CATS: Mitigating Correlation Shift for Multivariate Time Series Classification

Anonymous authors
Paper under double-blind review

Abstract

Unsupervised Domain Adaptation (UDA) leverages labeled source data to train models for unlabeled target data. Given the prevalence of multivariate time series (MTS) data across various domains, the UDA task for MTS classification has emerged as a critical challenge. In this work, we identify a key property of MTS: multivariate correlations vary significantly across domains, and further formalize this phenomenon as a novel type of domain shift, termed correlation shift. To mitigate correlation shift, we propose a scalable and parameter-efficient Correlation Adapter for MTS (CATS). Designed as a plug-and-play technique compatible with various Transformer variants, CATS employs temporal convolution to capture local temporal patterns and a graph attention module to model the changing multivariate correlation. The adapter reweights the target correlations to align the source correlations with a theoretically guaranteed precision. A correlation alignment loss is further proposed to mitigate correlation shift, bypassing the alignment challenge from the non-i.i.d. nature of MTS data. Extensive experiments on four real-world datasets demonstrate that (1) compared with vanilla Transformer-based models, CATS increases over 10% average accuracy while only adding around 1% parameters, and (2) all Transformer variants equipped with CATS either reach or surpass state-of-the-art baselines.

1 Introduction

Multivariate time series classification (MTS) is a fundamental task with applications spanning diverse fields, including finance (Zhao et al., 2023; Shahi et al., 2020; Mondal et al., 2014; LeBaron et al., 1999), healthcare (Zeger et al., 2006; Touloumi et al., 2004; Dockery & Pope, 1996; Bernal et al., 2017), climate science (Yoo & Oh, 2020; Ghil et al., 2002; Belda et al., 2014; Baranowski et al., 2015), transportation (Rezaei & Liu, 2019; MontazeriShatoori et al., 2020; Vu et al., 2018) and power systems (Hoffmann et al., 2020; Fütterer et al., 2017; Susto et al., 2018). Recently, deep learning models (Vaswani, 2017; Liu et al., 2023; Wu et al., 2022) have demonstrated remarkable capability in capturing temporal dependencies for MTS, showcasing significant promise in numerous applications.

However, the deployment of these models often encounters a critical challenge: *domain shifts* (Koh et al., 2021; Luo et al., 2019; Zhang et al., 2013) between the labeled source domain data during training and the target domain data during testing. The domain shift often leads to a notable degradation in model performance on the target domain. Moreover, obtaining labels for the test data is quite hard in most real-world scenarios (Ganin et al., 2016; Long et al., 2015). As a result, the UDA problem on MTS (He et al., 2023; Wilson et al., 2020) has emerged as a critical research area (He et al., 2023; Wilson et al., 2020), aiming to leverage the labeled source domain data to enhance the model performance on the unlabeled target domain.

Previous studies on UDA for MTS primarily focus on learning domain-invariant features through adversarial training (Wilson et al., 2020; 2023), contrastive learning (Ozyurt et al., 2022) or divergence metrics (He et al., 2023; Cai et al., 2021). However, these approaches have notable limitations: (1) Model architecture perspective: **existing UDA methods exhibit limited adaptability across model architectures.** Most prior approaches are tightly coupled with specific backbone networks, typically built on lightweight RNNs or CNNs (Liu & Xue, 2021; Wilson et al., 2020; He et al., 2023; Li et al., 2022b; Wilson & Cook, 2020). This coupling limits their scalability and applicability to more expressive and modern MTS analysis models, such as iTransformer (Liu et al., 2023) and Crossformer (Zhang & Yan, 2023). Compared with existing UDA

backbone networks, these models are typically Transformer variants[1] with larger parameter capacities and are better suited for modeling rich and complex temporal patterns in real-world time series. However, these architectures are not well supported by existing UDA frameworks. This gap motivates the key question: *Can we develop model-agnostic UDA methods that effectively enhance the adaptability of powerful general MTS models?* (2) Data distribution perspective: **Prior works lack architectural designs for addressing the variance of multivariate correlations across domains.** We observe consistent and significant shifts in inter-variable dependencies, i.e., correlations, across domains, and we term this new type of domain shift as **correlation shift**. Although a few existing studies (Wang et al., 2024a; Cai et al., 2021) implicitly acknowledge this issue, their primary focus has been on designing loss functions for alignment, with little discussion on how architectural modules can be tailored to fundamentally mitigate correlation shift. This oversight in model design leaves significant room for improvement on this challenge.

To overcome these limitations, we seek to align the correlation distributions between domains from both the model architecture and the training objective perspectives. At the model level, we propose a scalable and parameter-efficient adapter, termed Correlation Adapter for Multivariate Time Series (CATS). Specifically, to capture temporal dependencies, CATS employs depthwise convolutions along the temporal dimension for both down-projection and up-projection. Compared to traditional adapters that rely on linear matrices, convolutions demonstrate superior capability in capturing local temporal patterns. Building on this, CATS incorporates a Graph Attention Network (GAT) to adaptively reweight inter-variable dependencies in the hidden layers. Theoretically, proper reweighting can align the correlation distributions across domains, thus mitigating correlation shift. To adapt CATS to the unlabeled target domain, we introduce a novel correlation alignment loss. This loss function not only effectively reduces correlation shift but also circumvents the limitations of divergence metrics, which often fail on the non-i.i.d. (non-independent and identically distributed) time series data. By integrating CATS with this tailored loss function, we present a more effective and efficient solution for unsupervised domain adaptation in multivariate time series.

In summary, our main contributions are as follows:

- **Problem.** We are the first to empirically validate and mathematically define the concept of correlation shift in MTS data. As a unique and important property of MTS data, correlation shift introduces a distinct perspective for addressing UDA challenges.
- **Model.** We propose the first scalable and parameter-efficient MTS adapter, CATS, designed to be highly adaptable across different model architectures. Empirically and theoretically, CATS effectively mitigates correlation shifts while capturing local temporal patterns for classification.
- **Training objective.** We introduce a novel correlation alignment loss, which directly addresses correlation shift and circumvents the alignment challenge posed by the non-i.i.d. MTS data.
- **Evaluation.** We conduct extensive experiments on four real-world time series domain adaptation datasets. The results demonstrate that CATS consistently enhances the performance of MTS analysis models, achieving a 10%+ average accuracy improvement, even under large domain shifts. Furthermore, MTS analysis models equipped with CATS outperform SOTA baselines by around 4% average accuracy, showcasing the superb effectiveness and adaptability of CATS.

2 Preliminaries

Multivariate time series classification. In the task of multivariate time series (MTS) classification, the dataset is comprised of a collection of time series samples along with their corresponding labels, denoted as $\mathcal{D} = \{(\mathbf{X}_i, y_i)\}_{i=1}^n$ with n being the sample number. Here, the i-th sample $\mathbf{X}_i \in \mathbb{R}^{D \times T}$ represents an individual time series that contains readouts of D observations over T time points, and y_i is the associated label. In this paper, we use $\mathbf{X}[j]$ to represent the j-th variable of the sample $\mathbf{X}$.

[1]As general MTS models are mostly Transformer-based, our discussion in this paper mainly focuses on Transformer variants. However, CATS can be easily extended to other block-wise non-Transformer architectures like TimesNet.

Adapters for large models. Recently, large-scale Transformers have achieved great success in various domains, including natural language processing (Vaswani, 2017; Devlin, 2018; Brown et al., 2020), computer vision (Radford et al., 2021; Alexey, 2020), and time series analysis (Liu et al., 2023; Wu et al., 2022). However, their large parameter size makes it impractical to fine-tune a separate model for every downstream task. To overcome this, a variety of parameter-efficient fine-tuning (PEFT) methods have been proposed (Han et al., 2024; Hu et al., 2021; Xu et al., 2023). Among them, *adapters* have attracted particular interest for their ability to transfer the knowledge of pretrained models to new tasks using only a small number of additional parameters.

Given the high similarity between the objectives of adapters and UDA, many studies (Zhang et al., 2021; Malik et al., 2023) have leveraged adapters to transfer knowledge learned from the source domain to the target domain. Such domain adapters are embedded between two consecutive Transformer blocks to adapt the model's learned representations to the target domain. Mathematically, these adapters can be expressed as:

$$\mathbf{H}_O^{(k)} = \mathbf{H}_I^{(k)} + \sigma(\mathbf{H}_I^{(k)}\mathbf{W}_{\downarrow}^{(k)})\mathbf{W}_{\uparrow}^{(k)} \tag{1}$$

where $\sigma(\cdot)$ represents the activation function, and $\mathbf{W}_{\downarrow}^{(k)} \in \mathbb{R}^{T\times r}$ and $\mathbf{W}_{\uparrow}^{(k)} \in \mathbb{R}^{r\times T}$ are the two linear matrices for down-projection and up-projection with r being a small hyperparameter. Here, $\mathbf{H}_I^{(k)}$ is the input of the k-th adapter block, and $\mathbf{H}_O^{(k)}$ is the output of the k-th adapter, i.e., the input of the $(k+1)$-th Transformer block.

Unsupervised domain adaptation. The goal of UDA is to leverage information from a labeled source domain $\mathcal{D}_s = \{(\mathbf{X}_{i,s}, y_{i,s})\}_{i=1}^{n_s}$ to enhance the model's understanding of an unlabeled target domain $\mathcal{D}_t = \{\mathbf{X}_{i,t}\}_{i=1}^{n_t}$. Generally, source and target samples are independently sampled from their respective distributions, i.e., $\mathcal{D}_s \sim \mathcal{P}_s(\mathbf{X}_s, y_s)$ and $\mathcal{D}_t \sim \mathcal{P}_t(\mathbf{X}_t, y_t)$. However, these distributions often exhibit significant shifts. There are two widely studied shifts: feature shift (Zhang et al., 2013) and label shift (Azizzadenesheli et al., 2019). Specifically, feature shift occurs when the distribution of features changes across domains, while the relationship between features and labels remains consistent. In contrast, label shift arises when the label distributions differ between domains, even if the feature distributions are similar.

3 Correlation Shift

Although label shift and feature shift are the two most commonly analyzed types of domain shifts in UDA tasks, focusing solely on these shifts is insufficient for MTS classification. A key characteristic of MTS is the interaction between different variables, such as the interplay between blood glucose levels and insulin in the human body (Basu et al., 2009; Wang et al., 2018). Correlation effectively models this inter-variable dependencies, thus making it central to many statistical and deep learning models for MTS (Box et al., 2015; Bollerslev, 1990; Wu et al., 2021).

Despite its importance, to our best understanding, no prior work has provided systematic research on the correlation distribution across domains for MTS data. To bridge this gap, we introduce a novel domain shift tailored specifically for MTS: correlation shift.

