Enhancing Multivariate Time-Series Domain Adaptation via Contrastive Frequency Graph Discovery and Language-Guided Adversary Alignment

Haoren Guo[1], Haiyue Zhu[2*], Jiahui Wang[1], Prahlad Vadakkepat[1], Weng Khuen Ho[1], Tong Heng Lee[1]

[1]National University of Singapore
[2]Agency for Science, Technology and Research (A*STAR)
{haorenguo_06, wjiahui}@u.nus.edu, {prahlad, wk.ho, eleleeth}@nus.edu.sg, zhu_haiyue@simtech.a-star.edu.sg

Abstract

Unsupervised domain adaptation (UDA) is a machine learning approach designed to minimize reliance on labeled data by aligning features between a labeled source domain and an unlabeled target domain, thereby reducing feature discrepancies, which is efficient for multivariate time series (MTS) prediction. However, most MTS UDA methods focus solely on aligning intra-series temporal features, overlooking the valuable information in inter-series dependencies. Research has highlighted that analyzing decomposed frequency dependencies in time series can reveal significant trends, noise patterns, and intricate temporal details. To address these unexplored frequency dependencies, we introduce the **F**requency **G**raph **D**iscovery Module (**FGD**), which uncovers and aligns shared frequency information and correlations across domains. Additionally, we propose a **F**requency-**C**ontextual **C**ontrastive **L**earning (**FCCL**) framework to better capture and align frequency-contextual representations in multivariate time series, ensuring the extraction of label-invariant information for prediction. Furthermore, considering existing models overlooking the valuable and abundant information outside source and target dataset, we enhance the MTS UDA prediction model with a **L**anguage-guided **A**dversary **A**lignment (**LAA**) module, which leverages the advancement and capabilities of Large Language Models (LLMs) to get text-encoded labeled embeddings and align the classification features, thereby improving prediction accuracy. Our model achieves state-of-the-art results on three public multivariate time-series datasets for unsupervised domain adaptation, as demonstrated by empirical evidence.

1 Introduction

Multivariate Time Series (MTS) data are extensively applied and researched across various fields. The advancement of data-driven models, particularly deep learning methods, has significantly improved performance in MTS-related tasks due to their ability to model latent dependencies within data (Ragab et al. 2023). However, these methods often require a large amount of labeled data for training, which can be costly and sometimes even impossible, such as forcing every single patient to record and submit their daily activity sensor data and label with their specific activities and training individual models for every particular patient. Targeting this bottleneck, Unsupervised Domain Adaptation (UDA) methods have emerged. These methods transfer knowledge from a labeled source domain to an unlabeled target domain without using labels from the target domain (Wang et al. 2023).

Most of the current UDA methods attempt to reduce domain discrepancy by learning domain-invariant features, typically through metric-based, such as Recurrent Neural Networks (RNNs) (Purushotham et al. 2022a) and Long Short-Term Memory (LSTM) (da Costa et al. 2020) or adversarial-based approaches, such as domain adversarial neural network (DANN) (Ganin et al. 2016) and CALDA (Wilson, Doppa, and Cook 2023). These methods have proven effective in reducing label dependency, particularly in tasks involving MTS data. Additionally, transfer learning has emerged as another prominent approach in time series analysis, integrating Contrastive Learning (CL) to capture contextual representations for downstream tasks (Eldele et al. 2021; Tonekaboni, Eytan, and Goldenberg 2021). In the UDA task, CLUDA (Ozyurt, Feuerriegel, and Zhang 2022) has shown promise by leveraging CL to align contextual representations across source and target domains.

Despite advancements, existing MTS UDA methods primarily align features within the intra-series temporal signal space, neglecting inter-series dependencies and multi-frequency information. Domain shifts in time series can manifest as changes in temporal and frequency characteristics, where frequency features are more robust to small shifts and noise, offering better trend and turbulence extraction (He et al. 2023; Guo et al. 2024). Motivated by this, we hypothesize that frequency features and inter-frequency correlations between source and target domains should exhibit similarity. To leverage this, we propose the Frequency Graph Discovery Module (FGD), which identifies inherent frequency relationships and aligns features at the frequency graph level. Complementing this, we introduce the Frequency-Contextual Contrastive Learning (FCCL) framework to align frequency-contextual representations from augmented time series, extracting label-invariant information. Together with adversarial training to reduce domain discrepancies, our proposed model, ConFGD, integrates FGD and FCCL to enhance prediction accuracy.

Existing models extract features only from source and target data, overlooking abundant information outside. In line with the saying, "Those who wanted to learn would

*Corresponding author

seek out a teacher, one who could propagate the doctrine, impart professional knowledge, and resolve doubt", we see large language models (LLMs) as this teacher, offering vast knowledge from books, articles, and other sources. The advancement and capabilities of LLMs inspired us to explore extending their potential to guide our MTS UDA task. Since dataset labels can serve as inherent prompts for generating LLM embeddings, we designed a time-efficient and computationally lightweight adversary module, Language-guided Adversary Alignment (LAA) module, that can be added not only to our ConFGD model but also to existing UDA models to improve the prediction ability and the upgraded model is denoted as ConFGD+.

We evaluate our method on the benchmark real-world MTS dataset HAR (Anguita et al. 2013), WISDM (Kwapisz, Weiss, and Moore 2011), and HHAR (Stisen et al. 2015), and achieve state-of-the-art (SOTA) performances. We further conduct experiments by introducing the LAA module to ConFGD and other existing models, such as CLUDA, to confirm its effectiveness and superior performance.

Contributions:

1. We propose and develop a novel Frequency Graph Discovery Module (FGD) to discover and align the inherent inter-series frequency channels information and relationship for Unsupervised Domain Adaptation (UDA) of time series.
2. To incorporate with the frequency embeddings, we capture the frequency-contextual representation by the novel design of a brand new Frequency-Contextual Contrastive Learning (FCCL) framework and further enhance the prediction capabilities by extracting the label-invariant information.
3. We are the first to introduce the LLM as a time-efficient and computationally lightweight adversarial language-guided model, which can be incorporated into not only our ConFGD model but also existing UDA models to enhance prediction performance.

