariant patterns across the source and the target domain with remarkable domain-specific patterns is crucial.

To achieve the goal, we disentangle the representation learning of the domain-invariant and the domain-specific patterns. The motivation is depicted through the electricity theft example in Fig. 1(b), showcasing the variations in average daily electricity consumption over a month for regular users and electricity thieves in two distinct Chinese cities. City A, with an adequate label set, serves as the source domain, while city B represents the target domain devoid of any labels. No matter what users we consider, we can see that city A consumes much more electrical power than city B. If the difference between such domain-specific patterns is ignored, we might derive the wrong domain-invariant pattern that users who consume electricity of more than 13 Kilowatt
Formally, we propose a **Class-wise Adversarial learning framework** for Domain adaptation in Time-series data (CADT), to learn the domain-invariant representations by the supervision of the class labels. CADT is equipped with *coupled interactive networks* to enable the disentangled representation learning, which extracts the domain-specific features and the domain-invariant features by different networks respectively. An additional class-wise hypersphere loss is injected into the traditional adversarial learning loss to enhance the discrimination ability of the task classifier. To capture various domain-specific time series patterns, we devise several kinds of *domain-preserving data augmentation* methods. Overall, our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to explore the disentangled domain-invariant and domain-specific representation learning for UDA in time series data.
- We propose an adversarial learning framework, CADT, equipped with coupled interactive networks for disentangled learning. It includes the class-wise hypersphere loss to improve model generalization and is enhanced by data augmentation to capture domain-specific patterns.
- We conduct extensive experiments on two public datasets and two real-world application datasets (electricity theft detection and electric vehicle accident prediction). The results show that CADT consistently outperforms several state-of-the-art baseline methods.

2 Related Works

Unsupervised domain adaptation is a specific instance of transfer learning [Zhuang et al., 2019; Long et al., 2015] with labeled source domain data and unlabeled target domain data, where the mapping representation spaces are shared but their marginal distributions remain different due to the dataset shift [Yang et al., 2021; Yan et al., 2021; Liu et al., 2019]. Abundant previous works survey on this topic in computer vision (CV) [Patel et al., 2015; Sun et al., 2015] and natural language processing (NLP) applications [Glorot et al., 2011; Pan and Yang, 2010]. Typically, the mapping representation involves minimizing the measure of discrepancy between source features and target features [Zellinger et al., 2017; Lee et al., 2019; Li et al., 2022]. With the remarkable success in deep learning, many studies utilize the deep neural networks as a feature extractor to learn representative domain-invariant representations [Sun and Saenko, 2016; Li et al., 2021; Long et al., 2016]. Inspired by the generative adversarial networks (GANs) [Goodfellow et al., 2014;
Arjovsky et al., 2017], many UDA approaches [Tzeng et al., 2017; Ganin and Lempitsky, 2015; Bousmalis et al., 2017; Zhang et al., 2021; Qiang et al., 2021] are proposed to learn domain-invariant representation based on adversarial learning. Specifically, these approaches consist of a feature extractor, a domain discriminator, and a label predictor. The feature extractor acts as the generator in GAN, whose goal is to produce domain-invariant representation in the mapping space to fool the domain discriminator. The domain discriminator is trained to detect if the extracted feature is from the source or the target domain. The domain adaptation and deep feature learning are within one training process. Furthermore, the label predictor predicts the label of training source samples based on features that are both invariant to domains and discriminative. Although time series data is abundant in real-world applications [Liu et al., 2022; Fulcher and Jones, 2014], few works focused on UDA for temporal data. Specifically, considering the sequential structure of temporal data, previous works [Purushotham et al., 2017; Tonutti et al., 2019] leveraged recurrent neural network as the feature extractor. CoDATS [Wilson et al., 2020] leverage 1-dimensional convolutional neural networks as the feature extractor. CoDATS achieves competitive performances on several human activity recognition tasks. However, these methods have vulnerable domain-invariant representations due to shared feature extractor and adversarial architecture. Additionally, a particular design for time series data is also essential due to its consecutive property.

3 Our Approach

In this section, we propose a novel CADT to fulfill domain adaptation for time series data.

Problem definition. In domain adaptation setting, there are two different data distributions: \( P \) from the source domain and \( Q \) from the target domain. We are given by \( n_s \) labeled samples \( \hat{P} = \{(x^t_i, y^s_i)\}_{i=1}^{n_s} \) drawn from the source distribution where \( y^s_i \in \{1, \cdots, k\} \) is the label of \( k \)-class classification, and \( n_t \) unlabeled samples \( \hat{Q} = \{x^t_i\}_{i=1}^{n_t} \) drawn from the target distribution. Our goal is to predict the label \( \hat{y}^t \) of each sample \( x^t \in \hat{Q} \).