Definition 1 (Correlation shift)**.** *Suppose the source multivariate data $\mathbf{X}_s \in \mathbb{R}^{D\times T}$ and the target multivaraite data $\mathbf{X}_t \in \mathbb{R}^{D\times T}$ follow the source distribution $\mathcal{P}_s$ and the target distribution $\mathcal{P}_t$, respectively. Here, D denotes the number of variables and T represents the feature dimension. Then,* ***correlation shift*** *occurs when the multivariate correlations between the source and target domains differ, formally defined as:*

$$\mathrm{Corr}(\mathbf{X}_s) \neq \mathrm{Corr}(\mathbf{X}_t) \tag{2}$$

where the correlation structure $\mathrm{Corr}(\cdot)$ *is given by:*

$$\begin{aligned} &\mathrm{Corr}(\mathbf{X}) := \mathrm{diag}(\mathbf{\Sigma})^{-1/2}\mathbf{\Sigma}\mathrm{diag}(\mathbf{\Sigma})^{-1/2} \\ &\mathbf{\Sigma} = \mathbb{E}_{\mathbf{X}\sim\mathcal{P}}\left[(\mathbf{X}-\mathbb{E}\mathbf{X})(\mathbf{X}-\mathbb{E}\mathbf{X})^T\right] \end{aligned} \tag{3}$$

This phenomenon naturally arises from discrepancies in inter-variable dependencies across domains. A practical example of the correlation shift can be observed in healthcare analytics. For example, in non-diabetic

individuals, there is typically a synchronous peak in blood glucose and insulin levels following sugar intake while in diabetic patients, the increase in insulin occurs with a noticeable delay after the peak in blood glucose (Basu et al., 2009; Wang et al., 2018). This delay represents a clear correlation shift when considering blood glucose and sugar intake as two interacting variables. Another widely-existing example comes from the weather data. Extensive studies (Draper & Long, 2004; Weissman et al., 2002; Back & Bretherton, 2005) have shown that the relationship between wind speed and precipitation varies geographically and this relationship tends to be significantly stronger in humid regions compared to arid areas. The widespread occurrence of correlation shifts impacts the transferability of learned representations, ultimately leading to deteriorated performance on target domains

To further validate the universality of correlation shifts, we conduct an empirical analysis on a real-world Human Activity Recognition (HAR) dataset (Anguita et al., 2013), to demonstrate the discrepancy in the correlation across different domains. Specifically, we iterate through the 30 domains in HAR, treating each domain as the source domain while considering the remaining 29 domains as target domains. For each source-target domain pair, we apply the Mann-Whitney U test (McKnight & Najab, 2010), a non-parametric hypothesis testing method, to determine whether there is a significant correlation shift between the source and the target domains. A detailed explanation for this correlation shift test is provided in Appendix A. We compute the rate of target domains suffering from significant correlation shifts for each source domain, and the results are shown in Figure 1, where the orange bars indicate the rate of target domains with significant correlation shifts. The red dashed line in Figure 1 marks the average rate of correlation shift, which is 78%. These findings provide clear evidence that correlation shifts are indeed prevalent in MTS datasets, thereby calling for solutions to mitigate them.

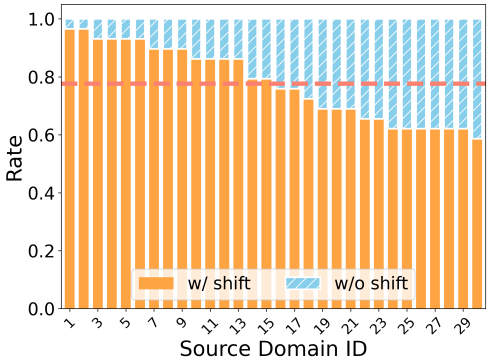


Figure 1: Rates of target domains with correlation shifts per source domain. The x-axis represents the source domain index while the y-axis indicates the rate of correlation shifts among the rest 29 domains. The red line marks the average rate of 78%.

4 Methodology

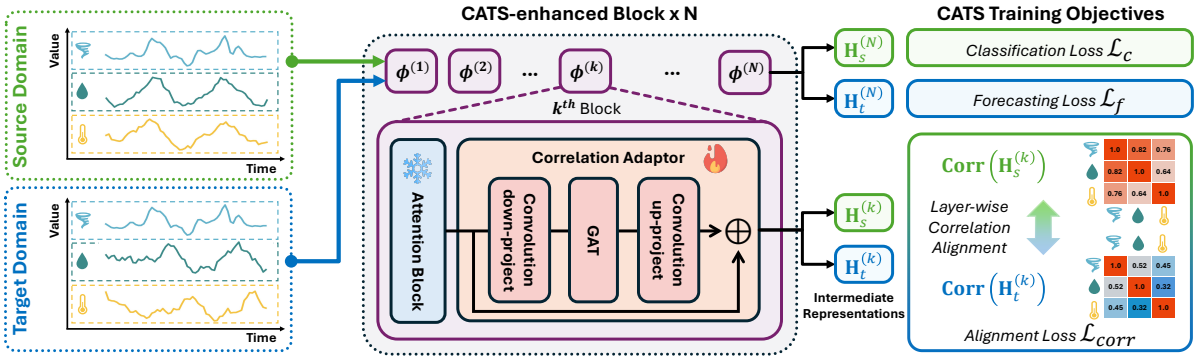


Figure 2: The main framework of CATS. CATS is integrated after each attention block of any Transformer variant, with only CATS trained and the backbone frozen. The training objective involves three loss functions: (1) classification loss on the labeled source domain, (2) forecasting loss on the unlabeled target domain, and (3) layer-wise correlation alignment loss to align these two domains.

In this section, we introduce our solution to mitigate correlation shift. We first propose CATS in Section 4.1, which demonstrates superior representation learning capabilities for MTS compared to traditional adapters. By reweighting multivariate correlation, CATS enjoys theoretical guarantees for mitigating correlation shift. In Section 4.2, we propose a novel training objective for CATS on unlabeled target domains, centered on

the correlation alignment loss to address correlation shift effectively. The overall framework of CATS is visualized in Figure 2.

4.1 CATS: Temporal-aware Correlation Adapter

MTS data exhibit two prominent properties: temporal dependencies and inter-variable dependencies. To model these properties, we first introduce up-project and down-project modules for time series adapters in Section 4.1.1 to capture temporal dependencies effectively. In Section 4.1.2, we further propose a reweighting module to adaptively refine inter-variable dependencies, thereby mitigating the impact of correlation shift. By integrating these components, we propose CATS which effectively enhances the model's adaptability in domain adaptation tasks involving multivariate time series classification.

4.1.1 Temporal Project via Convolution

As discussed in Section 2, adapters hold significant potential for addressing domain adaptation challenges in Transformer models. However, these previous adapters often rely on the assumption that the data are i.i.d. (independent and identically distributed), which does not hold for MTS. A key property of MTS is that temporally adjacent data points often exhibit strong similarity. However, the use of linear matrices in existing adapters fails to capture this local similarity, leading to noticeable declines in performance. Inspired by temporal convolution network (TCN) (Fan et al., 2023; Farha & Gall, 2019; Hewage et al., 2020), we posit that convolutions on temporal dimension could better leverage local similarity on MTS, thus serving as a better substitute as project layer, compared with linear matrices in adapters from Eq. equation 1.

One potential drawback of using convolutions along the temporal dimension is the increase of the number of trainable parameters. On previous adapters in Eq. equation 1, the parameter complexity of the linear matrices are $\mathcal{O}(T \times r)$. In contrast, convolutions have a parameter complexity of $\mathcal{O}(D^2 \times r)$, where r is the kernel size. When the hidden layer dimension D approaches or exceeds the time length T, the number of trainable parameters for convolutions can become quite large. To address this issue, we adopt depthwise convolutions (Chollet, 2017), where each variable is convolved with its own kernel. This approach reduces the parameter complexity to $\mathcal{O}(D \times r)$, significantly improving efficiency. Note that depthwise convolutions ignore multivariate correlations. We will address this issue in Section 4.1.2.

To empirically validate the superiority of temporal depthwise convolutions (TDC), we compare the performance between a typical adapter in Eq. equation 1 and a TDC-based adapter which only uses temporal depthwise convolutions instead of the linear layers. Specifically, given a backbone pretrained on the source dataset, we will train these two adapters on the target domain. Note that the task is designed to assess the representation learning ability of different adapters on time series rather than focusing solely on domain adaptation. Therefore, we use the labels from the target domain when training adapters. A detailed experimental setting is provided in Appendix B. The experiment results, illustrated in Figure 3, indicate that the TDC-based adapter demonstrates significantly higher accuracy (around 10% improvement) after both adapters converge. This clearly supports the premise that, for time series data, temporal convolutions are a superior alternative to traditional linear matrices.

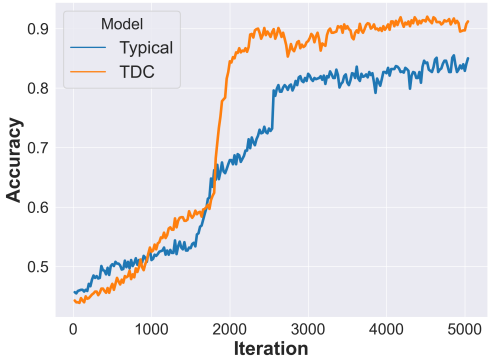


Figure 3: The accuracy comparison on HAR dataset between the typical adapter and the TDC-based adapter. With the backbone (TimesNet) pretrained on the domain 1, both adapters are trained on the domain 10.

4.1.2 Correlation Alignment via Reweighting

In this section, our objective is to identify an effective reweighting module to mitigate the correlation shift. Since correlation shift arises from discrepancies in multivariate correlations between the source and target domains, a natural approach to addressing it is to adaptively reweight the correlations in the target domain.

Interestingly, in the case of Gaussian variables, we prove that a simple linear mapping[2] is sufficient to serve as an optimal reweighting function to align not only the correlation but also the joint distribution between variables. Mathematically, this finding could be formalized as Proposition 1. All proofs in this section are postponed to Appendix C.

Proposition 1 (Gaussian Probability Alignment)**.** *Suppose source data* $\mathbf{X}_s \in \mathbb{R}^{D\times T}$ *and target data* $\mathbf{X}_t \in \mathbb{R}^{D\times T}$ *follow* $\mathcal{N}_s(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$ *and* $\mathcal{N}_t(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$*, respectively. There exists a reweighting matrix* $\mathbf{A} \in \mathbb{R}^{D\times D}$ *and a bias vector* $\mathbf{b} \in \mathbb{R}^D$*, such that the multivariate joint probability of the reweighted target domain perfectly aligns with that of the source domain, that is for every* $i, j = 1, 2, ..., D$

$$\Pr\left(\mathbf{X}_s[i], \mathbf{X}_s[j]\right) = \Pr\left(\mathbf{Y}[i], \mathbf{Y}[j]\right)$$

where $\mathbf{Y} = \mathbf{A}\mathbf{X}_t + \mathbf{b}$ *and* $\mathbf{b} = \mathbf{0}$ *for most MTS data.*

This insight offers a promising direction for designing effective solutions without resorting to complex and computationally expensive methods. Importantly, even when the distributions of random variables are complex and difficult to characterize precisely, this simple linear mapping approach can still perfectly match the correlation between variables, thereby effectively mitigating correlation shifts, as shown in the following proposition.

Proposition 2 (Correlation Alignment)**.** *Suppose source data* $\mathbf{X}_s \in \mathbb{R}^{D\times T}$ *and target data* $\mathbf{X}_t \in \mathbb{R}^{D\times T}$ *follow the source distribution* $\mathcal{P}_s$ *and the target distribution* $\mathcal{P}_t$*, respectively. There exists a reweighting matrix* $\mathbf{A} \in \mathbb{R}^{D\times D}$*, such that the correlation of the source distribution and target distribution can be perfectly aligned, formally expressed as:*

$$\mathrm{Corr}(\mathbf{X}_s) = \mathrm{Corr}(\mathbf{A}\mathbf{X}_t) \tag{4}$$

Intuitively, the interaction between variables can be considered as a fully-connected graph, where each node represents a variable and edges model the inter-variable dependency, and the reweighting matrix $\mathbf{A}$ serves as the adjacency matrix of the graph. However, in practical scenarios, solving the reweighting matrix $\mathbf{A}$ is often computationally expensive and non-trivial, particularly when the distributions of the variables are highly complex. To address this, we leverage a Graph Attention Network (GAT) (Velickovic et al., 2017) to adaptively approximate the matrix $\mathbf{A}$. Mathematically, we formalize this insight through the following theorem:

Theorem 1 (Attention Approximation)**.** *The optimal reweighting matrix* $\mathbf{A}$ *in Proposition 2 can be approximated by an attention matrix* $\tilde{\mathbf{A}}$ *with an arbitrary precision, where* $\tilde{\mathbf{A}}$ *is generated by a one-layer Graph Attention Network with an infinite hidden dimension.*

By integrating the GAT and TDCs, we finally propose CATS to both well capture temporal dependencies and solve correlation shift. Mathematically, the CATS layer $\phi(\cdot)$ could be expressed as:

$$\phi(\mathbf{X}) = \mathbf{X} + \mathrm{TDC}_{\uparrow}\left(\sigma(\mathrm{GAT}\left(\mathrm{TDC}_{\downarrow}\left(\mathbf{X}\right)\right))\right) \tag{5}$$

where $\mathrm{TDC}_{\downarrow}$ and $\mathrm{TDC}_{\uparrow}$ represent the temporal convolution layers for down-project and up-project, respectively. Here, GAT represents one GAT layer on a fully connected graph. We prove in Appendix C.4 that CATS in Eq. equation 5 could also approximate the reweighting matrix $\mathbf{A}$ in a manner similar to a one-layer GAT.