2 Methodology

2.1 Problem Formulation

In this paper, the objective is to perform UDA on the multivariate time series classification tasks. There are two datasets sampled from two different distributions respectively which are given as $\mathcal{D}_{src} = \{(\mathbf{X}^i_{src}, y^i_{src})\}_{i=1}^{N_s}$ and $\mathcal{D}_{trg} = \{\mathbf{X}^i_{trg}\}_{i=1}^{N_t}$. $\mathcal{D}_{src}$ represents the **labeled** source domain dataset with N_s number of samples. $\mathbf{X}^i_{src} = \{\mathbf{x}^{it}_{src}\}_{t=1}^{T} \in \mathbb{R}^{D\times T}$ is a sample of the source domain with T time steps and D sensor observations and y^i_{src} is the label for the given sample. $\mathcal{D}_{trg}$ represents the **unlabeled** target domain dataset with N_t number of samples. Similar as the samples from $\mathcal{D}_{src}$, the sample from the target domain is $\mathbf{X}^i_{trg} = \{\mathbf{x}^{it}_{trg}\}_{t=1}^{T} \in \mathbb{R}^{D\times T}$. In addition, the labels for the target domain are applicable during testing, therefore we specifically define the labeled testing target domain dataset as $\mathcal{D}^{test}_{trg} = \{(\mathbf{X}^i_{test\_trg}, y^i_{test\_trg})\}_{i=1}^{N^{test}_t}$ where N^{test}_t is the number of samples in the test target domain dataset and $\mathbf{X}^i_{test\_trg} = \{\mathbf{x}^{it}_{test\_trg}\}_{t=1}^{T} \in \mathbb{R}^{D\times T}$. The same for both source and target domain samples, $\mathbf{x}^{it} = \{x^{itd}\}_{d=1}^{D} \in \mathbb{R}^D$. Although the source and target domains are drawn from different distributions, each representing distinct marginal distributions, the conditional distributions for both domains are identical. We assume the two domains share the same label space. The main goal of this work is to minimize the distribution shift between the source and target domains and achieve good generalization on the target domain $\mathcal{D}_{trg}$ by exploiting the labeled source domain samples.

2.2 Architecture Overview

The framework of our proposed **ConFGD** for multivariate time series unsupervised domain adaptation is shown in Fig. 1. First of all, both augmented time series from both domains are decomposed into multi-frequency level signals by implementing **Discrete Wavelet Transform (DWT)**, and the corresponding features, $\mathbf{H}_{src}$ and $\mathbf{H}_{trg}$, are extracted by the **temporal projection** network respectively. After that, the **frequency graph discovery module (FGD)** comprising the encoder, aggregate module, and decoder, is trained to capture and align the graph, including both edge and node attributes, across various frequency levels. The classification feature $\mathbf{v}^S_{node}$ extracted from $\mathbf{H}_{src}$ by the **graph discovery encoder** is utilized to predict the label y_{src} of time series $\mathbf{X}_{src}$. The **domain discrimination** is trained to distinguish domains by utilizing in the encoder node embeddings $\mathbf{v}^S_{node}$ and $\mathbf{v}^T_{node}$. We arbitrarily labeled the source domain as 0 and the target domain as 1 for the training. To enhance the capture of contextual representations, **frequency-contextual contrastive learning (FCCL)** is employed across each domain through the utilization of a **momentum-updated** temporal projection and encoder. The ConFGD+ introduces an **Language-Guided Alignment (LAA)** module which is shown in Fig. 2 to align the embeddings gotten from label prompts with the classification features $\mathbf{v}^S_{node}$. This is explained in Sec. 2.5.

2.3 Frequency Graph Discovery Module

The information from time series signals is equally important in both the time and frequency domain. Signals in the time domain are vulnerable to noise and disruptions, making it challenging to discern trends and detailed information due to their volatility. In contrast, frequency-domain methods transform these signals to emphasize their spectral features, making it easier and more feasible to extract and recognize turbulence and trends. Higher frequency components usually include finer details and generally suggest random variations and noise. The lower frequency components usually offer insights into trend dynamics and more stable dependencies. Therefore, we hypothesize that if the source and target domains share the same label space, the information shared by the frequency channels and the correlation among the multi-frequency level signals should be similar.

DWT Frequency Decomposed Fourier Transform and Wavelet Transform are two prevailing methods for transforming signals between the time and frequency domain. Compared with Wavelet Transform, Fourier Transform mostly focuses on the overall dependencies of the time domain such