3.1 Model Overview

We illustrate the architecture of our method CADT in Fig. 2. The cornerstone of CADT is the coupled interactive networks that aims to learn the domain-invariant representations and the domain-specific representations respectively for instances from two domains. The two networks, therefore, interact with each other to exchange their information. Each of the two interactive networks will output two representations \( h_s \) and \( h_t \), corresponding to the instances from the source domain and the target domain respectively. To achieve the goal of disentangling the domain-invariant representations from domain-specific ones, we design a domain discriminator \( \mathcal{D} \) and a domain classifier \( \mathcal{C} \) to guide the interactive networks. The domain discriminator ensures that the target domain-invariant representations cannot be distinguished from the source domain-invariant representations. In contrast, the domain classifier aims to preserve the characteristics of each domain within their respective domain-specific representations. We enhance the discrimination ability by using a task classifier \( T \) to leverage label information from the source domain and equipping the domain discriminator with a set of class-wise hyperspheres to of the task classifier. To better capture the plentiful domain-specific patterns, we design several domain-preserving data augmentations.

3.2 Coupled Interactive Networks

Most existing time-series domain adaptation models adopt a deep neural network as the single extractor to directly capture domain-invariant features [Wilson et al., 2020; Purushotham et al., 2018]. However, these methods have vulnerable domain-invariant representations due to shared feature extractor and adversarial architecture. Additionally, a particular design for time series data is also essential due to its consecutive property.
et al., 2017; Tonutti et al., 2019]. These methods neglect domain-specific information, which is also very important for domain adaptation as illustrated in Fig. 1(b). In order to extract both the domain-invariant and domain-specific features, we propose a couple of interactive networks.

Generally, domain-invariant and domain-specific features shall reflect different characteristics of the given time series data. In particular, domain-invariant features tend to describe the “high-level” patterns, which underlie the superficial data and are difficult to be discovered intuitively. These features need to capture the abstract, underlying patterns that are shared between different domains which can be complex and not easily understood intuitively (e.g., the essential characteristics of electricity stealing). Domain-specific features, such as the scale and amplitude of electricity usage data, typically represent more concrete and identifiable information compared to domain-invariant features. These features can be specific to a particular domain and may not be applicable to other domains. Deep neural network may overfit on irrelevant domain-specific noises while learning these “low-level” features. Inspired by this, we propose to use a shallow neural network to capture the “low-level” domain-specific representations and use a deep neural network to capture the “high-level” domain-invariant representations respectively.

Intuitively, the network that is responsible for extracting domain-invariant features requires domain-specific information to understand which characteristics are truly common, and vice versa. We therefore make the two networks communicate with each other to share their information by adding interactions among their different layers. Fig. 3 shows the architecture of the interactive networks. We use \( h^c_k \) to denote the output of the \( k \)-th layer of the domain-specific network and use \( h^v_j \) to indicate the output of the \( j \)-th layer of the domain-invariant network. The operation of interaction between two networks can be expressed as

\[
F_{\text{inter}}(h^c_k, h^v_j) = F_{k,j}(h^c_k) \circ h^v_j,
\]

where \( F_{\text{inter}}(h^c_k, h^v_j) \) corresponds to passing the information of \( h^c_k \) to \( h^v_j \), \( F_{k,j} \) is the transform function which can be implemented by a sub-network or other forms, and \( \circ \) is defined as concatenation.

Note that this interaction results in a blend of domain-invariant and domain-specific representations due to the gradient flow between the two networks during backpropagation in model learning. We add a residual connection [He et al., 2016] between the interactive layer and the output layer, allowing the network’s loss to directly influence the interactive layer and lower layers, thus preserving the characteristics of features before interaction.

