

# Source-Free Domain Adaptation with Temporal Imputation for Time Series Data

Mohamed Ragab

Institute for Infocomm Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

Centre for Frontier AI Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

mohamedr002@e.ntu.edu.sg

Emadeldeen Eldele

Nanyang Technological University  
Singapore

Centre for Frontier AI Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

emad0002@ntu.edu.sg

Min Wu

Institute for Infocomm Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

wumin@i2r.a-star.edu.sg

Chuan-Sheng Foo

Institute for Infocomm Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

Centre for Frontier AI Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

foo\_chuan\_sheng@i2r.a-star.edu.sg

Xiaoli Li

Institute for Infocomm Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

Centre for Frontier AI Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

xlli@i2r.a-star.edu.sg

Zhenghua Chen\*

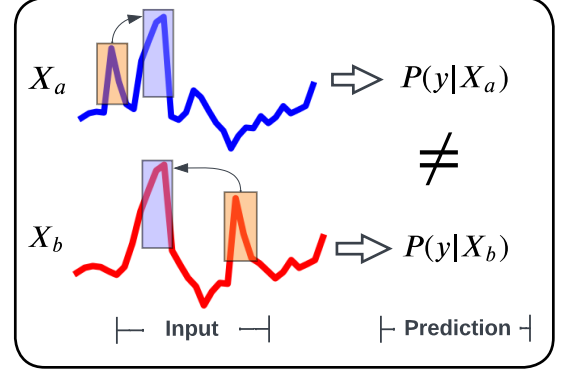
Institute for Infocomm Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

Centre for Frontier AI Research,  
Agency for Science Technology and  
Research (A\*STAR)  
Singapore

chen0832@e.ntu.edu.sg

## ABSTRACT

Source-free domain adaptation (SFDA) aims to adapt a pretrained model from a labeled source domain to an unlabeled target domain without access to the source domain data, preserving source domain privacy. Despite its prevalence in visual applications, SFDA is largely unexplored in time series applications. The existing SFDA methods that are mainly designed for visual applications may fail to handle the temporal dynamics in time series, leading to impaired adaptation performance. To address this challenge, this paper presents a simple yet effective approach for source-free domain adaptation on time series data, namely MAsk and impUte (MAPU). First, to capture temporal information of the source domain, our method performs random masking on the time series signals while leveraging a novel temporal imputer to recover the original signal from a masked version in the embedding space. Second, in the adaptation step, the imputer network is leveraged to guide the target model to produce target features that are temporally consistent with the source features. To this end, our MAPU can explicitly account for temporal dependency during the adaptation while avoiding the imputation in the noisy input space. Our method is the first to handle temporal consistency in SFDA for time series data and can be seamlessly equipped with other existing SFDA methods. Extensive experiments conducted on three real-world time series datasets demonstrate that our MAPU achieves significant performance gain over existing methods. Our code is available at [https://github.com/mohamedr002/MAPU\\_SFDA\\_TS](https://github.com/mohamedr002/MAPU_SFDA_TS).



**Figure 1: How do the temporal relations matter in time series? Despite similarities in the values of the two signals, variations in the temporal position of their observations can result in different predictions.**

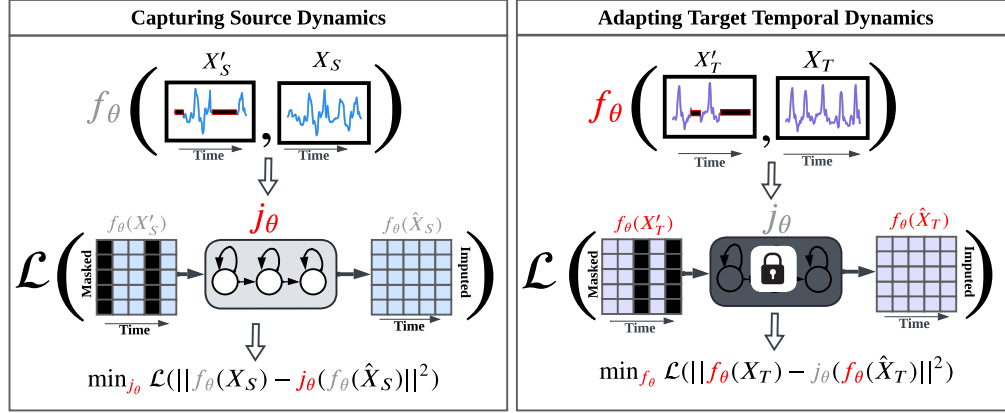
## KEYWORDS

Source-free domain adaptation, time series data, temporal imputation

## 1 INTRODUCTION

Deep learning has achieved impressive performance in numerous time series applications, such as machine health monitoring, human activity recognition, and healthcare. However, this success heavily