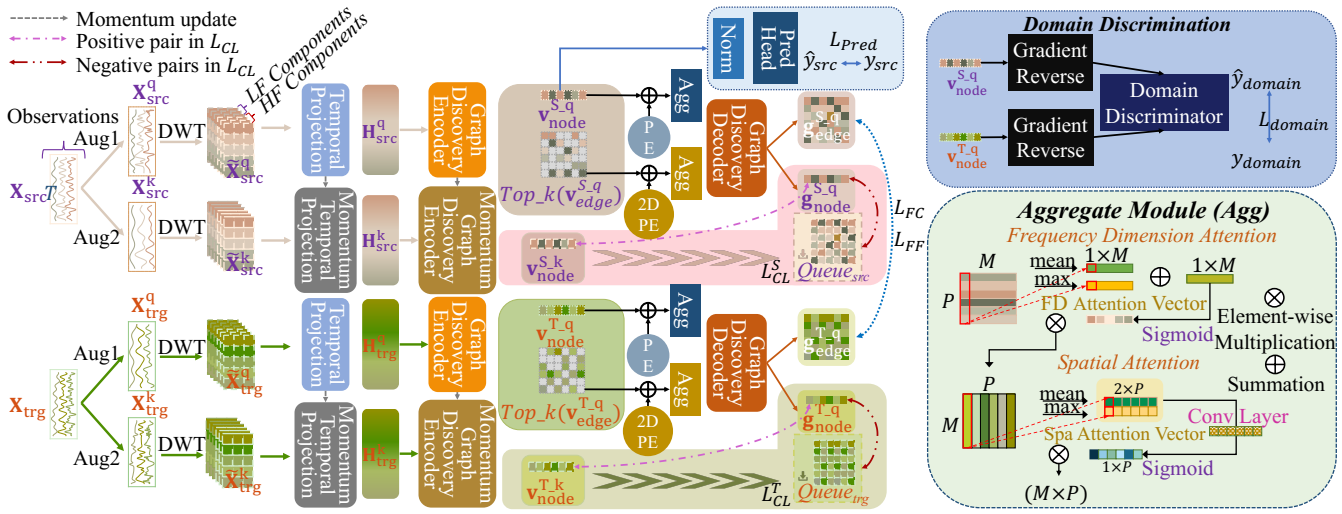


Figure 1: **ConFGD Model Architecture (Best view in color)**. The left-side graph shows the overall structure. Source and target samples are augmented into query $\mathbf{X}^q$ and key $\mathbf{X}^k$, then decomposed by DWT. They are passed through the temporal projection layer and graph discovery encoder to generate node and edge embeddings, where $\mathbf{v}_{node}^{S\_q}$ is used for prediction (L_{pred}). The aggregate modules (Agg) (detailed in the lower right) integrate query node and edge embeddings with positional encodings. These combined embeddings are decoded into the frequency correlation graph for L_{FC}, L_{FF}, and L_{CL}. The domain discrimination framework (L_{domain}) is shown in the upper right.

as global seasonal and temporal information (Karlton 2020). Even though the windowed Fourier Transform technique is optimized to capture the local feature, it is restricted by the constant window to manage the input shorter or longer than the window (Gabor 1946). Consequently, the Wavelet Transform is selected for frequency decomposition due to its scaled window, which addresses the limitation of inconsistent input lengths and enables the capture of local properties. More specifically, we employ the Discrete Wavelet Transform (DWT) in feature-wise. The decompose implementation is denoted as $DWT(\cdot)$ which can be illustrated as $\tilde{\mathbf{X}}_{coeff} = DWT(\mathbf{X})$. $\tilde{\mathbf{X}}_{coeff}$ is the concatenate coefficients of all the levels gotten by DWT. There is a number of S sets of coefficients within $\tilde{\mathbf{X}}_{coeff}$. To simply extract the low and high-frequency part of the time series, we multiply each set of coefficients $\tilde{\mathbf{X}}_{coeff}[s]$ with a parameter λ_s. The low-frequency parameter λ_s^{lf} exhibits a small value at high levels, *i.e.*, when s is small, conversely for the high-frequency parameter λ_s^{hf}. This can be denoted as

$$\tilde{\mathbf{X}}_{coeff}^{lf}[s], \tilde{\mathbf{X}}_{coeff}^{hf}[s] = \lambda_s^{lf}\tilde{\mathbf{X}}_{coeff}[s], \lambda_s^{hf}\tilde{\mathbf{X}}_{coeff}[s]. \quad (1)$$

Then, the filtered high- and low-frequency coefficient sets would be inverse back to the time domain by implementing the Inverse Discrete Wavelet Transform (IDWT),

$$\begin{aligned}\overline{\mathbf{X}} &= Concat(\{IDWT(\tilde{\mathbf{X}}_{coeff}^{hf}[s]\}_{s=1}^{S}),\\ \underline{\mathbf{X}} &= Concat(IDWT(\{\tilde{\mathbf{X}}_{coeff}^{lf}[s]\})_{s=1}^{S},\\ \tilde{\mathbf{X}} &= Concat(\overline{\mathbf{X}}, \underline{\mathbf{X}}), \{\overline{\mathbf{X}}, \underline{\mathbf{X}}\} \in \mathbb{R}^{D\times T\times S}, \tilde{\mathbf{X}} \in \mathbb{R}^{D\times T\times 2S}\end{aligned} \quad (2)$$

where the $Concat$ is a concatenate implementation along the frequency channel and $\overline{\mathbf{X}}$ and $\underline{\mathbf{X}}$ represent the decomposed high-frequency and low-frequency time series for each feature and $\tilde{\mathbf{X}}$ is the integration of the multi-frequency levels. Motivated by SASA (Cai et al. 2021), we allocate independent temporal projection layers for each frequency channel, $\mathbf{H} = Concat(\{T[s](\tilde{\mathbf{X}}[s])\}_{s=1}^{2S})$, where $T[s]$ is the independent temporal projection layer for each channel, and $\mathbf{H} \in \mathbb{R}^{2S\times P}$ is the hidden representation and P is the hidden dimension of each $T[s]$.

Graph Discovery Module The objective of the graph discovery module is to find the intrinsic relationship among the frequency channels. The interaction among the frequency channels is modeled as a frequency correlation graph $\mathcal{G}$ which contains the edge and node information. The graph includes self-loop which means each node also points to itself. Inspired by the graph discovery architecture in the VCDN (Li et al. 2020), our Graph Discovery Module utilizes graph neural networks (see Appendix C) as the encoder and decoder, further integrating an aggregation module between the encoder and decoder to capture the frequency channel and spatial channel information.

To simplify the notation, we let Q represent $2S$ in the previous chapter. The fully connected graph encoder is denoted as $G^{enc}(\cdot)$, and the process can be described as

$$\mathbf{v}_{node}, \mathbf{v}_{edge} = G^{enc}(\mathbf{H}), \mathbf{v}_{node} \in \mathbb{R}^{Q\times P}, \mathbf{v}_{edge} \in \mathbb{R}^{Q^2\times P} \quad (3)$$

where the $\mathbf{v}_{node}, \mathbf{v}_{edge}$ are denoted as the encoder node and edge embedding respectively.

The encoder node embedding $\mathbf{v}_{node}$ is also the classification feature that is utilized for the prediction. We define the prediction head as $Pred(\cdot)$ and the *prediction loss* is

expressed as

$$\hat{y}^i_{src} = Pred(\mathbf{v}^{Si}_{node}), \quad L_{pred} = \frac{1}{N_s}\sum_{i=1}^{N_s} L_{ce}(\hat{y}^i_{src}, y^i_{src}), \tag{4}$$

where the L_{ce} is the cross-entropy loss.