### 3.3 Domain-invariant Representations Learning

**Adversarial Learning.** Idea of adversarial learning is used in prior UDA methods [Wilson et al., 2020; Purushotham et al., 2017]. We use \( \mathcal{H}^s(x) \) to denote the domain-invariant feature output by the network from the current time point. Then the output of the domain discriminator is \( D(\mathcal{H}^s(x)) \). To distinguish the source data from the target data, we should minimize the following objective in discriminator:

\[
\mathcal{L}_{\text{dis}} = -\mathbb{E}_{x^s \sim p_s}\log(D(\mathcal{H}^s(x^s))) + \mathbb{E}_{x^t \sim p_t}\log(1 - D(\mathcal{H}^t(x^t))).
\]

On the contrary, the interactive networks are trying to make the domain-invariant features indistinguishable. It has the opposite objective from the discriminator, i.e.,

\[
\mathcal{L}_{\text{net}} = -\mathbb{E}_{x^s \sim p_s}\log(1 - D(\mathcal{H}^s(x^s))) + \mathbb{E}_{x^t \sim p_t}\log(D(\mathcal{H}^t(x^t))).
\]

Besides, for a source data instance \( x^s \), its domain-invariant feature \( \mathcal{H}^s(x^s) \) is also put into the task classifier to predict the task label \( T(\mathcal{H}^s(x^s)) \). Given the ground truth label \( y^s \), we optimize the classifier by:

\[
\mathcal{L}_{\text{task}} = \mathbb{E}_{(x^s,y^s) \sim p_s} \ell(T(\mathcal{H}^s(x^s)), y^s),
\]

where \( \ell \) is the loss function which is usually instantiated as the cross-entropy loss.

By iteratively optimizing the domain discriminator and the interactive networks, we can obtain inseparable domain-invariant representations. However, the classification margin of the task classifier may be decreased by the discriminator. The task classifier forces the data instances with different classes in the source domain to be separated from each other, expecting to make the correct decision boundary. On the contrary, the discriminator favors tightening the distances between all the instances regardless of their classes so that their domains cannot be distinguished. Extremely, when all the instances are collapsed to the same embedding, the discriminator thoroughly loses its discrimination ability. [Koltchinskii and Panchenko, 2002]

**Class-wise Hyperspheres.** To address the above issue of the existing adversarial learning, we propose to inject a class-wise hyperspheres constraint into the adversarial learning. The general idea is to restrict source instances of the same class being projected into the same hypersphere in the domain-invariant feature space, so that the adversarial learning cannot narrow the gaps between different classes. The class-wise hyperspheres constraint is applied to the labeled source data. As for the target data, once the source data are restricted in several hyperspheres (Fig. 2(b)), each target data will be placed into one of the hyperspheres in order to confuse the discriminator (Fig. 2(c)). Eventually, all the source data and target data are projected into the hyperspheres with a large margin (Fig. 2(d)). This approach helps
beled data, our goal is to minimize the radius of the hyper-
sphere. Specifically, we calculate the distance of each source
data point of the domain classifier is
penalize features with distance larger than $R_j$.

$$L_{domain} = \sum_{j=1}^{k} \left[ R_j^2 + \sigma \mathbb{E}_{x \in \mathcal{S}_j} \max(0, ||H^s(x) - c_j||^2 - R_j^2) \right].$$

The first term $R_j$ minimizes the radius directly, while the second term is a soft penalty to restrict source data inside the sphere. Specifically, we calculate the distance of each source domain-invariant feature $H^d(x)$ to its class center $c_{y^*, n}$, then penalize features with distance larger than $R_j$ by measuring
$$\max(0, ||H^d(x) - c_{y^*, n}||^2 - R_j^2),$$
and $\sigma$ is a hyper-parameter controlling the trade-off between the penalty term and radius.

### 3.4 Domain-specific Representations Learning

In our framework, the domain-specific features are learned through the objective of predicting the data from either the source or the target. For simplicity, we use $H^d(x)$ to denote the domain-specific feature output by the network. The output of the domain classifier is $C(H^d(x))$. Then the objective