\*Corresponding Author



**Figure 2: Adaptation with Temporal Imputation.** Left: A temporal imputer network is trained to predict the full sequence from its masked version to capture the temporal information of the source domain. Right: Once trained, the temporal imputer network guides the target model to produce features that are temporally consistent with the source domain. (Best in viewed in colors. Components in red color are trainable, while those in gray color are non-trainable).

relies on the laborious annotation of large amounts of data. To address this issue, unsupervised domain adaptation (UDA) has gained traction as a way to leverage pre-labeled source data for training on unlabeled target data, while also addressing the distribution shift between the two domains [32]. There is a growing interest in applying UDA to time series data [26], with existing methods seeking to minimize statistical distance across the source and target features [3, 38] or using adversarial training to find domain invariant features [11, 25, 33, 34]. However, these approaches require access to the source data during the adaptation process, which may not be always possible, due to data privacy regulations.

To address this limitation, a more practical setting, i.e., source-free domain adaptation (SFDA), has been proposed, where only a source-pretrained model is available during the adaptation process [12]. In recent years, several SFDA methods have been developed for visual applications [12, 13, 20, 24, 28]. One prevalent paradigm has incorporated some auxiliary tasks to exploit the characteristics of visual data to improve the source-free adaptation [2, 13, 21]. However, all these methods are primarily designed for visual applications and may fail to handle the temporal dynamics of time series data.

In time series data, temporal dependency refers to the interdependence between values at different time points, which has a significant impact on predictions [30]. As demonstrated in Figure 1, even two signals with similar observations can lead to differing predictions if the temporal order is different. Such temporal dynamics make adapting the temporal information between two shifted domains a key challenge in unsupervised domain adaptation. The problem becomes even more challenging under source-free adaptation settings, where no access to the source data is provided during the target adaptation. Therefore, our key question is how to effectively adapt temporal information in time series data in the absence of the source data.

In this work, we address the above challenges and propose a novel SFDA approach, i.e., MAsk and imPUte (MAPU), for time series data. Our method trains an autoregressive model to capture the

temporal information on the source domain, which is then transferred to the target domain for adaptation. The key steps of our approach are illustrated in Figure 2. First, the input signal undergoes temporal masking. Both the masked signal and the original signal are then fed into an encoder network, which generates the corresponding feature representation. Subsequently, the temporal imputation network is trained to impute the original signal from the masked signal in the feature space, enabling smoother optimization for the temporal imputation task. During adaptation, the imputation network is used to guide the target model to generate target features that can be imputed by the source imputation network. Our method is versatile and can be integrated with the existing SFDA methods to provide them with temporal adaptation capability. The main contributions of this work can be summarized as follows:

- To the best of our knowledge, we are the first to achieve the source-free domain adaptation for time series applications.
- We propose a novel temporal imputation task to ensure sequence consistency between the source and target domains.
- We propose a versatile methodology for integrating temporal adaptation capability into existing SFDA methods.
- We conduct extensive experiments and demonstrate that our approach results in a significant improvement in adaptation performance on real-world data, and is particularly effective for time series adaptation tasks.