Besides, the encoder node embedding $\mathbf{v}_{node}$ is also used to align the domain distributions. We introduce adversarial learning (Ganin et al. 2016), to enhance the indistinguishability of the domain discriminator $D_{disc}(\cdot)$. This encourages the model to classify both domains as the same class. To achieve this, the gradients of $\mathbf{v}^S_{node}$ and $\mathbf{v}^T_{node}$ are reversed by the gradient reversal layer $F(\cdot)$. This layer is designed to train $G^{enc}(\cdot)$ to maximize the *domain classification loss*, which is minimized during the training of $D_{disc}(\cdot)$. The gradient reverse process is defined as $F(x) = x, \quad \frac{dF}{dx} = -\mathbf{I}$. The pseudo domain labels are defined as d_{src} and d_{trg}, the *domain classification loss* can be written as

$$\begin{aligned} L_{domain} =& \frac{1}{N_s}\sum_{i=1}^{N_s} L_{ce}(D_{disc}(R(v^{Si}_{node})), d_{src}) \\ &+ \frac{1}{N_t}\sum_{i=1}^{N_t} L_{ce}(D_{disc}(R(v^{Ti}_{node})), d_{trg}) \end{aligned} \tag{5}$$

Next, we implement $\mathbf{v}_{edge} = Top\_k(\mathbf{v}_{edge})$ to take the top k important edge embeddings of each node to reduce the computational burden and the disturbance of the relatively irrelevant edge information and set the discarded feature into zeros. Then, the *1D* frequency positional encoding and *2D* frequency positional encoding is added to the node and edge embeddings respectively (See Appendix B).

To aggregate the encoder features ($\mathbf{v}_{edge}$ and $\mathbf{v}_{node}$) with their positional encoding and integrate both frequency and spatial information, we introduce the aggregate module. We utilize max pooling to capture the most prominent features and introduce average pooling to obtain smooth global information. By integrating these two pooling strategies, the module functions like residual layers, effectively capturing features from both frequency and spatial channels while preventing gradient vanishing. Additionally, to manage computational complexity, we include only one learnable layer at the end of this module, ensuring it remains lightweight and does not add significant computational overhead. The overall flow is denoted as below,

$$\begin{aligned} \mathbf{v}' &= \mathbf{v} \otimes \sigma(AP(\mathbf{v}) + MP(\mathbf{v})), \\ \mathbf{w} &= \mathbf{v}'^T \otimes \sigma(BN(Conv(AP(\mathbf{v}'^T) \oplus MP(\mathbf{v}'^T)))), \end{aligned} \tag{6}$$

where $\otimes$ is the feature-wise multiplication, $\oplus$ is the concatenation of two matrices, σ is the sigmoid activation function. The AP and MP are average pooling and max pooling. Due to the dimension for $\mathbf{v}_{edge}$ and $\mathbf{v}_{node}$ being different, in Fig. 1, we use M and P to imply the dimension would remain the same after the aggregation. Then, the aggregate feature are denoted as $\mathbf{w}_{node} \in \mathbb{R}^{Q\times P}$ and $\mathbf{w}_{edge} \in \mathbb{R}^{Q^2\times P}$. After getting the aggregate features, a parameterized decoder $G^{dec}(\cdot)$ is applied to take in both the node and edge aggregated information to build up the frequency correlation graph

$$\mathcal{G} \sim \{\mathbf{g}_{node}, \mathbf{g}_{edge}\} = G^{dec}(\mathbf{w}_{node}, \mathbf{w}_{edge}), \tag{7}$$

where $\mathbf{g}_{node} \in \mathbb{R}^{Q\times P}$ is the node correlation projection and $\mathbf{g}_{edge} \in \mathbb{R}^{Q^2\times 1}$ is the edge correlation projection. The $\mathbf{g}_{node}$ is used for calculating the *frequency feature-wise loss* (L_{FF}) and $\mathbf{g}_{edge}$ is used for calculating the *frequency contrastive loss* (L_{FC}). The L_{FF} is to calculate the expectation of the discrepancy among the frequency graph over feature-wise between domains and the L_{FC} is to align the frequency features correlations where we assume the correlation between the same set of frequency pairs should have similar distribution and properties, and the features from the different frequency pairs should be different. These two losses are denoted as

$$\begin{aligned} L_{FF} &= \mathbb{E}(|\mathbf{g}^S_{node} - \mathbf{g}^T_{node}|), \\ L_{FC} &= -\frac{1}{Q}\sum_{i=1}^{Q} \log \frac{e^{\mathbf{g}^{Si}_{edge}(\mathbf{g}^{Ti}_{edge})^T}}{\sum_{j=1,j\neq i}^{Q} e^{\mathbf{g}^{Si}_{edge}(\mathbf{g}^{Tj}_{edge})}}. \end{aligned} \tag{8}$$

where the i and j are the frequency level vector in the edge correlation projection.

2.4 Frequency-Contextual Contrastive Learning

Contrastive learning (CL) has been widely proven to learn and capture contextual representation effectively in various unsupervised representation learning scenarios (Wu et al. 2024; Eldele et al. 2021) ; see related work (Appendix A). In the CLUDA (Ozyurt, Feuerriegel, and Zhang 2022), CL has been demonstrated to be efficient in capturing and aligning the contextual representation of multivariate time series data which preserves the label invariant information and makes the domain alignment and prediction tasks easier. To enhance the existing FGD module, we propose the FCCL approach. This method is designed to align the frequency-contextual information from two augmented views of the same sample and distinguish it from the frequency-contextual information from other samples.