Similar to typical domain adapters in Eq. equation 1, we integrate CATS within two consecutive blocks of a Transformer-based model. Since Transformers always consist of multiple blocks, this allows each instance of CATS to make minor adjustments to the target domain's distribution. Cumulatively, these incremental adjustments are capable of mitigating large domain shift from the final block. The idea of gradually reducing domain shift is conceptually similar to but bears subtle difference from gradual domain adaptation (He et al., 2024). Gradual domain adaptation leverages intermediate domains for supervision, which are often predefined. In contrast, CATS does not rely on any intermediate domain, making it a more flexible and efficient solution for UDA.

[2]Although adapters in Eq. equation 1 contains linear matrices, it is not a linear mapping due to the existence of the activation function.

4.2 Training Procedures

This section introduces the training objective for CATS on the unlabeled target domain. The goal is to preserve the pre-trained model's rich knowledge for accurate time series classification while leveraging CATS to effectively minimize distribution shifts across domains.

To achieve this, we first propose the correlation alignment loss, specifically designed to address correlation shift. Given CATS's ability to adaptively reweight multivariate correlation, as discussed in Section 4.1.2, our objective is to align the correlation distribution of the source domain with that of the target domain transformed by CATS. Specifically, given the output of the k-th block $\mathbf{H}_s^{(k)}$ and $\mathbf{H}_t^{(k)}$ from the source domain and the target domain respectively, we minimize the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) of the correlation distribution between $\mathbf{H}_s^{(k)}$ and $\phi(\mathbf{H}_t^{(k)})$. Mathematically, the correlation alignment loss can be expressed as:

$$\mathcal{L}_{\texttt{corr}} = \sum_{k=1}^{K} \sum_{\substack{\mathbf{H}_s^{(k)} \\ \mathbf{H}_t^{(k)}}} \text{MMD}\left(\text{corr}\left(\mathbf{H}_s^{(k)}\right), \text{corr}\left(\phi\left(\mathbf{H}_t^{(k)}\right)\right)\right)$$

where $\text{corr}(\mathbf{H}) = \text{vec}\left(\frac{\mathbf{H}\mathbf{H}^T}{\|\mathbf{H}\|_{\mathcal{F}}^2}\right)$, $\text{MMD}(\cdot,\cdot)$ denotes the MMD loss, $\text{vec}(\cdot)$ denotes the vectorization operator and $\phi(\cdot)$ represent our CATS adapter defined in Eq. 5.

Compared to directly aligning hidden features using MMD, the correlation alignment loss offers a unique advantage in terms of optimization difficulties. This is primarily because MMD assumes that data distributions are i.i.d., while time series data inherently exhibit non-i.i.d. characteristics. Consequently, directly applying MMD to align feature distributions often increases the difficulty of optimization, potentially leading to suboptimal performance. In contrast, correlation alignment loss focuses on aligning correlations rather than directly aligning raw features. Within the same domain, these correlations across variables tend to be more stable compared to the feature distributions. For instance, in a financial time series dataset, the correlation between stock prices of two closely related companies might remain consistent over time, even though the individual stock price values fluctuate significantly (Kim & Baginski, 2016). Thus, within a single domain, if we consider the correlation of MTS data as a new "feature", this "feature" tends to exhibit higher similarity across different samples. Consequently, using MMD to align correlations becomes less challenging in terms of optimization. Therefore, this property makes correlation alignment loss a more stable and effective approach for reducing distributional discrepancies, particularly in MTS tasks, where temporal dependencies and multivariate correlation play a crucial role

In addition to mitigating correlation shift, it is crucial to enhance CATS's ability to understand the task and data accurately. To ensure that the features extracted by CATS are beneficial for the classification task, we compute a classification loss $\mathcal{L}_{\texttt{c}}$ on the labeled source domain. Furthermore, to improve CATS's understanding of the input features from the unlabeled target domain, we introduce a forecasting loss $\mathcal{L}_{\texttt{f}}$ on the target domain, encouraging CATS to accurately capture local temporal patterns. These two losses are discussed in Appendix D in detail. To sum up, the final loss can be computed as follows:

$$\mathcal{L} = \mathcal{L}_{\texttt{c}} + \lambda_{\texttt{corr}}\mathcal{L}_{\texttt{corr}} + \lambda_{\texttt{f}}\mathcal{L}_{\texttt{f}} \tag{6}$$

where $\lambda_{\texttt{f}}$ and $\lambda_{\texttt{corr}}$ are two hyperparameters.

5 Experiments

5.1 Experimental Settings

Datasets. We conduct experiments on 4 real-world datasets, including HAR (Anguita et al., 2013), WISDM (Weiss, 2019), HHAR (Stisen et al., 2015), and Boiler (Cai et al., 2021). For HAR, HHAR, and WISDM datasets, we rank all possible source-target domain pairs based on the magnitude of domain shift, dividing them into 10 groups in the ascending order. From each group, we select one source-target domain pair for evaluation. For Boiler, due to its limited number of domains, we choose the domain pairs with the largest

Table 1: Main accuracy results for MTS classification on the UDA task. The higher the accuracy is, the better. For three Transformer variants, the columns of 'w/o CATS', 'w/ CATS' and 'Δ' represent the accuracy without CATS, the accuracy with CATS, and the accuracy improvement due to CATS. **Bold font** indicates the best performance across **all the methods**, and underline symbol represents the best performance among UDA baselines.

Dataset	UDA Baseline					Transformer			TimesNet			Crossformer			iTransformer		
Source → Target	CORAL	Raincoat	CLUDA	SASA	UDApter	w/o CATS	w/ CATS	Δ	w/o CATS	w/ CATS	Δ	w/o CATS	w/ CATS	Δ	w/o CATS	w/ CATS	Δ
HAR 24 → 27	78.76	96.88	82.14	86.72	96.46	91.81	98.23	6.42	93.69	97.34	3.65	80.53	93.80	13.27	82.30	**99.11**	16.81
HAR 3 → 13	63.63	91.67	77.55	78.78	90.90	79.79	**98.98**	19.19	84.96	87.86	2.90	84.84	96.96	12.12	75.75	97.97	22.22
HAR 16 → 13	47.47	71.87	69.39	61.61	66.67	73.96	77.78	3.82	67.67	83.84	16.17	69.69	87.87	18.18	74.74	**85.86**	11.12
HAR 3 → 8	51.76	78.13	78.57	64.70	71.76	54.11	75.12	21.01	64.70	**92.92**	28.22	61.17	62.35	1.18	74.11	91.77	17.66
HAR 19 → 2	61.53	76.56	60.00	69.23	59.34	53.84	73.52	19.68	53.84	82.41	28.57	48.35	53.84	5.49	59.34	**84.61**	25.27
HAR 11 → 28	60.86	73.95	64.91	76.52	66.95	66.95	77.40	10.45	70.43	**80.00**	9.57	47.82	78.26	30.44	57.39	77.40	20.01
HAR 16 → 10	50.56	71.88	68.42	56.17	61.79	35.95	68.54	32.59	62.92	72.91	9.99	61.79	78.26	16.47	67.41	**87.64**	20.23
HAR 25 → 10	19.10	57.81	57.89	56.18	56.18	48.31	57.40	9.09	46.06	65.17	19.11	52.80	**71.91**	19.11	47.19	65.17	17.98
HAR 18 → 10	37.07	48.43	57.89	37.07	46.06	35.95	59.55	23.60	38.20	**69.66**	31.46	44.94	62.92	17.98	40.44	48.51	8.07
HAR 19 → 10	44.94	50.21	49.12	39.32	43.82	37.07	49.48	12.41	42.94	46.56	3.62	66.29	**64.04**	-2.25	37.07	50.56	13.49
HAR Average	51.57	71.74	66.59	62.63	65.99	57.77	73.59	15.22	62.54	77.87	15.33	61.82	75.02	13.20	61.57	**78.86**	17.29
WISDM 12 → 9	82.71	91.35	82.50	75.30	83.95	82.50	85.19	2.69	72.83	**92.60**	19.77	66.67	91.36	24.69	66.67	90.12	23.45
WISDM 5 → 31	59.03	80.72	82.93	75.90	82.93	75.90	74.70	-1.20	81.92	81.92	0.00	67.46	**93.97**	26.51	65.06	83.13	18.07
WISDM 25 → 31	48.19	61.44	53.66	43.47	59.03	56.62	**61.44**	4.82	57.83	60.24	2.41	37.34	43.47	6.13	44.57	59.03	14.46
WISDM 0 → 30	65.04	61.16	62.75	63.10	59.03	58.22	60.19	1.97	58.22	61.16	2.94	62.13	**77.66**	15.53	58.25	62.14	3.89
WISDM 10 → 22	61.67	73.33	76.67	51.66	71.67	71.67	76.67	5.00	73.00	76.67	3.67	66.67	**91.67**	25.00	56.67	78.33	21.66
WISDM 12 → 2	36.59	53.65	63.41	58.53	41.46	48.78	62.19	13.41	48.78	46.34	-2.44	51.21	**81.70**	30.49	51.21	67.07	15.86
WISDM 6 → 11	43.42	56.57	56.10	47.36	41.46	43.36	42.10	-1.26	56.57	59.21	2.64	42.10	62.10	20.00	27.63	**63.15**	35.52
WISDM 11 → 21	28.84	38.46	58.54	40.38	30.76	18.84	19.23	0.39	38.46	**59.61**	21.15	30.76	58.54	27.78	17.30	55.76	38.46
WISDM 19 → 3	7.69	15.38	51.22	50.00	23.07	19.23	38.46	19.23	23.07	42.30	19.23	11.53	**53.84**	42.31	19.92	19.23	-0.69
WISDM 3 → 11	38.16	21.36	48.78	25.00	15.78	17.10	18.42	1.32	15.78	**60.52**	44.74	15.79	22.36	6.57	13.15	18.42	5.27
WISDM Average	47.13	55.34	63.66	53.07	50.91	49.22	53.86	4.64	52.65	64.06	11.41	45.17	**67.67**	22.50	42.04	59.64	17.60
HHAR 7 → 3	55.57	94.08	85.09	79.86	86.87	61.26	88.96	27.70	83.58	94.96	11.38	74.17	**95.19**	21.02	84.87	92.24	7.37
HHAR 6 → 7	56.99	84.37	76.15	58.24	83.50	74.15	84.55	10.40	58.87	89.56	30.69	62.42	81.00	18.58	75.15	**93.32**	18.17
HHAR 6 → 3	48.14	74.33	65.79	66.52	65.86	57.55	83.58	26.03	62.36	75.27	12.91	64.55	81.83	17.28	67.36	**84.47**	17.11
HHAR 6 → 5	45.47	75.58	45.47	61.70	45.64	47.38	66.54	19.16	49.90	70.98	21.08	49.32	75.43	26.11	42.15	**76.40**	34.25
HHAR 7 → 5	36.75	63.47	48.06	57.05	45.64	43.52	63.63	20.11	54.35	66.53	12.18	35.97	66.53	30.56	43.32	**70.99**	27.67
HHAR 0 → 7	41.97	68.32	33.89	34.34	62.83	44.25	**71.81**	27.56	38.20	68.47	30.27	51.15	70.35	19.20	56.57	66.97	10.40
HHAR 4 → 0	22.54	23.66	34.73	25.16	25.16	23.72	24.94	1.22	23.63	28.67	5.04	21.88	**37.20**	15.32	27.32	31.29	3.97
HHAR 3 → 0	26.70	17.41	**35.15**	22.10	22.10	9.63	23.20	13.57	17.25	20.56	3.31	23.41	28.22	4.81	25.16	29.54	4.38
HHAR 2 → 7	5.42	54.68	26.36	32.98	41.33	34.65	58.89	24.24	41.54	**69.10**	27.56	43.63	58.03	14.40	44.25	63.63	19.38
HHAR 1 → 0	36.19	57.32	47.65	45.90	44.30	41.27	58.69	17.42	44.30	60.97	16.67	19.47	31.29	11.82	48.95	**64.28**	15.33
HHAR Average	37.57	61.44	49.83	48.39	52.32	43.74	62.48	18.74	47.40	64.51	17.11	44.60	62.51	17.91	51.51	**67.31**	15.80
Boiler 1 → 2	93.76	97.05	97.29	97.33	92.64	91.98	97.86	5.88	97.86	**98.15**	0.29	94.92	98.11	3.19	91.51	**98.15**	6.64
Boiler 3 → 2	87.16	95.02	87.16	96.05	92.17	90.83	**98.15**	7.32	92.17	97.84	5.67	97.68	98.11	0.43	97.47	**98.15**	0.68
Boiler Average	90.46	96.03	92.22	96.69	92.41	91.41	98.00	6.59	95.02	98.00	2.98	96.30	98.11	1.81	94.49	**98.15**	3.66