\begin{table}
\begin{tabular}{ccccccccc}
\hline
\textbf{S} & \textbf{T} & \textbf{CNN} & \textbf{RNN} & \textbf{DDNN} & \textbf{MultiRocket} & \textbf{RDANN} & \textbf{VRADA} & \textbf{SASA} & \textbf{CoDATS} & \textbf{DAF} & \textbf{CLUDA} & \textbf{CADT} \\
\hline
\textbf{WISDM} & & & & & & & & & & & & \\
9 & 1 & 49.99±3.4 & 65.97±2.0 & 56.75±12 & \textbf{75.94±2.9} & \textbf{55.50±7.4} & 40.42±7.3 & 41.00±2.9 & 75.31±2.8 & 33.75±3.3 & \textbf{50.07±4.1} & 76.25±2.2 \\
11 & 3 & 45.82±11 & 41.67±0.4 & 59.60±13 & 41.85±0.4 & 59.60±13 & 41.85±0.4 & 59.60±13 & 41.85±0.4 & 59.60±13 & 41.85±0.4 & 59.60±13 \\
7 & 3 & 65.63±1.1 & 70.94±2.0 & 64.92±0.9 & 60.93±0.3 & 65.63±1.1 & 60.93±0.3 & 65.63±1.1 & 60.93±0.3 & 65.63±1.1 & 60.93±0.3 & 65.63±1.1 \\
25 & 3 & 49.72±12 & 59.49±7.1 & 64.60±1.9 & 59.73±5.3 & 49.72±12 & 59.49±7.1 & 49.72±12 & 59.49±7.1 & 49.72±12 & 59.49±7.1 & 49.72±12 \\
2 & 3 & 55.68±12 & 62.08±3.1 & 61.45±5.4 & 53.98±5.6 & 49.72±12 & 59.49±7.1 & 49.72±12 & 59.49±7.1 & 49.72±12 & 59.49±7.1 & 49.72±12 \\
22 & 3 & 57.55±5.5 & 46.56±3.2 & 64.53±9.2 & 48.18±1.0 & 43.48±2.9 & 49.11±10 & 38.64±3.7 & 51.92±10 & 43.48±2.9 & 49.11±10 & 38.64±3.7 \\
23 & 3 & 61.05±3.8 & 42.04±5.5 & 59.19±1.3 & 52.04±5.7 & 49.11±10 & 38.64±3.7 & 51.92±10 & 43.48±2.9 & 49.11±10 & 38.64±3.7 & 51.92±10 \\
27 & 3 & 62.36±9.3 & 63.36±3.6 & 70.23±0.8 & 52.04±5.7 & 49.11±10 & 38.64±3.7 & 51.92±10 & 43.48±2.9 & 49.11±10 & 38.64±3.7 & 51.92±10 \\
17 & 2 & 64.28±0.2 & 62.44±0.7 & 69.18±0.5 & 61.92±0.1 & 62.44±0.7 & 69.18±0.5 & 61.92±0.1 & 62.44±0.7 & 69.18±0.5 & 61.92±0.1 & 62.44±0.7 \\
16 & 1 & \textbf{75.22±0.1} & 70.94±0.6 & 71.72±1.0 & 72.03±1.0 & 71.72±1.0 & 72.03±1.0 & 71.72±1.0 & 72.03±1.0 & 71.72±1.0 & 72.03±1.0 & 71.72±1.0 \\
1 & 1 & 53.85±0.0 & 55.94±7.1 & 56.09±2.0 & 52.44±1.5 & 53.85±0.0 & 55.94±7.1 & 56.09±2.0 & 52.44±1.5 & 53.85±0.0 & 55.94±7.1 & 56.09±2.0 \\
6 & 1 & 61.45±0.1 & 62.60±4.0 & 60.32±6.3 & 58.90±2.2 & 58.58±5.1 & 49.18±2.3 & 43.48±2.9 & 49.11±10 & 38.64±3.7 & 51.92±10 & 43.48±2.9 \\
18 & 1 & 60.66±0.3 & 61.41±0.8 & 58.24±5.7 & 56.15±0.1 & 58.24±5.7 & 56.15±0.1 & 58.24±5.7 & 56.15±0.1 & 58.24±5.7 & 56.15±0.1 & 58.24±5.7 \\
12 & 1 & 49.99±1.4 & 53.24±6.0 & 56.97±6.3 & 53.24±6.0 & 56.97±6.3 & 53.24±6.0 & 56.97±6.3 & 53.24±6.0 & 56.97±6.3 & 53.24±6.0 & 56.97±6.3 \\
15 & 1 & \textbf{59.38±0.1} & 67.35±0.2 & 67.12±1.0 & 66.94±0.1 & 66.94±0.1 & 66.94±0.1 & 66.94±0.1 & 66.94±0.1 & 66.94±0.1 & 66.94±0.1 & 66.94±0.1 \\
14 & 1 & 54.84±0.5 & 53.59±0.7 & 53.65±0.8 & 49.71±0.6 & 53.59±0.7 & 49.71±0.6 & 53.59±0.7 & 49.71±0.6 & 53.59±0.7 & 49.71±0.6 & 53.59±0.7 \\
\hline
\end{tabular}
\caption{Accuracy of different models on WISDM, UCIHAR, HHAR and SleepEDF. The best result is in bold. S and T represent source and target.}
\end{table}
of the domain classifier is to minimize:
\[
\mathcal{L}_{domain} = -(\mathbb{E}_{x^t \sim \rho} \log(\mathcal{C}(\mathcal{H}^t(x^t)))) \\
+ \mathbb{E}_{x^s \sim \hat{\rho}} \log(1 - \mathcal{C}(\mathcal{H}^s(x^s))). \tag{6}
\]

Optimizing this objective alone cannot ensure that the model captures the domain-specific features fully, since there are usually several domain characteristics, not just one. But once the model learns a particular characteristic that is totally different between source and target, the model stops learning other domain knowledge because it has already made a distinction between the two domains.