The overall FCCL framework is shown in Fig. 1. Motivated by MoCo (He et al. 2019), we implement the CL with the momentum contrast technique. In addition, the semantic-preserving augmentation strategy is utilized to get two augmentations for each sample as the key $\mathbf{X}^k$ and query $\mathbf{X}^q$ respectively where $\mathbf{X}^q, \mathbf{X}^k \in \mathbb{R}^{D\times T}$. Both would undergo decomposition into multiple frequency channels using DWT following with as Sec. 2.3, $\{\mathbf{X}^q, \mathbf{X}^k\} \rightarrow \{\tilde{\mathbf{X}}^q, \tilde{\mathbf{X}}^k\}$. The $\tilde{\mathbf{X}}^q$ and $\tilde{\mathbf{X}}^k$ are passed to the temporal projection layer $T(\cdot)$ and $\tilde{T}(\cdot)$ to get the frequency embeddings $\mathbf{H}^q$ and $\mathbf{H}^k$ respectively. After that, according to Eqn. 3, frequency embeddings $\mathbf{H}^q$ and $\mathbf{H}^k$ are processed by the graph discovery encoder $G^{enc}(\cdot)$ and $\tilde{G}^{enc}(\cdot)$ to get their encoder node embeddings $\mathbf{v}^q_{node}$ and $\mathbf{v}^k_{node}$. The weights of $\tilde{T}(\cdot)$ and $\tilde{G}^{enc}(\cdot)$ are momentum updated via

$$\{\theta_{\tilde{T}}, \theta_{\tilde{G}^{enc}}\} \leftarrow \alpha\{\theta_{\tilde{T}}, \theta_{\tilde{G}^{enc}}\} + (1-\alpha)\{\theta_T, \theta_{G^{enc}}\}, \tag{9}$$

$\alpha \in [1, 0)$ is the momentum coefficient to update the weights. To avoid model collapse, where the query and key networks might converge to trivial or identical representations, the query encoder node embeddings are passed through $G^{dec}(\cdot)$ to obtain the node correlation projection $\mathbf{g}^q_{node}$. The objective of FCCL is to bring $\mathbf{g}^{qi}_{node}$ closer to its positive

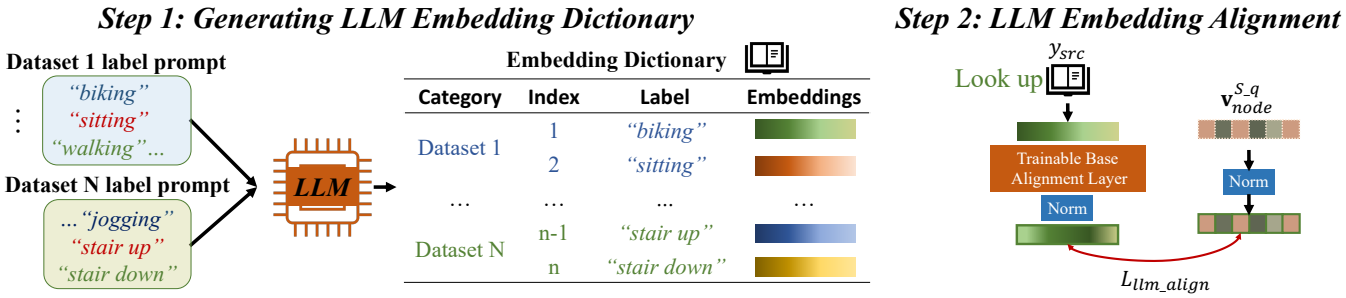


Figure 2: Language-guided Adversary Alignment (LAA).

sample, *i.e.*, $\mathbf{v}_{node}^{ki}$, but further to its negative samples in $Queue \leftarrow \{\mathbf{v}_{node}^{kj}\}_{j=1}^{J}$ where J is the queue size of the large set of negative pairs collected from previous batch($J \gg B$, $J\%B = 0$, B is the batch size), which is efficient to capture better frequency-contextual representations (Ozyurt, Feuerriegel, and Zhang 2022). The contrastive loss L_{CL} is denoted as

$$-\frac{1}{B}\sum_{i=1}^{B}\log\frac{e^{\mathbf{g}_{node}^{qi}(\mathbf{v}_{node}^{ki})^T/\tau}}{e^{\mathbf{g}_{node}^{qi}(\mathbf{v}_{node}^{ki})^T/\tau}+\sum_{j=1}^{J}e^{\mathbf{g}_{node}^{qi}(\mathbf{v}_{node}^{kj})^T/\tau}}. \tag{10}$$

The τ is the temperature scale which is larger than 0. The source and target domains' contrastive losses are denoted as L_{CL}^{S} and L_{CL}^{T} respectively, and the queues for them are denoted as $Queue_{src}$ and $Queue_{trg}$.

2.5 ConFGD+: Language-guided Adversary Alignment

The **ConFGD+** framework extends our proposed **ConFGD** framework with a Language-guided Adversary Alignment (LAA) module, as illustrated in Fig. 2, to improve the prediction accuracy by aligning the classification feature $\mathbf{v}_{node}^{S}$ with the label embeddings guided by the pre-trained LLM text encoder. The LAA can also be added to other MTS UDA models.

Given the large number of parameters in LLM, obtaining embeddings from label prompts can slow down inference. Instead of directly integrating the LLM into the model and extracting features for each sample during training, we store all label embeddings, $\mathbf{Eb} = LLM(LP)$, in a dictionary $\mathbf{Dict}$ before training starts,

$$\mathbf{Dict} \leftarrow \{Dataset_{name}, y_{src}^{i}, LP, \mathbf{Eb}\} \tag{11}$$

where the LP is the label prompts, *i.e.*, the text label of the ground truth y_{src}^{i}, along with the $Dataset_{name}$ stored in the $\mathbf{Dict}$, facilitates locating the specific set of labels while loading data. Consequently, the dictionary is constructed only once, making the process efficient in time and computation.

During the model training, the way to get embeddings is similar to looking up the dictionary where the index is $[Dataset_{name}, y_{src}^{i}]$ for each sample. Then, the source dataset $\mathcal{D}_{src}$ is denoted as $\mathcal{D}_{src}^{+} = \{([\mathbf{X}_{src}^{i}, \mathbf{Eb}^{i}], y_{src}^{i})\}_{i=1}^{N_s}$. As the dimension of $\mathbf{Eb}$ is high, we adopt the approach outlined in (OpenAI 2024) to reduce its dimension to match that of the flattened $\mathbf{v}_{node}^{S_q} \in \mathbb{R}^{1\times QP}$ by $\mathbf{Eb}' = \mathbf{Eb}[:Q \times P] \in \mathbb{R}^{1\times QP}$.