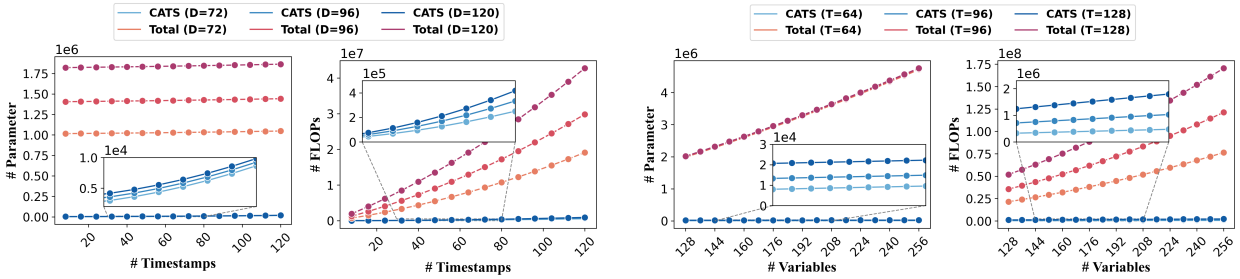
or the smallest domain shift. Detailed descriptions of datasets and domain pair selection are provided in Appendix E and F, respectively.

Baselines. We compare CATS with three different types of UDA methods, including (1) correlation-related UDA: CORAL, (2) MTS UDA: SASA, CLUDA, and Raincoat, and (3) adapter-based UDA: UDApter. Descriptions of baseline methods are in Appendix G

Parameter settings. Unless otherwise specified, we use default hyperparameter settings in the released code of corresponding publications. For CATS, we use TDCs with a kernel size $r = 5$ and a padding of 2. For training, we use Adam optimizer with a learning rate of 1e-4, and set $\lambda_c = 0.5$ and $\lambda_f = 0.5$. We evaluate the performance of CATS on three different Transformer-based MTS models: Crossformer(Zhang & Yan, 2023), Transformer (Vaswani, 2017), TimesNet (Wu et al., 2022), and iTransformer (Liu et al., 2023). The implementation details are provided in Appendix I.

5.2 Experimental Results

Main results. The main evaluation results of the accuracy are presented in Table 1. On each dataset in the table, the difficulty of UDA tasks for source-target domain pairs increases progressively from top to bottom. The experimental results reveal three noteworthy conclusions: **(1) CATS significantly enhances the UDA classification performance of general MTS models.** Specifically, on four datasets CATS

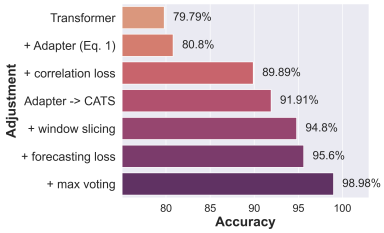


(a) Parameter curve with varying T (b) FLOP curve with varying T (c) Parameter curve with varying D (d) FLOP curve with varying D

Figure 4: The parameter (FLOP) curve of CATS on Transformer with varying variable number D and time length T. The three red curves represent the parameter counts (or FLOPs) of the full Transformer model, while the three blue curves represent CATS alone. In Figure (c), the three red curves overlap due to their relatively small differences compared to the large overall values.

improves the average accuracy of Transformer, TimesNet, Crossformer, and iTransformer on the target domain by 18.56%, 17.60%, 15.80%, and 3.66% accuracy, respectively. Please note that although CATS are initially designed for Transformers, it still performs well on TimesNet (a CNN model), which highlights CATS's high adaptability across different architectures. (2) **CATS could have stable performance even under large shifts.** Even under scenarios with the largest domain shifts, such as HAR 19 → 10 and HHAR 1 → 10, CATS demonstrates robust performance, delivering 6.82% and 15.31% improvement on average for all four models. These results clearly validate the effectiveness of CATS, even in scenarios with large domain shifts. **(3) CATS-enhanced MTS models outperform state-of-the-art (SOTA) baselines in classification accuracy.** Across all four datasets, CATS-enhanced models achieve the best performance, with average accuracy improvements of 7.12%, 0.40%, 5.87%, and 1.46% accuracy, respectively, compared to SOTA baselines. These results highlight the superiority of CATS in addressing UDA challenges for MTS data.

Scalability evaluation. To validate the scalability of CATS, we adjust the time series length T and the number of variables D of the Transformer, recording the parameter count and FLOPs (Floating Point Operations per Second) for CATS and the full model. The experimental results are shown in Figure 4. The results reveal that, regardless of the values of T and D, CATS consistently requires two orders of magnitude fewer parameters and FLOPs compared to the full model. Interestingly, as the hidden layer dimension D increases, the parameter count and FLOPs of the Transformer exhibit quadratic growth, whereas CATS scales linearly. This observation confirms CATS's suitability for large-scale MTS tasks with varying input dimensions.



(a) Step-by-Step accuracy improvement on HAR 3 → 13 scenario from vanilla Transformer to Transformer enhanced by CATS.

Scenario	Vanilla	Corr	CATS
24 → 27	91.81	97.34	98.23
16 → 13	73.96	77.78	77.78
19 → 2	53.84	70.33	73.52
16 → 10	39.95	64.00	68.54
18 → 10	35.95	58.43	59.55
Average	59.10	73.58	75.52

(b) Loss ablation study. *Vanilla* denotes only using classification loss, *Corr* introduces additional correlation loss, and *CATS* represents using all the losses in this paper.

Scenario	Vanilla	GAT	TDC	CATS
24 → 27	91.81	96.46	96.46	98.23
16 → 13	73.96	76.10	78.40	77.78
19 → 2	53.84	70.33	72.53	73.52
16 → 10	39.95	66.29	62.91	68.54
18 → 10	35.95	58.43	57.30	59.55
Average	59.10	73.52	73.52	75.53

(c) Module ablation study. *Vanilla* represents original models, *GAT* / *TDC* using a GAT / TDC module as the adapter, and *CATS* denotes using our adapter.

Figure 5: Ablation study on the HAR dataset.

Ablation Study. To validate the effectiveness of CATS and the proposed loss in Eq. equation 6, we perform ablation studies on the HAR dataset using a Transformer backbone. We first present stepwise results on the $3 \to 13$ scenario, incrementally adding components from the vanilla Transformer to the CATS-enhanced version (details in Appendix H). As shown in Figure 5a, each component yields consistent improvements in target domain accuracy. Furthermore, we performed a detailed loss ablation study by fixing the model architecture to the CATS-enhanced Transformer, with results shown in Figure 5b. We also conducted a module ablation study using the full set of loss functions, with results reported in Figure 5c. Together, these analyses comprehensively demonstrate that each design in CATS consistently contributes to improved UDA performance across diverse scenarios.

6 Related Works

Unsupervised Domain Adaptation. Unsupervised domain adaptation (UDA) leverages labeled data from a source domain to make predictions on an unlabeled target domain and has gained traction across various fields (Ganin & Lempitsky, 2015; Zhang et al., 2018; Ramponi & Plank, 2020; Liu et al., 2021). Existing UDA methods generally fall into three categories: (1) *Adversarial training* uses a domain discriminator to distinguish domains while training the model to extract domain-invariant features (Hoffman et al., 2018; Long et al., 2018; Tzeng et al., 2015); (2) *Multi-task supervision* introduces auxiliary tasks, such as data augmentation (Volpi et al., 2018) or reconstruction (Ghifary et al., 2016; Zhuo et al., 2017), to guide feature learning; (3) *Statistical divergence* methods reduce domain gaps using metrics like MMD (Yan et al., 2017; Zhang & Wu, 2020; Yan et al., 2019), optimal transport (Courty et al., 2017; 2016), or contrastive domain discrepancy (CDD) (Kang et al., 2019). CORAL-based methods (Sun & Saenko, 2016; Lee et al., 2019; Li et al., 2022a) also align feature correlations. However, unlike CATS, these approaches ignore the temporal structure of MTS and the importance of aligning distribution means, as discussed in Appendix J.

Unsupervised domain adaptation for time series. While adaptation methods have achieved significant success in computer vision, relatively fewer approaches have been developed to address the unique challenges of domain adaptation for time series data. CoDATS (Wilson et al., 2020) employ domain discriminators for temporal feature alignment. SASA (Cai et al., 2021) aligns invariant unweighted spare associative structures for time series data. RainCoat (He et al., 2023) utilizes MMD to minimize frequency feature distribution in a polar coordinate across domains. Additionally, CLUDA (Ozyurt et al., 2022) leverage contrastive learning to enhance model robustness with data augmentations, while LogoRA (Zhang et al., 2024) combines global and local feature analysis to maintain domain-invariant representations for complex time series structures. More discussion on related works are provided on Appendix K.

7 Conclusion

In this paper, we study the problem of unsupervised domain adaptation for multivariate time series classification. We begin by identifying a previously overlooked domain shift in MTS data: correlation shift, where correlations between variables vary across domains. To mitigate this shift, we propose a scalable and parameter-efficient adapter, CATS, serving as a plug-and-play technique compatible with various Transformer variants. Supported by a solid theoretical foundation for mitigating correlation shift, CATS effectively captures dynamic temporal patterns while adaptively reweighting multivariate correlations. To further reduce correlation discrepancies, we introduce a correlation alignment loss, which aligns multivariant correlations across domains, addressing the non-i.i.d. nature of MTS data. Extensive evaluations on real-world datasets demonstrate that CATS consistently and significantly improves the accuracy of Transformer backbones while introducing minimal additional parameters.

Broader Impact Statement

This work focuses exclusively on addressing the technical challenge of domain adaptation for multivariate time series classification. All experiments are conducted using publicly available benchmark datasets, ensuring full transparency and reproducibility. Because the study does not involve any human participants, private data, or personally identifiable information, it poses no ethical, privacy, or societal risks.

GenAI Usage Disclosure

In this manuscript, generative AI tool is used to edit and improve the quality of the text, including checking the spelling, grammar, punctuation and clarity.

References

Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*, 2020.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3, 2013.

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

Larissa E Back and Christopher S Bretherton. The relationship between wind speed and precipitation in the pacific itcz. *Journal of climate*, 18(20):4317–4328, 2005.

Piotr Baranowski, Jaromir Krzyszczak, Cezary Slawinski, Holger Hoffmann, Jerzy Kozyra, Anna Nieróbca, Krzysztof Siwek, and Andrzej Gluza. Multifractal analysis of meteorological time series to assess climate impacts. *Climate Research*, 65:39–52, 2015.

Ananda Basu, Chiara Dalla Man, Rita Basu, Gianna Toffolo, Claudio Cobelli, and Robert A Rizza. Effects of type 2 diabetes on insulin secretion, insulin action, glucose effectiveness, and postprandial glucose metabolism. *Diabetes care*, 32(5):866–872, 2009.

Michal Belda, Eva Holtanová, Tomáš Halenka, and Jaroslava Kalvová. Climate classification revisited: from köppen to trewartha. *Climate research*, 59(1):1–13, 2014.