## 2 RELATED WORK

### 2.1 Time series Domain Adaptation

Several methods have been proposed to address the challenge of distribution shift in time series data. These methods can be broadly categorized into two groups: discrepancy-based methods and adversarial-based methods. Discrepancy-based methods use statistical distances to align the feature representations of the source and target domains. For instance, AdvSKM leverages the maximum mean discrepancy

(MMD) distance in combination with a hybrid spectral kernel to consider temporal dependencies during domain adaptation [22]. Another example is SASA, which learns the association structure of time series data to align the source and target domains [3]. On the contrary, adversarial-based methods use adversarial training to mitigate the distribution shift between the source and target domains. For instance, CoDATS utilizes a gradient reversal layer (GRL) for adversarial training with weak supervision on multi-source human activity recognition data [33]. Furthermore, DA\_ATTN couples adversarial training with an un-shared attention mechanism to preserve the domain-specific information [11]. Recently, SLARDA presents an autoregressive adversarial training approach for aligning temporal dynamics across domains [25].

Albeit promising, the design of these methods is based on the assumption that source data is available during the adaptation step. However, accessing to source data may not be possible in practical situations due to privacy concerns or storage limitations. Differently, our MAPU adapts a model pretrained on source data to new domains without access to source data during adaptation, which can be a more practical solution for high-stake applications.

## 2.2 Source Free Domain Adaptation

Source-Free Domain Adaptation (SFDA) is a new problem-setting, where we do not have access to the source domain data during adaptation. This objective can be achieved in several ways. One approach is to leverage a model pretrained on the source domain to generate synthetic source-like data during the adaptation step [17, 19, 24, 28]. Another approach is to use adversarial training between multiple classifiers to generalize well to the target classes [5, 35]. Another prevalent approach uses softmax scores or their corresponding entropy to prioritize confident samples for pseudo-labeling, assuming that the model should be more confident on source samples and less confident on target samples [12, 13, 20].

Despite the strong potential demonstrated by these methods, they are primarily designed for visual applications and may fail to effectively align temporal dynamics in time series data. In contrast, our method addresses this challenge through a novel temporal imputation task, ensuring temporal consistency between domains during the adaptation.

## 3 METHODOLOGY

### 3.1 Problem definition

Given a labeled source domain  $\mathcal{D}_S = \{X_S^i, y_S^i\}_{i=1}^{n_S}$ , where  $X_S \in \mathcal{X}_S$  can be a uni-variate or multi-variate time series data with a sequence length  $L$ , while  $y_S \in \mathcal{Y}_S$  represents the corresponding labels. In addition, we have an unlabeled target domain  $\mathcal{D}_T = \{X_T^i\}_{i=1}^{n_T}$ , where  $X_T \in \mathcal{X}_T$ , and it also shares the same label space with  $\mathcal{D}_S$ . Following the existing UDA settings, we assume a difference across the marginal distributions, i.e.,  $P(X_S) \neq P(X_T)$ , while the conditional distributions are stable, i.e.,  $P(y_S|X_S) \approx P(y_T|X_T)$ .

This work aims to address the source-free domain adaptation problem, where access to source data is strictly prohibited during the adaptation phase to ensure data privacy. Furthermore, we adopt the vendor-client source-free paradigm [6, 14–16], which allows the influence of the source pretraining stage. This assumption is realistic in use-cases where there is a collaboration between various entities,

but it is not possible to share source data due to data privacy, security, or regulatory issues.

### 3.2 Overview

We present our MAPU to achieve source-free adaptation on time series temporal data while considering the temporal dependencies across domains. The pipeline of the proposed method is illustrated in Figure 4. Given the input signal and a temporally masking signal, our method comprises two stages: (1) training an autoregressive network, referred to as the *imputer network*, which captures the temporal information of the source domain through a novel temporal imputation task, and (2) leveraging the source-pretrained imputer network to guide the target encoder towards producing temporally consistent target features in the adaptation stage. Next, we will first elaborate on the temporal masking procedure before delving into the details of each stage.

### 3.3 Temporal Masking

In this section, we explain our process of temporal masking. We start by dividing the input signal,  $X$ , into several blocks along the time dimension. Then, we randomly choose some of these blocks and set their values to zero, creating a masked version of the signal called  $X'$ . This process is applied to both the source and target domains. Our aim is to challenge the model to use the information from surrounding blocks to fill in the missing parts and capture the temporal dependencies in the input signal. Further discussion on the impact of the masking ratio on the adaptation performance can be found in the experiment section.