Considering the base of $\mathbf{Eb}'$ and the classification feature $\mathbf{v}_{node}^{S}$ are not consistent, the $\mathbf{Eb}'$ is multiplied by a learnable identity matrix $\mathbf{W}_{\mathbb{I}}$ to slightly align their basement. The *language-guided alignment loss* is expressed as

$$L_{llm\_align} = \frac{1}{N_s}\sum_{i=0}^{N_s}(1-\frac{\mathbf{v}^{Si}\cdot\mathbf{W}_{\mathbb{I}}\mathbf{Eb}'^{i}}{\max\left(\|\mathbf{v}^{Si}\|_2\cdot\|\mathbf{W}_{\mathbb{I}}\mathbf{Eb}'^{i}\|_2,\epsilon\right)}), \tag{12}$$

where the ϵ is a constant set to avoid a zero denominator.

2.6 Overall Loss

(a) **ConFGD**: $\min L_{ConFGD} = L_{pred} + \lambda_{domain}L_{domain} + \lambda_{freq}(L_{FF} + L_{FC}) + \lambda_{CL}(L_{CL}^{S} + L_{CL}^{T})$.

(b) **ConFGD+**: $\min L_{ConFGD+} = L_{ConFGD} + \lambda_{llm\_align}L_{llm\_align}$.

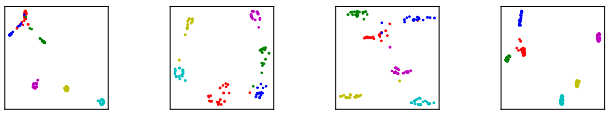
3 Experiment Preparation

To evaluate the effectiveness of our proposed ConFGD and ConFGD+, we conduct experiments on three **benchmark** datasets, HAR (Anguita et al. 2013), WISDM (Kwapisz, Weiss, and Moore 2011) and HHAR (Stisen et al. 2015). Besides, we utilize GPT3 from OpenAI (OpenAI 2024) to create the text embedding dictionary for the LAA module during data preparation. The dataset preparation details are specified in Appendix D. To demonstrate how our proposed ConFGD and ConFGD+ models significantly improve the accuracy on target domains, we randomly select 10 source-target domain pairs for HAR and WISDM, and 7 pairs for HHAR during the evaluation, where the domains are distinct by different participants. More dataset details are specified in Appendix D.

During the experiments, we choose 2 w/o UDA models and 11 UDA models as the **baseline** models. The 2 w/o UDA models are different from those including the graph discovery encoder. The 11 baseline models are VRADA (Purushotham et al. 2022b), CoDATS (Wilson, Doppa, and Cook 2020), AdvSKM (Liu and Xue 2021), CAN (Kang et al. 2019), CDAN (Long et al. 2018), DDC (Tzeng et al. 2014), DeepCORAL (Sun and Saenko 2016), DSAN (Zhu et al. 2020), HoMM (Chen et al. 2020), MMDA (Rahman et al. 2020) and CLUDA (Ozyurt, Feuerriegel, and Zhang 2022) where

Dataset	Metric	TCN(w/o UDA)	w/o UDA	VRADA	CoDATS	AdvSKM	CAN	CDAN	DDC	DeepCORAL	DSAN	HoMM	MMDA	CLUDA	ConFGD	ConFGD+
HAR	Avg. Acc	60.48	80.60	77.03	66.24	61.93	67.98	69.58	61.57	71.21	74.70	70.66	59.18	91.21	94.86	**96.81**
	Std.	18.79	15.07	6.65	19.33	19.58	14.26	15.80	18.81	11.07	11.23	13.99	18.66	6.78	4.63	3.87
	Avg. F1	53.63	77.69	71.05	59.57	55.15	62.97	62.69	54.85	66.86	70.89	64.36	49.15	90.63	94.84	**96.69**
	Std.	19.56	17.45	9.05	19.32	20.25	16.35	19.94	19.56	12.58	12.39	16.19	19.58	7.54	5.12	4.09
WISDM	Avg. Acc	66.26	62.08	65.69	64.41	64.91	58.45	57.90	66.07	63.54	59.10	60.22	53.77	72.32	79.15	**80.67**
	Std.	11.69	8.68	11.40	12.88	10.36	12.15	17.93	10.05	14.03	17.29	14.19	17.93	7.55	4.50	5.03
	Avg. F1	50.20	46.04	47.67	46.71	46.79	45.33	37.66	48.01	44.04	46.48	43.33	35.30	53.92	62.25	**65.42**
	Std.	9.56	10.68	17.39	10.54	9.04	13.08	14.79	8.37	9.59	18.02	13.16	14.09	15.50	16.16	15.25
HHAR	Avg. Acc	68.84	76.51	73.57	59.89	64.96	73.69	63.92	64.81	73.38	62.38	71.44	58.95	75.90	83.08	**84.11**
	Std.	12.72	16.31	19.63	11.12	15.75	16.64	20.27	16.83	13.64	18.58	14.48	12.68	14.61	12.46	12.24
	Avg. F1	66.44	73.92	70.98	57.43	60.28	69.83	57.18	61.41	71.25	60.47	67.93	56.21	74.83	82.09	**83.94**
	Std.	13.92	20.26	17.56	11.43	18.19	21.01	22.52	18.98	17.85	20.42	17.95	12.33	15.68	14.50	12.31

Table 1: Average evaluation results (Accuracy (%) and Macro F1 (%)) of the baseline models over 10 pairs of source-target domains on the HAR and WISDM and 7 pairs of source-target domains on HHAR (Higher is better. The best is in **bold** and the second best is marked with underline) with the standard deviation (Std., lower is more stable) .



(a) w/o UDA (b) CLUDA (c) ConFGD (d) ConFGD+

Figure 3: Embedding t-SNE visualizations of w/o UDA, CLUDA, ConFGD, and ConFGD+ on the benchmark dataset HAR (Anguita et al. 2013). The classes are differentiated by colors. The "star" shapes are from the source domain, and the "round" shapes are from the target domain.