James Lopez Bernal, Steven Cummins, and Antonio Gasparrini. Interrupted time series regression for the evaluation of public health interventions: a tutorial. *International journal of epidemiology*, 46(1):348–355, 2017.

Tim Bollerslev. Modelling the coherence in short-run nominal exchange rates: a multivariate generalized arch model. *The review of economics and statistics*, pp. 498–505, 1990.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Ruichu Cai, Jiawei Chen, Zijian Li, Wei Chen, Keli Zhang, Junjian Ye, Zhuozhang Li, Xiaoyan Yang, and Zhenjie Zhang. Time series domain adaptation via sparse associative structure alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6859–6867, 2021.

David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S Jensen. Lightts: Lightweight time series classification with adaptive ensemble distillation. *Proceedings of the ACM on Management of Data*, 1(2):1–27, 2023.

Xiwen Chen, Peijie Qiu, Wenhui Zhu, Huayu Li, Hao Wang, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. Timemil: Advancing multivariate time series classification via a time-aware multiple instance learning. *arXiv preprint arXiv:2405.03140*, 2024.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems*, 30, 2017.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Douglas Dockery and Arden Pope. Epidemiology of acute health effects: summary of time-series studies. 1996.

David W Draper and David G Long. Simultaneous wind and rain retrieval using seawinds data. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7):1411–1423, 2004.

Jin Fan, Ke Zhang, Yipan Huang, Yifei Zhu, and Baiping Chen. Parallel spatio-temporal attention-based tcn for multivariate time series prediction. *Neural Computing and Applications*, 35(18):13109–13118, 2023.

Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3575–3584, 2019.

Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33:22092–22103, 2020.

Johannes Fütterer, Maksymilian Kochanski, and Dirk Müller. Application of selected supervised learning methods for time series classification in building automation and control systems. *Energy Procedia*, 122: 943–948, 2017.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 597–613. Springer, 2016.

Michael Ghil, Matthias R Allen, Michael D Dettinger, Kayo Ide, Dummerle Kondrashov, Michael E Mann, Andrew W Robertson, Amira Saunders, Yudong Tian, Ferenc Varadi, et al. Advanced spectral methods for climatic time series. *Reviews of geophysics*, 40(1):3–1, 2002.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. Domain adaptation for time series under feature and label shifts. In *International Conference on Machine Learning*, pp. 12746–12774. PMLR, 2023.

Yifei He, Haoxiang Wang, Bo Li, and Han Zhao. Gradual domain adaptation: Theory and algorithms. *Journal of Machine Learning Research*, 25(361):1–40, 2024.

Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, and Yonghuai Liu. Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24:16453–16482, 2020.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pp. 1989–1998. Pmlr, 2018.

Maximilian Hoffmann, Leander Kotzur, Detlef Stolten, and Martin Robinius. A review on time series aggregation methods for energy system models. *Energies*, 13(3):641, 2020.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Tsung-Yu Hsieh, Suhang Wang, Yiwei Sun, and Vasant Honavar. Explainable multivariate time series classification: a deep neural network which learns to attend to important variables as well as time intervals. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 607–615, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *arXiv preprint arXiv:2403.04468*, 2024.

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

Shima Khoshraftar and Aijun An. A survey on graph representation learning methods. *ACM Transactions on Intelligent Systems and Technology*, 15(1):1–55, 2024.

Sungil Kim and Michael E Baginski. A cross correlation-based stock forecasting model. In *Proceedings of The National Conference On Undergraduate Research*, 2016.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021.

Blake LeBaron, W Brian Arthur, and Richard Palmer. Time series properties of an artificial stock market. *Journal of Economic Dynamics and control*, 23(9-10):1487–1516, 1999.

Kong Aik Lee, Qiongqiong Wang, and Takafumi Koshinaka. The coral+ algorithm for unsupervised domain adaptation of plda. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5821–5825. IEEE, 2019.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Rongjin Li, Weibin Zhang, and Dongpeng Chen. The coral++ algorithm for unsupervised domain adaptation of speaker recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7172–7176. IEEE, 2022a.

Zijian Li, Ruichu Cai, Jiawei Chen, Yuguan Yan, Wei Chen, Keli Zhang, and Junjian Ye. Time-series domain adaptation via sparse associative structure alignment: learning invariance and variance. *arXiv preprint arXiv:2205.03554*, 2022b.

Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021.

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

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2507–2516, 2019.

Bhavitvya Malik, Abhinav Ramesh Kashyap, Min-Yen Kan, and Soujanya Poria. Udapter–efficient domain adaptation using adapters. *arXiv preprint arXiv:2302.03194*, 2023.

Patrick E McKnight and Julius Najab. Mann-whitney u test. *The Corsini encyclopedia of psychology*, pp. 1–1, 2010.

Prapanna Mondal, Labani Shit, and Saptarsi Goswami. Study of effectiveness of time series modeling (arima) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2): 13, 2014.

Mohammadreza MontazeriShatoori, Logan Davidson, Gurdip Kaur, and Arash Habibi Lashkari. Detection of doh tunnels using time-series classification of encrypted traffic. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 63–70. IEEE, 2020.

Yilmazcan Ozyurt, Stefan Feuerriegel, and Ce Zhang. Contrastive learning for unsupervised domain adaptation of time series. *arXiv preprint arXiv:2206.06243*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*, 2020.

Shahbaz Rezaei and Xin Liu. Deep learning for encrypted traffic classification: An overview. *IEEE communications magazine*, 57(5):76–81, 2019.

Tej Bahadur Shahi, Ashish Shrestha, Arjun Neupane, and William Guo. Stock price forecasting with deep learning: A comparative study. *Mathematics*, 8(9):1441, 2020.

Yingxia Shao, Hongzheng Li, Xizhi Gu, Hongbo Yin, Yawen Li, Xupeng Miao, Wentao Zhang, Bin Cui, and Lei Chen. Distributed graph neural network training: A survey. *ACM Computing Surveys*, 56(8):1–39, 2024.

Kartik Sharma, Yeon-Chang Lee, Sivagami Nambi, Aditya Salian, Shlok Shah, Sang-Wook Kim, and Srijan Kumar. A survey of graph neural networks for social recommender systems. *ACM Computing Surveys*, 56 (10):1–34, 2024.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pp. 127–140, 2015.

Baochen Sun and Kate Saenko. 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.

Gian Antonio Susto, Angelo Cenedese, and Matteo Terzi. Time-series classification methods: Review and applications to power systems data. *Big data application in power systems*, pp. 179–220, 2018.

Giota Touloumi, Richard Atkinson, Alain Le Tertre, Evangelia Samoli, Joel Schwartz, Christian Schindler, Judith M Vonk, Giuseppe Rossi, Marc Saez, Daniel Rabszenko, et al. Analysis of health outcome time series data in epidemiological studies. *Environmetrics: The official journal of the International Environmetrics Society*, 15(2):101–117, 2004.

Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4068–4076, 2015.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5495–5504, 2018.

Ly Vu, Hoang V Thuy, Quang Uy Nguyen, Tran N Ngoc, Diep N Nguyen, Dinh Thai Hoang, and Eryk Dutkiewicz. Time series analysis for encrypted traffic classification: A deep learning approach. In *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 121–126. IEEE, 2018.

Xinlei Wang, Xiaoqin Zhao, Ranran Zhou, Yunjuan Gu, Xiaohui Zhu, Zhuqi Tang, Xinlu Yuan, Wei Chen, Rongping Zhang, Chen Qian, et al. Delay in glucose peak time during the oral glucose tolerance test as an indicator of insulin resistance and insulin secretion in type 2 diabetes patients. *Journal of diabetes investigation*, 9(6):1288–1295, 2018.

Yucheng Wang, Yuecong Xu, Jianfei Yang, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Sea++: Multi-graph-based higher-order sensor alignment for multivariate time-series unsupervised domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 2024a.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. 2024b.

Gary Weiss. WISDM Smartphone and Smartwatch Activity and Biometrics Dataset , 2019.

David E Weissman, Mark A Bourassa, and Jeffrey Tongue. Effects of rain rate and wind magnitude on seawinds scatterometer wind speed errors. *Journal of Atmospheric and Oceanic Technology*, 19(5):738–746, 2002.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1768–1778, 2020.

Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. Calda: Improving multi-source time series domain adaptation with contrastive adversarial learning. *IEEE transactions on pattern analysis and machine intelligence*, 2023.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Qiao Xiao, Boqian Wu, Yu Zhang, Shiwei Liu, Mykola Pechenizkiy, Elena Mocanu, and Decebal Constantin Mocanu. Dynamic sparse network for time series classification: Learning what to "see". *Advances in Neural Information Processing Systems*, 35:16849–16862, 2022.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*, 2023.

Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2272–2281, 2017.

Hongliang Yan, Zhetao Li, Qilong Wang, Peihua Li, Yong Xu, and Wangmeng Zuo. Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation. *IEEE Transactions on Multimedia*, 22 (9):2420–2433, 2019.

Tae-Woong Yoo and Il-Seok Oh. Time series forecasting of agricultural products' sales volumes based on seasonal long short-term memory. *Applied sciences*, 10(22):8169, 2020.

Scott L Zeger, Rafael Irizarry, and Roger D Peng. On time series analysis of public health and biomedical data. *Annu. Rev. Public Health*, 27(1):57–79, 2006.

Huanyu Zhang, Yi-Fan Zhang, Zhang Zhang, Qingsong Wen, and Liang Wang. Logora: Local-global representation alignment for robust time series classification. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International conference on machine learning*, pp. 819–827. Pmlr, 2013.

Rongsheng Zhang, Yinhe Zheng, Xiaoxi Mao, and Minlie Huang. Unsupervised domain adaptation with adapter. *arXiv preprint arXiv:2111.00667*, 2021.

Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

Wen Zhang and Dongrui Wu. Discriminative joint probability maximum mean discrepancy (djp-mmd) for domain adaptation. In *2020 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Yue Zhang, Shun Miao, Tommaso Mansi, and Rui Liao. Task driven generative modeling for unsupervised domain adaptation: Application to x-ray image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pp. 599–607. Springer, 2018.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Cheng Zhao, Ping Hu, Xiaohui Liu, Xuefeng Lan, and Haiming Zhang. Stock market analysis using time series relational models for stock price prediction. *Mathematics*, 11(5):1130, 2023.

Junbao Zhuo, Shuhui Wang, Weigang Zhang, and Qingming Huang. Deep unsupervised convolutional domain adaptation. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 261–269, 2017.

A Empirical Evidence of Correlation Shift

In this section, we conduct an empirical analysis on the Human Activity Recognition (HAR) dataset (Anguita et al., 2013) to demonstrate the prevalence of correlation shift. The HAR dataset consists of 30 domains. Then we iterate through the 30 domains in HAR, treating each domain as the source domain while considering the remaining 29 domains as target domains. Our objective is to examine whether multivariate correlations differ significantly between samples from the source and target domains.

For each sample from either the source or the target domain, we compute the multivariate correlation matrix, which is an $D \times D$ high-dimensional structure with D being the number of variables. However, directly analyzing such high-dimensional matrices is challenging. Hence, we calculate the element-wise mean of the correlation matrices for each domain. Mathematically, if the mean values for the correlation matrices from the source and target domains come from different distributions, it implies a significant difference in the distribution of the overall correlations.

To formally test this, we set the null hypothesis $\mathcal{H}_0$: *the mean distributions of the source and target domains originate from the same underlying distribution.* We then apply the Mann-Whitney U test (McKnight & Najab, 2010) to verify $\mathcal{H}_0$. If the p-value is less than 0.05, it indicates a statistical confidence of over 95% in rejecting $\mathcal{H}_0$. This rejection implies that the mean values are from different distributions, confirming the statistical significance of the correlation shift between the source and target domains.

Finally, for every source domain, we calculate the rate of target domains with a significant correlation shift among the rest 29 domains. The experiment result is provided in Figure 1. The x-axis represents the source domain ID, and the y-axis values of the orange bars indicate the rate of target domains with significant correlation shifts. The red dashed line in Figure 1 marks the average rate of correlation shift, which is 78%. These findings provide clear evidence that correlation shifts are prevalent in multivariate time series datasets, highlighting the need to address such shifts.