### 3.4 Capturing Source Temporal Dynamics

In the pretraining stage, current methods typically map the source data from the input space to the feature space using an encoder network, represented as  $f_\theta : \mathcal{X}_S \rightarrow \mathcal{H}_S$ . The extracted features are then passed through a classifier network,  $g_\theta : \mathcal{H}_S \rightarrow \mathcal{Y}_S$ , to make class predictions for the source data. However, to effectively adapt to other time series domains, it is important to consider the temporal relations in the source domain. Using only cross-entropy for training the source network may neglect this aspect. To address this, we propose a temporal imputation task that aims to recover the input signal from a temporally masked signal in the feature space.

The imputation task is performed by an imputer network  $j_\theta$  that takes the masked signal and maps it to the original signal. The input signal  $X_S$  and masked signal  $X'_S$  are first transformed into their corresponding feature representations  $H_S$  and  $H'_S$  by the encoder  $f_\theta$ . The task of the imputer network is represented as  $\hat{H}'_S = j_\theta(f_\theta(X'_S)) \rightarrow H_S = f_\theta(X_S)$ , where  $\hat{H}'_S$  is the imputed signal. The imputer network is trained to minimize the mean square error between the features of the original signal and the imputed signal, which can be formulated as:

$$\min_{j_\theta} \mathcal{L}_{mapu}^S = \frac{1}{n} \sum_{i=1}^{n_S} \|f_\theta(X_S^i) - j_\theta(f_\theta(X_S'^i))\|_2^2, \quad (1)$$

where  $H_S = f_\theta(X_S^i)$  are the latent features of the original signal,  $\hat{H}'_S = j_\theta(f_\theta(X_S'^i))$  is the output of the imputer network, and  $n_S$  is the total number of source samples.

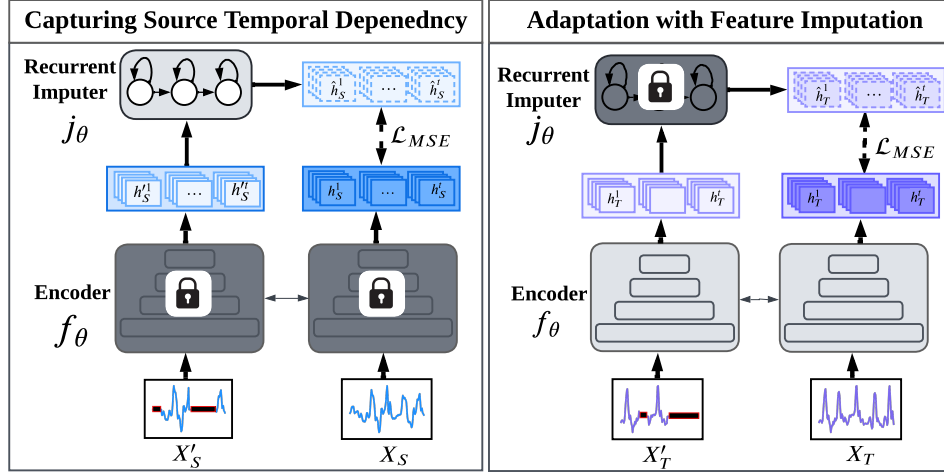


Figure 3: Adaptation with Temporal Imputation for time series data. **Left:** The pretraining stage of the temporal imputer network  $j_\theta$  to capture the temporal dynamics of the source domain. First, we perform random masking across the time dimension of the source signal. Given the original source signal  $X_S$  and its temporally masked signal  $X'_S$ , the encoder network  $f_\theta$  is used to generate the corresponding latent features  $H_S$  and  $H'_S$  respectively. Subsequently,  $j_\theta$  is updated to produce imputed features  $\hat{H}_S$  from masked features  $H'_S$  using the mean square error loss. **Right:** The adaptation stage of the encoder network on the target domain data. The encoder  $f_\theta$  is updated to produce source-like features that are imputable by the pretrained  $j_\theta$ .