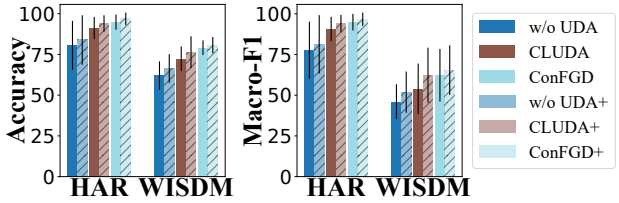


Figure 4: LAA Module Evaluation Results

the specifications are in Appendix E.1. To further assess the effectiveness of our proposed LAA module in ConFDG+, we incorporate this module into the w/o UDA and CLUDA models for experimentation. These models are only trained on the source domain and selected by the performance on the validation source domain dataset. More specific training details, time of execution, and contrastive learning augmentation strategy are specified in Appendix E.2 and E.3.

4 Experiment Results

4.1 Performance Comparison

The average evaluation results for baseline models across 10 source-target pairs on HAR and WISDM, and 7 pairs on HHAR, are summarized in Table 1, with full UDA results available in Appendix F. For HAR, our ConFGD outperforms the best baseline (CLUDA) by 4.00% in accuracy (94.86 vs. 91.21) and 4.44% in Macro F1 (94.84 vs. 90.63). On WISDM, ConFGD surpasses CLUDA by 9.40% in accuracy (79.15 vs. 72.32) and 15.44% in Macro F1 (62.25 vs. 53.92), showing superior performance on the more challenging WISDM task. For HHAR, ConFGD improves accuracy by 9.46% (83.08 vs. 75.90) and Macro F1 by 9.70% (82.09 vs. 74.83). Integrating the LAA module into ConFGD (resulting in ConFGD+) further boosts all results, highlighting the LAA module's effectiveness (detailed in Sec. 4.2). Moreover, our ConFGD and ConFGD+ models demonstrate significantly lower standard deviation (Std.) in accuracy (HAR, WISDM) and Macro F1 (HAR), indicating greater prediction stability. Overall, our models achieve superior and more consistent performance compared to the baselines.

Embedding visualization: Fig. 3 visualizes the embeddings to illustrate the domain discrepancies learned by different models. In Fig. 3a, w/o UDA exhibits a clear domain shift, where the green class splits into two clusters, and some classes overlap. Fig. 3b and Fig. 3c represent CLUDA and ConFGD, the best baseline CL framework models, which show clearer clusterings compared to w/o UDA, confirming the effectiveness of contrastive learning. However, slight domain shifts and occasional class mixing are still present. ConFGD, with its novel FCCL framework aligning contextual representations at the frequency level, further reduces class mixing compared to CLUDA. Fig. 3d, depicting ConFGD+, achieves the most distinct clusterings, with source and target domain embeddings well-aligned and no visible class mixing, thanks to its language-guided alignment. These findings, supported by additional t-SNE visualizations in Appendix I, validate the effectiveness of ConFGD and ConFGD+.

4.2 LAA Module Evaluation

To validate the effectiveness of LAA across different LLMs, we tested it on five paired WISDM datasets using GPT-3 (Brown et al. 2020), BERT (Devlin et al. 2019), and LLaMA2 (Touvron et al. 2023) as text embedding frameworks. The results (Appendix G) confirmed its robustness, with GPT-3 showing slightly better accuracy and LLaMA2 excelling in F1 score. Ultimately, GPT-3 was chosen as the primary model for its advanced natural language understanding, high-quality embeddings, and extensive training on diverse textual data, making it ideal for our domain adaptation tasks.

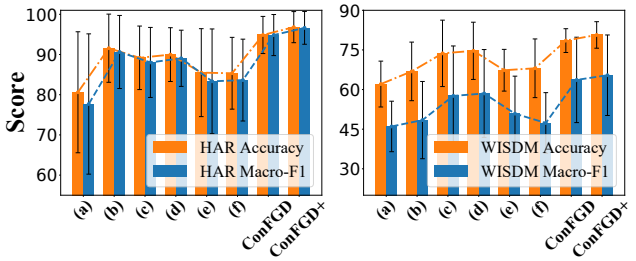


Figure 5: Ablation study: (a) w/o UDA, (b) w/o L_{FC}&L_{FF}, (c) w/o L_{CL}, (d) w/o L_{CL}&L_{FC}&L_{FF}, (e) w/o L_{domain} and (f) w/o PE.

To extensively prove the effectiveness of our introduced LAA module, we add it to not only our proposed ConFGD but also the w/o UDA model and the best baseline model CLUDA where the upgraded model is denoted with a "+" sign. Fig. 4 shows the performance comparisons between the basic models and the upgraded "+" models. The performances of all the upgraded models surpass their basic models. To be noticed, on WISDM, the Macro-F1 score of CLUDA+ increased by 15.50% from CLUDA which is considered as a significant improvement. Furthermore, it also proves the accuracy and effectiveness of the LAA module. We provide all the detailed results of the LAA module evaluations in Appendix H.

4.3 Ablation Study

We conduct an ablation study of our proposed ConFDG framework by comparing different ablation models (discarding some parts of the variants) to gain a deeper insight into the different components of our framework. The ablation models used for the evaluations are (a) w/o UDA, (b) w/o L_{FC}&L_{FF}, (c) w/o L_{CL}, (d) w/o L_{CL}&L_{FC}&L_{FF}, (e) w/o L_{domain} and (f) w/o PE. Fig. 5 presents ablation results on HAR and WISDM averaged over 10 random initializations, showing all components improve upon the base w/o UDA, validating the framework's effectiveness. Detailed results are in Appendix J.

Sensitivity Analysis: The sensitivity analysis results for the HAR and WISDM datasets are shown in Fig. 6. We tested λ_{CL} and λ_{freq} values ranging from 0.05 to 0.2, and the ConFGD model exhibited stability across this range. Notably, the standard deviation of the F1 score on WISDM is only 0.71 for λ_{CL} and 0.69 for λ_{freq}, indicating highly stable performance despite variations in these hyperparameters.

Top_k Study: In $\mathbf{v}_{edge}$, Top_k is a tunable parameter, with smaller Top_k improving efficiency. Fig. 7 shows Top_k results on HAR and WISDM source-target pairs with standard deviation (std.) bars and averages. The std. values are very small, with the std. for HAR Macro-F1 being only 0.17 (0.18% of the average value), indicating stable performance across Top_k. Reducing Top_k enhances efficiency while maintaining accuracy. Appendix E.2 provides execution times, showing $Top\_k = 5$ achieves better results and faster times than CLUDA. Full results are in Appendix K.