B Empirical Comparison Between TDC and Linear Matrices

To assess the representative learning ability between TDC and linear matrices in adapters, we compare the performance of these two adpaters on HAR dataset with the domain 1 being the source domain and domain 10 being the target domain. Furthermore, we leverege TimesNet as the backbone, and set both the time length T and the hidden dimension D as 128 to make their parameters compatible. Then after pretraining the backbone on the source domain, we only train these adapters on the target domain with accessible labels. We use Adam optimizer with a learning rate of 1e-4 during training.

C Theoretical Analysis of Correlation Shift

C.1 Proof of Correlation Alignment

Proposition 2 (Correlation Alignment)**.** *Suppose source data $\mathbf{X}_s \in \mathbb{R}^{D\times T}$ and target data $\mathbf{X}_t \in \mathbb{R}^{D\times T}$ follow the source distribution $\mathcal{P}_s$ and the target distribution $\mathcal{P}_t$, respectively. There exists a reweighting matrix $\mathbf{A} \in \mathbb{R}^{D\times D}$, such that the correlation of the source distribution and target distribution can be perfectly aligned, formally expressed as:*

$$\mathrm{Corr}(\mathbf{X}_s) = \mathrm{Corr}(\mathbf{A}\mathbf{X}_t) \tag{4}$$

Proof. Let $\mathbf{\Sigma}_t = \mathbb{E}_{\mathbf{X}_t\sim\mathcal{P}_t}\left[(\mathbf{X}_t - \mathbb{E}\mathbf{X}_t)(\mathbf{X}_t - \mathbb{E}\mathbf{X}_t)^T\right]$ and $\hat{\mathbf{\Sigma}}_t = \mathbb{E}_{\mathbf{Y}}\left[(\mathbf{Y} - \mathbb{E}\mathbf{Y})(\mathbf{Y} - \mathbb{E}\mathbf{Y})^T\right]$ where $\mathbf{Y} = \mathbf{A}\mathbf{X}_t$. Then based on the spectral theorem, we can decompose the covariance matrices $\mathbf{\Sigma}_s$ and $\hat{\mathbf{\Sigma}}_t$:

$$\mathbf{\Sigma}_s = \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^T \tag{7}$$

$$\mathbf{\Sigma}_t = \mathbf{U}_t\mathbf{\Lambda}_t\mathbf{U}_t^T \tag{8}$$

Then since $\mathbf{Y} = \mathbf{A}\mathbf{X}$, the covariance matrix $\hat{\mathbf{\Sigma}}_t$ could be expressed as

$$\begin{aligned}\hat{\mathbf{\Sigma}}_t &= \mathbf{A}\mathbf{\Sigma}_t\mathbf{A}^T \\ &= \mathbf{A}\mathbf{U}_t\mathbf{\Lambda}_t\mathbf{U}_t^T\mathbf{A}^T\end{aligned} \tag{9}$$

Let $\mathbf{A} = \mathbf{U}_s\Lambda_s^{\frac{1}{2}}\mathbf{\Lambda}_t^{-\frac{1}{2}}\mathbf{U}_t^T$. Then we have

$$\begin{aligned}\hat{\mathbf{\Sigma}}_t &= \mathbf{A}\mathbf{U}_t\mathbf{\Lambda}_t\mathbf{U}_t^T\mathbf{A}^T \\ &= \mathbf{U}_s\mathbf{\Lambda}_s^{\frac{1}{2}}\mathbf{\Lambda}_t^{-\frac{1}{2}}\mathbf{U}_t^T\mathbf{U}_t\mathbf{\Lambda}_t\mathbf{U}_t^T\mathbf{U}_t\mathbf{\Lambda}_t^{-\frac{1}{2}}\mathbf{\Lambda}_s^{\frac{1}{2}}\mathbf{U}_s^T \\ &= \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^T \\ &= \mathbf{\Sigma}_s\end{aligned} \tag{10}$$

Therefore, with the reweighting matrix, the covariance matrix of $\mathbf{Y}$ on the target domain could equal to the covariance matrix of $\mathbf{X}_s$ on the source domain. Then, since the correlation matrix is only defined by the covariance matrix as shown in Eq. equation 3, the correlation matrices for $\mathbf{Y}$ on the target domain and $\mathbf{X}_s$ on the source domain are totally the same. □

C.2 Proof of Gaussian Probability Alignment.

Proposition 1 (Gaussian Probability Alignment)**.** *Suppose source data $\mathbf{X}_s \in \mathbb{R}^{D\times T}$ and target data $\mathbf{X}_t \in \mathbb{R}^{D\times T}$ follow $\mathcal{N}_s(\boldsymbol{\mu}_s, \mathbf{\Sigma}_s)$ and $\mathcal{N}_t(\boldsymbol{\mu}_t, \mathbf{\Sigma}_t)$, respectively. There exists a reweighting matrix $\mathbf{A} \in \mathbb{R}^{D\times D}$ and a bias vector $\mathbf{b} \in \mathbb{R}^D$, such that the multivariate joint probability of the reweighted target domain perfectly aligns with that of the source domain, that is for every $i, j = 1, 2, ..., D$*

$$\Pr\left(\mathbf{X}_s[i], \mathbf{X}_s[j]\right) = \Pr\left(\mathbf{Y}[i], \mathbf{Y}[j]\right)$$

where $\mathbf{Y} = \mathbf{A}\mathbf{X}_t + \mathbf{b}$ and $\mathbf{b} = \mathbf{0}$ for most MTS data.

Proof. Based on Proposition 2, it is obvious that $\mathbf{\Sigma}_s = \hat{\mathbf{\Sigma}}_t$ where $\hat{\mathbf{\Sigma}}_t$ is the covariance matrix of $\mathbf{Y}$. Then the mean of $\mathbf{Y}$ could be expressed as

$$\mathbb{E}[\mathbf{Y}] = \mathbf{A}\mathbb{E}[\mathbf{X}] + \mathbf{b} \tag{11}$$

Therefore, as long as $\mathbf{b} = (\mathbf{I} - \mathbf{A})\mathbb{E}\mathbf{X}$, we will have $\mathbb{E}\mathbf{Y} = \mathbb{E}\mathbf{X}$. In the real world, it is quite common to normalize the MTS data during data preprocessing (Liu et al., 2023; Wu et al., 2022; Wang et al., 2024b). Under this circumstance, the expectation of normalized data would be zero, and hence $\mathbf{b} = \mathbf{0}$.

Finally, since the covariances and means of two Gaussian distribution is the same, then these two distribution are also the same. Therefore, we have $\Pr(\mathbf{X}_s[i], \mathbf{X}_s[j]) = \Pr(\mathbf{Y}[i], \mathbf{Y}[j])$. □

C.3 Proof of Attention Approximation

Theorem 1 (Attention Approximation)**.** *The optimal reweighting matrix $\mathbf{A}$ in Proposition 2 can be approximated by an attention matrix $\tilde{\mathbf{A}}$ with an arbitrary precision, where $\tilde{\mathbf{A}}$ is generated by a one-layer Graph Attention Network with an infinite hidden dimension.*

Proof. Our proof proceeds as follows: First, we leverage the universal approximation theorem to establish that an infinitely wide Multi-Layer Perceptron (MLP) can effectively approximate the reweighting matrix $\mathbf{A}$. Then, we demonstrate that a one-layer GAT possesses the same learning capacity as an infinite-width MLP under appropriate conditions. This equivalence allows us to conclude that a one-layer GAT can effectively learn and approximate the matrix $\mathbf{A}$, thereby completing the proof.

Step 1. Revisit the Universal Approximation Theorem. First, the Universal Approximation Theorem (Leshno et al., 1993; Hornik, 1991; Hornik et al., 1989) states:

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function defined on a compact subset, and let $\sigma(\cdot)$ be a nonlinear, measurable activation function. For any $\epsilon > 0$, there exists a MLP $F(t;\theta)$ with a single hidden layer of infinite width, such that:

$$\sup_{t\in\mathcal{T}} |f(t) - F(t;\theta)| < \epsilon \tag{12}$$

where $\mathcal{T}$ is the compact input space, and θ represents the network parameters.

Step2. Approximate the reweighting matrix by an MLP. Now, let us consider a continuous, differentiable function $f : \mathbb{R} \to \mathbb{R}$ such that $f(i \cdot D + j) = \mathbf{A}[i, j]$, where i, j are indices of the reweighting matrix $\mathbf{A}$ and D is the matrix dimension. Based on the Universal Approximation Theorem, we can approximate $f(\cdot)$ using an MLP with a single hidden layer of dimension M, when M is quite large:

$$f(t) \approx F(t) = \mathbf{w}_2^T \sigma(\mathbf{w}_1 t) \tag{13}$$

where $\mathbf{w}_1 \in \mathbb{R}^M$ and $\mathbf{w}_1 \in \mathbb{R}^M$ are two learnable parameters. By stacking the approximated elements, we can have a reconstructed matrix $\mathbf{A}_{mlp}[i, j] = F(i \cdot D + j) \approx \mathbf{A}[i, j]$. Therefore, the reweight output $\mathbf{Y}_{mlp}$ in Proposition 2 could be expressed as:

$$\begin{aligned}\mathbf{Y}_{mlp}[i] &= (\mathbf{A}_{mlp}\mathbf{X})[i] \\ &= \sum_{d=1}^{D} \mathbf{A}_{mlp}[i, d]\mathbf{X}[d] \\ &= \sum_{d=1}^{D} \mathbf{w}_2^T \sigma(\mathbf{w}_1 t_{i,d})\mathbf{X}[d]\end{aligned} \tag{14}$$

where $t_{i,d}$ is the the input for the row index i and column index d.

Step3. Relating an MLP with a GAT. Then we aim to show that a GAT layer has the same learning ability as an MLP. Let $\mathbf{X}$ and $\mathbf{Y}_{gat}$ represent the input and output of the GAT layer, respectively. To align with the notations used in the original GAT paper (Veličković et al., 2017), we denote $\mathbf{x}_i$ as the i-th row of $\mathbf{X}$, i.e., $\mathbf{x}_i = \mathbf{X}[i]$. Then, The formula of a GAT layer could be expressed as follows:

$$\mathbf{Y}_{gat}[i] = \sum_j \alpha_{i,j}\mathbf{W}\mathbf{x}_j, \tag{15a}$$

$$\alpha_{i,j} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}\mathbf{x}_i || \mathbf{W}\mathbf{x}_j\right]\right)\right)}{\sum_k \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}\mathbf{x}_i || \mathbf{W}\mathbf{x}_k\right]\right)\right)} \tag{15b}$$

where $\mathbf{W} \in \mathbb{R}^{M\times M}$ and $\mathbf{a} \in \mathbb{R}^M$ with M being the input dimension. Here, $||$ represents the concatenation operation, and $||_k$ represents the concatenation operation over all the possible element with the subscripts k.