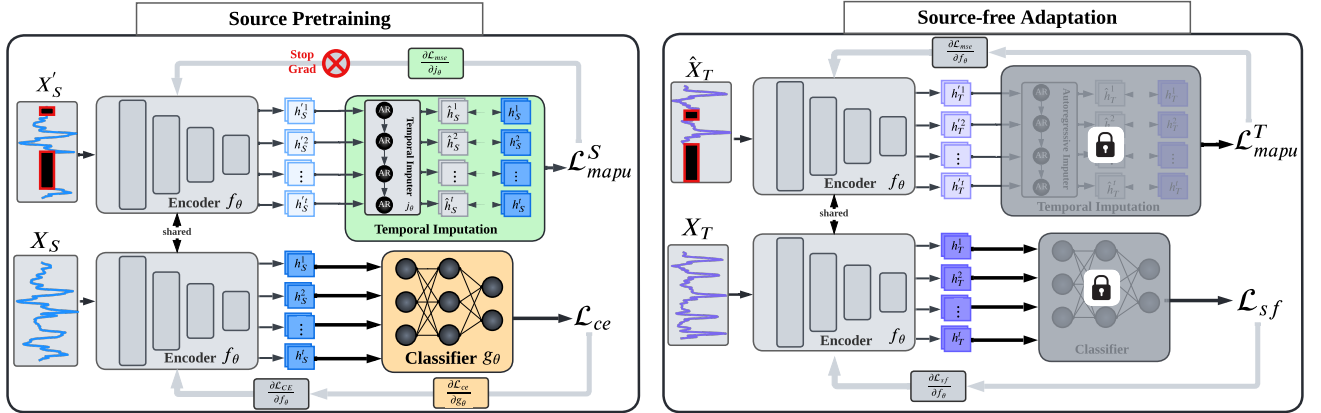


Figure 4: Integrating our temporal imputation with other source-free methods (Best viewed in colors). **Left:** In the pretraining stage, the source model is trained using conventional cross-entropy loss  $\mathcal{L}_{ce}$ , and the temporal imputer network is trained using  $\mathcal{L}_{mapu}^S$  to impute the features of the masked signals and capture the source temporal information on the feature space. **Right:** In the adaptation stage, the target model is jointly trained with both a generic source-free adaptation loss  $\mathcal{L}_{sf}$  and our temporal imputation loss  $\mathcal{L}_{mapu}^T$  to perform the adaptation while ensuring temporal consistency with the source features.

### 3.5 Temporal Adaptation with Feature Imputation

In the adaptation stage, the goal is to train the target encoder network to produce target features temporally consistent with the source features. The target encoder network  $f_\theta$  is used to extract latent feature representations from a target sample  $X_T$  and its masked version  $X'_T$ . The fixed source-pretrained imputer network  $j_\theta$  is then used to reconstruct the features of the original signal from the masked

features. However, due to domain differences, the source imputer may not be able to accurately reconstruct the target features. Thus, the encoder network  $f_\theta$  is updated to produce target features that can be accurately reconstructed by the imputer network. This can be expressed as the following optimization problem:

**Algorithm 1** Adaptation with Temporal Imputation

- 
- 1: **Input:** Target sample  $X_T$ , source pretrained encoder  $f_\theta$ , source classifier  $g_\theta$ , imputer network  $j_\theta$
  - 2: **Initialize:** Construct temporally masked version of  $X_T$ :  $X'_T$
  - 3: **Extract:** Latent feature representations:  $H_T = f_\theta(X_T)$ ,  $H'_T = f_\theta(X'_T)$
  - 4: **Impute:** Masked features using imputer network:  $\hat{H}_T = j_\theta(H'_T)$
  - 5: **Update:** Encoder  $f_\theta$  to produce target features that can be accurately reconstructed by  $j_\theta$  using Equation 2
  - 6: **Output:** Updated encoder  $f_\theta$
- 

$$\min_{f_\theta} \mathcal{L}_{mapu}^T = \frac{1}{n} \sum_{i=1}^{n_T} \|f_\theta(X_T)^i - j_\theta(f_\theta(X'_T)^i)\|_2^2, \quad (2)$$

where  $H_T = f_\theta(X_T)$  are the original target features,  $\hat{H}_T = j_\theta(f_\theta(X'_T))$  are the adapted target features produced by the imputer network to minimize the mean square error loss, and  $n_T$  is the total number of target samples. Notably, only the encoder network is optimized, producing features that can be accurately imputed by the fixed source-pretrained imputer network. To reduce the imputation loss, the adapted target features should be temporally consistent with the source features.