Intuitively, this issue can be solved by encouraging the model to learn features from different observation data each time. To be specific, we design the following domain-preserving data augmentation methods on domain classifier:

- **Time-series window cropping**: We randomly crop several slices from the original temporal data and paste them all as a new instance. This method tends to capture local domain characteristics existing in some segments like oscillation.

  - **Shuffling**: We split time series data into some segments and then perform re-ordering to generate a synthetic time series. This method tends to capture local domain knowledge.

- **Adding noise**: We add noises like Gaussian noise and white noise to avoid our network capturing local noise as the domain-specific pattern considering that time series data is long-lasting and may contain noisy patterns like impulses.

- **Segment reversing**: We randomly perform reversing on some segments along the time-axis of data. This method aims to eliminate the temporal information and preserve the frequency information which is invariant to reversing, such as amplification and phase spectrum.

**Overall Objective.** We review all parts of our model and give an overall learning objective. Our model is optimized by adversarial learning. In the first stage, we optimize the model components except for the domain discriminator, the objective is to minimize:
\[
\mathcal{L}_{all} = \mathcal{L}_{task} + \alpha \mathcal{L}_{domain} + \beta \mathcal{L}_{net} + \gamma \mathcal{L}_{sphere}. \tag{7}
\]

In the second stage, we optimize the discriminator and fix other components, the objective is to minimize \(\mathcal{L}_{dis}\). The two stages are optimized iteratively. The whole learning algorithm of CADT is presented in the appendix.

## 4 Experiments

In this section, we conduct sufficient experiments to validate the effectiveness of the proposed model.

### 4.1 Experimental Setup

**Datasets** In our experiments, we utilize five datasets: CHARGE and ELEC from industrial practice, and UCI-HAR [Reyes-Ortiz et al., 2016], WISDM [Kwapisz et al., 2011], SleepEDF [Ragab and Eldele, 2022] from publicly available datasets. Details of datasets will be provided in the supplementary materials.1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CNN w/o CH</th>
<th>CNN w/o CIN</th>
<th>CADT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHARGE</td>
<td>49.04</td>
<td>49.61</td>
<td>56.50</td>
</tr>
<tr>
<td>UCI-HAR</td>
<td>48.83</td>
<td>76.81</td>
<td>85.04</td>
</tr>
<tr>
<td>WISDM</td>
<td>71.10</td>
<td>77.00</td>
<td>76.99</td>
</tr>
</tbody>
</table>

Table 2: The performance of using CADT with different domain adaptation (CNN), coupled interactive networks (w/o CIN), class-wise hyperspheres (w/o CH)

**Baselines** We compare our proposed method with five types of baselines. The first category comprises deep learning networks with no adaptation, including CNN, RNN, DDNN [Qian et al., 2019], and MultiRocket [Tan et al., 2022]. MultiRocket is a fast time series classification (TSC) algorithm known for achieving state-of-the-art accuracy. We design these baselines to serve as a lower bound, revealing the gap of data distribution between source domain and target domain. The second type is RNN-based domain adaptation methods including R-DANN [Tonutti et al., 2019], VRADA [Purushotham et al., 2017] SASA [Cai et al., 2021]. Other types are CNN-based (CoDATS [Wilson et al., 2020]), attention-based (DAF [Jin et al., 2022]) domain adaptation methods and TCN-based (CLUDA [Ozyurt et al., 2023]). The implementation detailed will be presented in the supplementary materials.

### 4.2 Performance Comparison

We compare the performance of CADT with other baselines on public and industrial datasets. Tab. 1 shows that on the CHARGE dataset, our method achieves the best accuracy, especially in challenging scenarios.

The ELEC dataset includes 5 cities with varying economic conditions and regulatory intensities related to electricity theft. Our model outperforms all baseline methods, achieving the highest acc score. This demonstrates its adaptability to domains with different economic situations and geographical locations.