By defining a special nonlinear, measurable activation function $\tilde{\sigma}([\mathbf{x}_i||\mathbf{x}_j\|_{k\neq i,j}\mathbf{x}_k]) = \frac{\exp(\text{LeakyReLU}([\mathbf{x}_i||\mathbf{x}_j]))}{\sum_k \exp(\text{LeakyReLU}([\mathbf{x}_i||\mathbf{x}_k]))}$, we could further simplified the attention coefficient into the following expression:

$$\begin{aligned}\alpha_{i,j} &= \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}\mathbf{x}_i || \mathbf{W}\mathbf{x}_j\right]\right)\right)}{\sum_k \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}\mathbf{x}_i || \mathbf{W}\mathbf{x}_k\right]\right)\right)} \\ &= \tilde{\sigma}\left(\left[\mathbf{a}^T\mathbf{W}\mathbf{x}_i || \mathbf{a}^T\mathbf{W}\mathbf{x}_j\|_{k\neq i,j}\mathbf{a}^T\mathbf{W}\mathbf{x}_k\right]\right) \\ &= \tilde{\sigma}\left(\mathbf{a}^T\mathbf{W}\left[\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_1 \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_D\right]\right) \text{ (where } k \neq i, j) \\ &= \tilde{\sigma}\left(\tilde{\mathbf{a}}^T\mathbf{T}_{i,j}\right)\end{aligned} \tag{16}$$

where $[\mathbf{x}_i, \mathbf{x}_j]$ represents the matrix stacked by the vector $\mathbf{x}_i$ and $\mathbf{x}_j$ along a new dimension. Here, $\tilde{\mathbf{a}} = \mathbf{W}^T\mathbf{a} \in \mathbb{R}^M$ and $\mathbf{T}_{i,j} = [\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_1 \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_D] \in \mathbb{R}^{M\times D}$. Hence, the output $\mathbf{Y}_{gat}$ could be simplified as:

$$\begin{aligned}\mathbf{Y}_{gat}[i] &= \sum_{j=1}^{D} \tilde{\sigma}\left(\tilde{\mathbf{a}}^T\mathbf{T}_{i,j}\right)\mathbf{W}\mathbf{x}_j \\ &= \sum_{d=1}^{D} \tilde{\sigma}\left(\tilde{\mathbf{a}}^T\mathbf{T}_{i,d}\right)\mathbf{W}\mathbf{X}[d]\end{aligned} \tag{17}$$

Obviously, Eq. equation 17 shares a strong similarity with Eq. equation 14. Intuitively, this similarity suggests that a GAT layer could exhibit learning behavior analogous to that of an MLP, making it capable of approximating the linear reweighting matrix $\mathbf{A}$. It would be totally fine to stop here and conclude the proof, but we can still go a step further to rigorously show that the outputs of these two neural networks are element-wise equivalent.

Step 4. Element-wise comparison between the outputs. First, let us expand the expression of $\mathbf{Y}_{mlp}$ as follows:

$$\begin{aligned}\mathbf{Y}_{mlp} &= \sum_{d=1}^{D} \mathbf{w}_2^T \sigma(\mathbf{w}_1 t_{i,d}) \mathbf{X}[d] \\ &= \sum_{d=1}^{D} \sum_{m=1}^{M} \mathbf{w}_2[m] \sigma\left(\mathbf{w}_1[m] t_{i,d}\right) \mathbf{X}[d]\end{aligned} \tag{18}$$

Similarly, we can also expand the expression of $\mathbf{Y}_{gat}$ as follows:

$$\begin{aligned}\mathbf{Y}_{gat} &= \sum_{d=1}^{D} \tilde{\sigma}\left(\tilde{\mathbf{a}}^T \mathbf{T}_{i,d}\right) \mathbf{W}\mathbf{X}[d] \\ &= \sum_{d=1}^{D} \mathbf{W} \tilde{\sigma}\left(\sum_{m=1}^{M} \tilde{\mathbf{a}}[m] \mathbf{T}_{i,j}[m]\right) \mathbf{X}[d]\end{aligned} \tag{19}$$

It is evident that the outputs of the GAT and MLP are essentially identical. The only difference lies in the order of summation and activation functions in Eq. equation 18 and Eq. equation 19. However, since the Universal Approximation Theorem imposes no constraints on the order of summation and activation functions, GAT also adheres perfectly to the universal approximation theorem, enabling it to approximate the reweighting matrix $\mathbf{A}$ with arbitrary precision.

□

C.4 Proof of CATS Approximation

Theorem 2 (CATS Approximation)**.** *Given a linear CATS module with an infinite hidden dimension, i.e.,*

$$\phi(\mathbf{X}) = \mathbf{X} + \mathrm{TDC}_{\uparrow}\left(\mathrm{GAT}\left(\mathrm{TDC}_{\downarrow}\left(\mathbf{X}\right)\right)\right), \tag{20}$$

then $\phi(\mathbf{X})$ could approximate $\mathbf{Y} = \mathbf{A}\mathbf{X}$ in Propositions 2 with an arbitrary precision.

Proof. When using a depthwise convolution with a stride of 1, zero-padding, and a convolution kernel where only the first element is 1 while all others are 0, the convolution operation automatically degenerates into the identity mapping $f(x) = x$. Under this circumstance, Eq. equation 20 would be further simplified as

$$\phi(\mathbf{X}) = \mathbf{X} + \mathrm{GAT}(\mathbf{X}) \tag{21}$$

Then, since Theorem 1 has no requirement on the approximated matrix $\mathbf{X}$, we could leverage a GAT layer with a infinite hidden dimension to approximate the matrix $\mathbf{A} - \mathbf{I}$ with an arbitrary precision. Therefore, the formula of CATS would be expressed as

$$\begin{aligned}\phi(\mathbf{X}) &= \mathbf{X} + \mathrm{GAT}(\mathbf{X}) \\ &\approx \mathbf{X} + (\mathbf{A} - \mathbf{I})\mathbf{X} = \mathbf{A}\mathbf{X}\end{aligned} \tag{22}$$

□

D Training Objectives

In this section, we aim to propose an effective unsupervised training strategy for CATS. Our training strategy is designed to meet three critical objectives: (1) enable the model to effectively extract features from target domain samples; (2) maintain strong classification capabilities; and (3) align feature distributions between the source and target domains. To meet these objectives, we introduce three distinct loss functions that collectively guide the effective training of CATS. The third objective has already been proposed in Section 4.2. Therefore, we only address the first two objectives here.

First, to improve the model's understanding of the target domain, prior UDA works (He et al., 2023; Ghifary et al., 2016; Zhuo et al., 2017) often rely on reconstruction loss, which serves as an additional supervision for the classification on unlabeled target domain. Reconstruction loss ensures that the decoded output closely resembles the input on the target domain, requiring the encoded features to preserve all information from the target domain. However, the usage of reconstruction loss could be harmful to MTS classification. In MTS classification tasks, temporal properties, such as periodicity and trends, are more strongly correlated with the labels whereas local noise may be detrimental to classification performance. But reconstruction loss, by design, does not differentiate between meaningful features and noise, making it a suboptimal choice for such tasks. To address this issue, we propose the use of forecasting loss as an alternative. Forecasting tasks inherently require the model to focus more on temporal properties like trends and periodicity, while ignoring local random fluctuations. Consequently, features extracted for forecasting naturally transfer well to the classification task.

However, directly applying the forecasting loss presents challenges since the forecasting task requires the time series data to be sliced into adjacent historical and future segments. To address this, given a sample $\mathbf{X}_i^t \in \mathbb{R}^{D\times T}$ from the target domain, we first slice the samples into overlapping time windows $\mathcal{W}_{i,k}^t = \mathbf{X}_i^t[:, k : k+L], k \in \{1, \ldots, T-L\}$ [3] with L being the window length, and then use those sliced time windows as the training inputs. Specifically, given a model integrated with CATS, we leverage all the blocks without the last classification head as the feature extractor $f_{\texttt{CATS}}(\cdot)$ and introduce a new forecasting projection head $g_{\texttt{f}}(\cdot)$. Then the forecasting loss could be represented as:

$$\mathcal{L}_{\texttt{f}} = \sum_{i=1}^{n_t} \sum_{k=1}^{T-2L} \text{MAE}(g_{\texttt{f}}(f_{\texttt{CATS}}(\mathcal{W}_{i,k}^t)), \mathcal{W}_{i,k+L}^t) \tag{23}$$

where MAE represent the mean absolute error. Here, $\mathcal{W}_{i,k}$ and $\mathcal{W}_{i,k+L}$ are two adjacent time windows, indicating the history information and future, respectively.

Second, to ensure that CATS maintains the classification capabilities of the pretrained model, we perform a classification task using the labeled data from the source domain. To make the temporal dimension align with the previous forecasting task, we calculate the cross-entropy loss for each sliced time window:

$$\mathcal{L}_{\texttt{c}} = \sum_{i=1}^{n_s} \sum_{k=1}^{T-L} \ell_{\texttt{CE}}(g_{\texttt{c}}(f_{\texttt{CATS}}(\mathcal{W}_{i,k}^s)), y_i^s) \tag{24}$$

where $\ell_{\texttt{CE}}$ is the cross-entropy loss function, $g_{\texttt{c}}$ is the classification head, and $\mathcal{W}_{i,k}^s$ is the sliced time window from the source domain. During the inference phase, we randomly sample m time windows from the entire time series and use a majority voting scheme to predict the label for the entire sequence based on the predictions from the sliced windows.

To sum up, we combine the classification loss $\mathcal{L}_{\texttt{c}}$ on the source domain, the forecasting loss $\mathcal{L}_{\texttt{f}}$ on the target domain, and the correlation alignment loss $\mathcal{L}_{\texttt{corr}}$ across these two domains to formulate the final loss function. This unified objective ensures that the model not only learns discriminative features for classification but also captures temporal properties and reduces correlation shift effectively. Mathematically, the loss function can be expressed as:

$$\mathcal{L} = \mathcal{L}_{\texttt{c}} + \lambda_{\texttt{f}}\mathcal{L}_{\texttt{f}} + \lambda_{\texttt{corr}}\mathcal{L}_{\texttt{corr}} \tag{25}$$

where $\lambda_{\texttt{fct}}$ and $\lambda_{\texttt{cor}}$ are two hyperparameters

[3] We use $\mathbf{X}[:, t_1 : t_2]$ to represent a sliced segment from time t_1 to time t_2 of $\mathbf{X}$. All the slicing notations follow Python standards.

E Dataset Description

Table 2: The statistics of datasets.

Dataset	# Domains	# Timestamps	# Variables
HAR	30	128	9
WIDSM	36	128	3
HHAR	9	128	3
Boiler	3	128	20

In this paper, we validate the effectiveness of CATS on four different datasets, HAR (Anguita et al., 2013), WISDM (Weiss, 2019), HHAR (Stisen et al., 2015), and Boiler (Cai et al., 2021). The statistics of datasets are provided in Table 2, and the detailed information is listed below.

- **HAR dataset**. The Human Activity Recognition Dataset has been collected from 30 subjects performing six different activities (Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, Laying). It consists of inertial sensor data that was collected using a smartphone carried by the subjects.
- **WISDM dataset**. WISDM Smartphone and Smartwatch Activity and Biometrics Dataset collects raw accelerometer and gyroscope sensor data from the smartphone and smartwatch at a rate of 20Hz. It is collected from 51 test subjects as they perform 18 activities for 3 minutes apiece.
- **HHAR dataset**. The Heterogeneity Dataset for Human Activity Recognition contains the readings of two motion sensors commonly found in smartphones. Reading were recorded while nine users executed six different activities scripted in no specific order carrying smartwatches and smartphones.
- **Boiler dataset**. The boiler data consists of sensor data from three boilers from 2014/3/24 to 2016/11/30. There are 3 boilers in this dataset and each boiler is considered as one domain. We slice the original time series data with a time window of 128 and a stride of 32.

F Domain Pair Selection

In this study, we utilize four datasets, each containing a large number of domains. As a result, exhaustively evaluating all possible source-target domain pairs is impractical (for example, 900 pairs for the HAR dataset). Therefore, selecting reasonable and effective source-target domain pairs becomes critically important.

To address this, we adopt the following domain pair selection mechanism: For each source-target domain pair, we compute the Wasserstein distance between samples sharing the same label in the source and target domains. We then sum the distances across all possible labels. Mathematically, this distance can be expressed as:

$$d = \sum_{y \in \mathcal{Y}} \text{Wass}(\mathcal{P}_S^y, \mathcal{P}_T^y) \tag{26}$$

where $\mathcal{P}_S^y$ and $\mathcal{P}_T^y$ represent the distributions of samples with label y in the source domain S and target domain T, respectively, and $\text{Wass}(\cdot, \cdot)$ denotes the Wasserstein distance. This distance d quantifies the similarity between the source and target domains: the smaller the distance, the smaller the domain shift, and the lower the difficulty of domain adaptation.

For HAR, HHAR and WISDM datasets, we divide all domain pairs into 10 groups, sorted by increasing the distance d. From each group, we sample one domain pair. This strategy ensures that the selected domain pairs represent varying levels of domain adaptation difficulty, from small to large domain shifts. For the Boiler dataset, due to its quite limited domain pairs (3 domains and 6 domain pairs in total), we only choose the domain pair with the largest d and the smallest d, respectively.