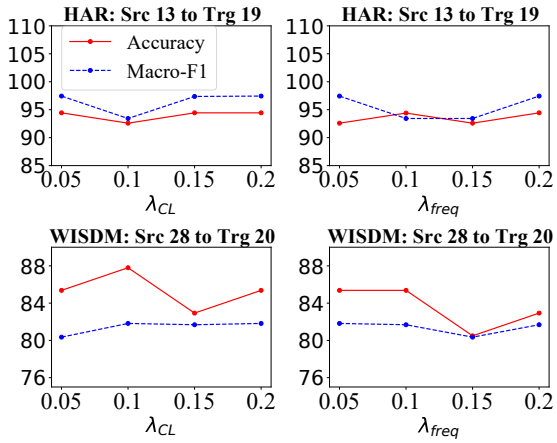


Figure 6: The sensitivity analysis for different for different λ_{CL} and λ_{freq}

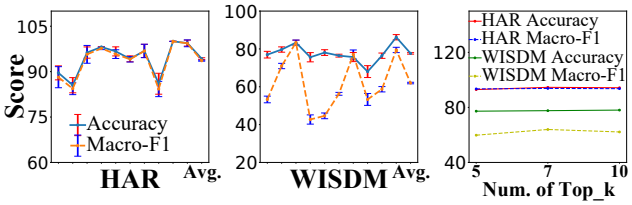


Figure 7: Top_k Study Evaluation Results [x-axis tick labels (Source-Target) — HAR: 1-2, 2-5, 13-19, 15-19, 18-21, 19-25, 20-1, 23-13, 24-22, 25-24, WISDM: 2-25, 7-2, 7-12, 7-26, 10-25, 12-19, 12-7, 13-2, 19-2, 28-20]

Extended version —
https://guohaoren.github.io/publications/ConFGD

5 Conclusion

In this work, we have proposed and developed a noteworthy and novel MTS UDA contrastive learning-based framework, ConFGD. First, we proposed and developed a novel FGD module to identify and align inherent inter-series frequency channel information and relationships for UDA of time series. Additionally, we introduced an FCCL framework to cooperate with the frequency embeddings to capture frequency-contextual representations and enhance prediction capabilities by extracting label-invariant information. Furthermore, we are pioneers in integrating the LLM as a time-saving and computationally efficient LAA module to further improve prediction accuracy by aligning the classification feature with LLM text-encoded labeled embeddings. Comprehensive experiments were conducted to validate the effectiveness and robustness of the proposed ConFGD and ConFGD+. Additionally, we upgraded other SOTA models with the LAA module to verify its superiority and effectiveness. Comparisons with various alternative SOTA approaches were made using two evaluation criteria on three benchmark datasets, effectively demonstrating the superiority and accuracy of our proposed method.

References

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

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

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

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

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

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C. K.; Li, X.; and Guan, C. 2021. Time-Series Representation Learning via Temporal and Contextual Contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2352–2359.

Gabor, D. 1946. Theory of communication. Part 3: Frequency compression and expansion. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26): 445–457.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59): 1–35.

Guo, H.; Zhu, H.; Wang, J.; Prahlad, V.; Ho, W. K.; de Silva, C. W.; and Lee, T. H. 2024. Remaining Useful Life Prediction via Frequency Emphasizing Mix-Up and Masked Reconstruction. *IEEE Transactions on Artificial Intelligence*.

He, H.; Queen, O.; Koker, T.; Cuevas, C.; Tsiligkaridis, T.; and Zitnik, M. 2023. Domain Adaptation for Time Series Under Feature and Label Shifts. In *International Conference on Machine Learning*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2019. Momentum Contrast for Unsupervised Visual Representation Learning. *arXiv preprint arXiv:1911.05722*.

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

Karlton, W. 2020. Time Frequency Analysis of Wavelet and Fourier Transform.

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

Li, Y.; Torralba, A.; Anandkumar, A.; Fox, D.; and Garg, A. 2020. Causal discovery in physical systems from videos. *Advances in Neural Information Processing Systems*, 33.

Liu, Q.; and Xue, H. 2021. Adversarial Spectral Kernel Matching for Unsupervised Time Series Domain Adaptation. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2744–2750. International Joint Conferences on Artificial Intelligence Organization.

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

OpenAI. 2024. Embeddings Guide. OpenAI Platform Documentation.

Ozyurt, Y.; Feuerriegel, S.; and Zhang, C. 2022. Contrastive learning for unsupervised domain adaptation of time series. *arXiv preprint arXiv:2206.06243*.

Purushotham, S.; Carvalho, W.; Nilanon, T.; and Liu, Y. 2022a. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations*.

Purushotham, S.; Carvalho, W.; Nilanon, T.; and Liu, Y. 2022b. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations*.

Ragab, M.; Eldele, E.; Tan, W. L.; Foo, C.-S.; Chen, Z.; Wu, M.; Kwoh, C.-K.; and Li, X. 2023. ADATIME: A Benchmarking Suite for Domain Adaptation on Time Series Data. *ACM Trans. Knowl. Discov. Data*.

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

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

Sun, B.; and Saenko, K. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. arXiv:1607.01719.

Tonekaboni, S.; Eytan, D.; and Goldenberg, A. 2021. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Wang, Y.; Xu, Y.; Yang, J.; Chen, Z.; Wu, M.; Li, X.; and Xie, L. 2023. SEnsor Alignment for Multivariate Time-Series

Unsupervised Domain Adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8): 10253–10261.

Wilson, G.; Doppa, J. R.; and Cook, D. J. 2020. 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*, KDD '20, 1768–1778. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379984.

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

Wu, Y.; Meng, X.; He, Y.; Zhang, J.; Zhang, H.; Dong, Y.; and Lu, D. 2024. Multi-view Self-Supervised Contrastive Learning for Multivariate Time Series. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 9582–9590.

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