We use two common human activity recognition. From Tab. 1, we find that CoDATS outperforms RNN-based methods, and our CADT outperforms all other baselines, showcasing the superiority of our disentangled CADT model. Additionally, we observe that CNN-based networks without adaptation can sometimes perform well, especially when participants have very similar health conditions.

The Sleep-EDF dataset is particularly unique; we observed that it is highly susceptible to negative transfer, resulting in suboptimal performance for transfer models other than CADT and CoDATS. However, overall, our model still achieves optimal performance.

### 4.3 Ablation Study

In this section, we validate the effectiveness of our method by demonstrating the performance influence of several components contributing to our method. Additionally, we provide an intuitive understanding of our learned hidden representations by projecting them into two-dimensional embeddings.

**Effectiveness of coupled interactive networks.** We investigate the effectiveness of our coupled interactive networks.

---

1 The code and supplementary materials are available at https://github.com/IJCAI-CADT/cadt
(CIN) for learning domain-specific and domain-invariant features. Initially, we compare the performance by removing the domain-specific component (“w/o CIN”). Results in Tab. 2 show reduced precisions in both datasets. We further give an intuition in Fig. 4 by projecting learned features linearly into a 2-dimensional space. By comparing Fig. 4(a) and Fig. 4(b), we find that CIN reduces the domain-invariant gap between the source and target domain. However, leveraging CIN helps to pull in the distance between them. We also drop the domain-invariant component of our model and compare the learned domain-specific features in Fig. 4(c) and Fig. 4(d). Additionally, the learned domain-specific features contain less class information in Fig. 4(d) since the distance among different categories is closer.

**Effectiveness of class-wise hypersphere.** We explore the effectiveness of class-wise hyperspheres (CH). “w/o CH” refers to without the class-wise hyperspheres constraint (CH). Results in Tab. 2 show that CH significantly improves our ability to predict target labels. In Fig. 5(a), without CH, domain-invariant features are dense and hard to distinguish. Fig. 5(b) shows that features of each class are restricted into a very compact cluster and features from different classes are far away in the latent space. Furthermore, we also display the process of convergence with and without CH. Fig. 6(a) shows that leveraging class-wise hyperspheres makes the curve more stable and achieves a better performance. In contrast, the performances without hyperspheres are very unstable, because the classification margin is small and the classifier learned in the source data is easy to make wrong predictions on the target data.

**Effectiveness of temporal data augmentation.** Temporal data augmentation methods are designed for better learning of domain information. We hope that the augmented data can help the model learn plentiful domain-specific features. We compare the performance and convergence of using data augmentation or not. Fig. 6(b) displays that temporal data augmentation helps the model learn better domain-specific features for domain classification.

**5 Conclusion**

In this paper, we propose CADT, a time-series domain adaptation framework that disentangles domain-invariant representations from domain-specific ones using coupled interactive networks. We employ adversarial learning with a novel class-wise hypersphere constraint to learn domain-invariant representations, and introduce domain-preserving data augmentation methods. Our extensive experiments on two public datasets and three real-world applications demonstrate the effectiveness of CADT compared to several state-of-the-art baselines.

**Acknowledgements**

This work is supported by National Natural Science Foundation of China (No. 62176233, No. 62441605).

**Contribution Statement**

Youmin Chen and Xinyu Yan contributed equally to this work.

**References**


[Qian et al., 2021] Wenwen Qian, Jiacheng Li, Changwen Zheng, Bing Su, and Hui Xiong. Robust local preserving and global aligning network for adversarial domain adaptation. IEEE Transactions on Knowledge and Data Engineering, pages 1–1, 2021.


[Yan et al., 2021] Yuguang Yan, Hanrui Wu, Yuzhong Ye, Chaoyang Bi, Min Lu, Dapeng Liu, Qingyao Wu, and Michael Kwok-Po Ng. Transferable feature selection for unsupervised domain adaptation. IEEE Transactions on Knowledge and Data Engineering, pages 1–1, 2021.

[Yang et al., 2021] Shuai Yang, Kui Yu, Fuyuan Cao, Lin Liu, Hao Wang, and Jiuyong Li. Learning causal representations for robust domain adaptation. IEEE Transactions on Knowledge and Data Engineering, pages 1–1, 2021.


[Zhang et al., 2021] Qian Zhang, Wenhui Liao, Guangquan Zhang, Bo Yuan, and Jie Lu. A deep dual adversarial network for cross-domain recommendation. IEEE Transactions on Knowledge and Data Engineering, pages 1–1, 2021.