The experimental results, summarized in Table 1, demonstrate the performance of our method across these selected domain pairs. Note that within each dataset, the domain pairs from the top to the bottom in Table

1 are ordered by increasing d, indicating progressively higher domain adaptation difficulty (e.g., in the HAR dataset, the pair $24 \rightarrow 27$ represents the smallest difficulty, while $19 \rightarrow 10$ represents the largest difficulty).

G Description of Baselines

In this paper, we compare CATS with 5 different baselines. These baselines could be roughly divided into three different categories.

First, correlation-related UDA method is

- **CORAL** (Sun & Saenko, 2016) learn a nonlinear transformation that aligns correlations of layer activations in deep neural networks.

Second, MTS-related UDA methods include

- **Raincoat** (He et al., 2023) uses time and frequency-based encoders on the polar coordinate of frequency to learn domain-invariant time series representations.
- **SASA** (Cai et al., 2021) introduces the intra-variables and inter-variables sparse attention mechanisms to extract associative structure time-series data with considering time lags for domain adaptation.
- **CLUDA** (Ozyurt et al., 2022) proposes a contrastive learning framework to learn domain-invariant, contextual representation for UDA of time series data.

Third, we introduce an adapter-related UDA method:

- **UDApter** (Malik et al., 2023) adds a domain adapter to learn domain-invariant information and a task adapter that uses domain-invariant information to learn task representations in the source domain.

H Step-by-step Incremental Adjustment

In the ablation study, we progressively adjusted the vanilla Transformer to the CATS-enhanced Transformer, resulting in a significant improvement in accuracy from 79.79% to 98.98%. Specifically, we introduced the following six incremental adjustments:

1. **+ Adapter (Eq. 1).** We incorporate the adapter defined in Eq. equation 1 into the vanilla Transformer and trained it using the classification loss function $\mathcal{L}_{\mathsf{c}}$ in Eq. 24 on the source domain. This modification results in an accuracy improvement of 1.01%.
2. **+ correlation loss.** We optimize the adapter using a combination of classification loss and correlation alignment loss. This step further enhances accuracy by 9.89%.
3. **Adapter → CATS.** We replace the adapter in Eq. 1 with CATS and train it with the combined classification and correlation alignment loss. This substitution improved accuracy by 2.02%.
4. **+ window slicing.** To align with the setting of forecasting loss, we slice the original samples with a length of 128 into overlapping time windows with a length of 48 and used these sliced windows as inputs to train CATS. This adjustment yields an additional accuracy gain of 2.89%.
5. **forecasting loss.** We introduce the forecasting loss, which uses consecutive time windows as input and their corresponding ground truth for prediction. The final loss function $\mathcal{L}$ in Eq. equation 6 is then leveraged to train CATS, resulting in an accuracy improvement of 0.8%.
6. **+ max voting.** We apply a max-voting method to assign the label of the original sample based on predictions from its sliced time windows. This final step further boosted accuracy by 3.32%.

Table 3: Hyperparameters of backbone models.

Hyperparameter	e_layers	d_model	d_ff	top_k	epoch (pretrain)
Value	3	128	256	3	10

I Implementation Details

We use the code from Time-Series-Library repository [4] to construct three different Transformer variants as backbone models, Transformer, TimesNet, and iTransformer. The hyperparameters for these three models follow the default configuration on Time-Series-Library repository, as shown in Table 3. For CATS, we use the TCNs with a kernel size $r = 5$ and a padding of 2. We use Xavier initialization for the down-project TDC and GAT, and zero initialization for the up-down TDC. For training, we set the length of sliced time windows as 48 and set the number of sampled windows m for max voting as 16. We use Adam optimizer with a learning rate of 1e-4, and set $\lambda_{\mathtt{c}} = 0.5$ and $\lambda_{\mathtt{f}} = 0.5$.

J Comparison Between Correlation Alignment Loss and CORAL Loss

CORAL loss (Sun & Saenko, 2016) is one widely-used domain adaptation loss, which focuses on minimizing the covariance between the source samples and the target samples. In this section, we will demonstrate that the correlation alignment loss offers advantages over the CORAL loss. Furthermore, we show that under certain simplified conditions, the correlation alignment loss can be reduced to the CORAL loss, providing a unified perspective on both approaches.

Our correlation alignment loss aim to use MMD to minimize the mean of the distributions of $\operatorname{corr}(\mathbf{H}^s)$ and $\operatorname{corr}(\mathbf{H}^t)$. Let the distributions of $\operatorname{corr}(\mathbf{H}^s)$ and $\operatorname{corr}(\mathbf{H}^t)$ be denoted as $\mathcal{C}^s$ and $\mathcal{C}^t$, respectively. Mathematically, we aim to optimize the following equation.

$$\begin{aligned}\mathcal{L}_{\mathtt{corr}} &= \mathrm{MMD}(\mathcal{C}^s, \mathcal{C}^t) \\ &= \left\|\mathbb{E}\left[\psi\left(\operatorname{corr}\left(\mathbf{H}^s\right)\right)\right] - \mathbb{E}\left[\psi\left(\operatorname{corr}\left(\mathbf{H}^t\right)\right)\right]\right\|_2\end{aligned} \tag{27}$$

where $\psi(\cdot)$ is one feature mapping function and $\operatorname{corr}(\mathbf{H}) = \operatorname{vec}\left(\frac{(\mathbf{H})(\mathbf{H})^T}{\|\mathbf{H}\|_{\mathcal{F}}^2}\right)$. Here, let us relax this feature mapping function to be the identity function, i.e., $\psi(\mathbf{X}) = \mathbf{X}$. Then our optimization objective could be further deduced:

$$\begin{aligned}\mathcal{L}_{\mathtt{corr}} &= \left\|\mathbb{E}\left[\operatorname{corr}\left(\mathbf{H}^s\right)\right] - \mathbb{E}\left[\operatorname{corr}\left(\mathbf{H}^t\right)\right]\right\|_2 \\ &= \left\|\mathbb{E}\left[\hat{\mathbf{h}}^s(\hat{\mathbf{h}}^s)^T\right] - \mathbb{E}\left[\hat{\mathbf{h}}^t(\hat{\mathbf{h}}^t)^T\right]\right\|_2 \\ &= \Bigg\|\mathbb{E}\left[\left(\hat{\mathbf{h}}^s - \mathbb{E}\left[\hat{\mathbf{h}}^s\right]\right)\left(\hat{\mathbf{h}}^s - \mathbb{E}\left[\hat{\mathbf{h}}^s\right]\right)^T\right] \\ &\quad - \mathbb{E}\left[\left(\hat{\mathbf{h}}^t - \mathbb{E}\left[\hat{\mathbf{h}}^t\right]\right)\left(\hat{\mathbf{h}}^t - \mathbb{E}\left[\hat{\mathbf{h}}^t\right]\right)^T\right] \\ &\quad + \mathbb{E}\left[\hat{\mathbf{h}}^s\left(\hat{\mathbf{h}}^s\right)^T\right] - \mathbb{E}\left[\hat{\mathbf{h}}^t\left(\hat{\mathbf{h}}^t\right)^T\right]\Bigg\|_2\end{aligned} \tag{28}$$

where $\hat{\mathbf{h}}^s$ and $\hat{\mathbf{h}}^t$ are the normalized vector from $\operatorname{vec}(\mathbf{H}^s)$ and $\operatorname{vec}(\mathbf{H}^t)$, i.e., $\hat{\mathbf{h}}^s = \frac{\operatorname{vec}(\mathbf{H}^s)}{\|\operatorname{vec}(\mathbf{H}^s)\|_2}$ and $\hat{\mathbf{h}}^t = \frac{\operatorname{vec}(\mathbf{H}^t)}{\|\operatorname{vec}(\mathbf{H}^t)\|_2}$. Due to the triangle inequality, we have

$$\begin{aligned}&\mathcal{L}_{\mathtt{corr}} \leq \mathcal{L}_{\mathtt{CORAL}} + \mathcal{L}_{\mathtt{mean}}, \\ &\text{where } \mathcal{L}_{\mathtt{CORAL}} = \left\|\mathbb{E}\left[\left(\hat{\mathbf{h}}^s - \mathbb{E}\left[\hat{\mathbf{h}}^s\right]\right)\left(\hat{\mathbf{h}}^s - \mathbb{E}\left[\hat{\mathbf{h}}^s\right]\right)^T\right] - \mathbb{E}\left[\left(\hat{\mathbf{h}}^t - \mathbb{E}\left[\hat{\mathbf{h}}^t\right]\right)\left(\hat{\mathbf{h}}^t - \mathbb{E}\left[\hat{\mathbf{h}}^t\right]\right)^T\right]\right\|_2 \\ &\text{and } \mathcal{L}_{\mathtt{mean}} = \left\|\mathbb{E}\left[\hat{\mathbf{h}}^s\left(\hat{\mathbf{h}}^s\right)^T\right] - \mathbb{E}\left[\hat{\mathbf{h}}^t\left(\hat{\mathbf{h}}^t\right)^T\right]\right\|_2\end{aligned} \tag{29}$$

[4] https://github.com/thuml/Time-Series-Library

Here, $\mathcal{L}_{\texttt{CORAL}}$ represents the original loss proposed by CORAL (Sun & Saenko, 2016), and $\mathcal{L}_{\texttt{mean}}$ minimizes the discrepancy between the mean distributions of the source and target domains. Thus, the correlation alignment loss not only aligns the multivariate correlation between the source and target domains, as CORAL does, but also reduces the mean differences between the two domains.

Compared to CORAL and its following works, the correlation alignment loss simultaneously supervises both covariance and mean alignment, ensuring more precise domain alignment. Notably, when the mean distributions of the source and target domains coincide, the correlation alignment loss naturally reduces to the CORAL loss.

K More Related Works

Graph Neural Networks. GNNs are effective for capturing dependencies within graphs. Graph Convolutional Networks (GCNs) (Zhang et al., 2019; Kipf & Welling, 2016) aggregate neighbor information by utilizing a localized first-order approximation of spectral graph convolutions. Graph Attention Networks (GATs) (Veličković et al., 2017) implement attention mechanisms that dynamically weigh the contributions of neighboring nodes. GRAND (Feng et al., 2020) learns node representations by randomly dropping nodes to augment data and enforcing the consistency of predictions among augmented data. GraphSAGE (Hamilton et al., 2017) generates embeddings for unseen nodes by sampling and aggregating features from the local neighborhood. For more recent works on GNNs, see (Sharma et al., 2024; Ju et al., 2024; Khoshraftar & An, 2024; Shao et al., 2024).

Multivariate Time Series Classification. Several recent works have sought to advance multivariate time series classification (MTSC) by improving interpretability, efficiency, and adaptability. LAXCAT (Hsieh et al., 2021) employs a CNN with dual attention to simultaneously identify the most informative variables and the temporal intervals that drive predictions, yielding both state-of-the-art accuracy and built-in explainability. DSN (Xiao et al., 2022) uses sparse connections learned via dynamic sparse training to cover multiple scales without extensive hyperparameter tuning. TimeMIL (Chen et al., 2024) formulates classification as a time-aware MIL problem and leverages a time-aware pooling mechanism and a wavelet-positional transformer to better localize sparse, anomalous patterns in long series. LightTS (Campos et al., 2023) compresses ensembles into lightweight student models by learning teacher-specific weights and identifying Pareto-optimal trade-offs.

L Limitations

Despite offering a concise and effective solution to the UDA problem in multivariate time series (MTS), our approach still faces two key limitations: (1) CATS is specifically designed under the assumption that there exists a significant correlation shift between domains. If the domain shift is primarily temporal shift, i.e., involving only changes in temporal patterns without notable differences in inter-variable correlations, CATS may offer only limited performance gains. (2) In real-world scenarios, time series data often suffer from missing values or irregular sampling. CATS does not incorporate specialized mechanisms to handle such inconsistencies, which may hinder its effectiveness and limit its applicability in these settings.