Algorithm 1 illustrates the adaptation procedure via temporal imputation. The process starts by first constructing a temporally masked version of the input target sample represented as  $X'_T$ . Next, the source-pretrained encoder is used to extract the latent features of both the original signal and the temporally masked signal, represented as  $H_T$  and  $H'_T$ , respectively. Finally, the encoder network is updated to make features of the masked signal recoverable by the source-pretrained imputer network, using the mean square error loss in Equation 2.

### 3.6 Integration with Other Source-free Methods

Our proposed MAPU is generic and can be integrated with other source-free adaptation methods. Typically, source-free adaptation involves a two-stage training procedure: (1) pretraining the source model with source domain data, and (2) adapting the pretrained model to the target domain. As shown in Figure 4, our MAPU can be seamlessly integrated into existing SFDA methods in both stages.

In the pretraining stage, MAPU operates in the feature space by training the temporal imputation network,  $j_\theta$ , to capture the temporal information from the source domain. The loss associated with the temporal imputation task does not propagate to the encoder model,  $f_\theta$ . As a result, the encoder can be trained exclusively with the conventional cross-entropy loss, ensuring that the imputation task does not negatively impact the pretraining performance. The total pretraining loss is formalized as:

$$\min_{f_\theta, j_\theta} \mathcal{L}_S = -\mathbb{E}_{(X_S, y_S) \sim \mathcal{D}_S} \mathcal{L}_{ce} + \mathcal{L}_{mapu}^S, \quad (3)$$

where  $\mathcal{L}_{ce} = \sum_{k=1}^K \mathbb{1}_{[y_S=k]} \log(\hat{p}_k)$  represents the standard cross-entropy loss between the predicted label and the true label,  $\hat{p}_k^i$  represents the predicted probability for class  $k$  and sample  $i$ , and

$\mathcal{L}_{mapu}^S$  represents the training loss for our temporal imputer network on the source data to capture the source temporal information.

In the target adaptation step, the objective is to optimize the target encoder,  $f_\theta$ , by balancing the temporal imputation loss and the generic source-free loss to achieve temporal consistency and perform adaptation on the target domain. This can be formalized as follows:

$$\min_{f_\theta} L_T = \mathbb{E}_{X_T \sim \mathcal{D}_T} \mathcal{L}_{sf} + \alpha \mathcal{L}_{mapu}^T, \quad (4)$$

where  $\alpha$  is a hyperparameter that regulates the relative importance of the temporal imputation task, and  $\mathcal{L}_{sf}$  represents the generic loss used by the SFDA method to adapt the target domain to the source domain.

## 4 EXPERIMENTAL SETTINGS

### 4.1 Datasets

We evaluate our proposed method on three real-world datasets spanning three time series applications, i.e., machine fault diagnosis, human activity recognition, and sleep stage classification. The selected datasets differ in many aspects, as illustrated in Table 1, which leads to a considerable domain shift across different domains.

**4.1.1 UCIHAR Dataset.** This dataset focuses on human activity recognition tasks. Three types of sensors have been used to collect the data, i.e., accelerometer sensor, gyroscope sensor, and body sensor, where each sensor provides three-dimensional readings, leading to a total of 9 channels per sample, with each sample containing 128 data points. The data is collected from 30 different users and each user is considered as one domain. In our experiments, five cross-user experiments are conducted, where the model is trained on one user and tested on different users to evaluate its cross-domain performance [1].

**4.1.2 Sleep Stage Classification (SSC) Dataset.** The Sleep Stage Classification (SSC) task involves categorizing Electroencephalography (EEG) signals into five distinct stages, namely Wake (W), Non-Rapid Eye Movement stages (N1, N2, N3), and Rapid Eye Movement (REM). To accomplish this, we utilize the Sleep-EDF dataset [10], which comprises EEG readings from 20 healthy subjects. In line with previous studies [7], we select a single channel, specifically Fpz-Cz, and utilize 10 subjects to construct five cross-domain experiments.

**4.1.3 Machine Fault Diagnosis (MFD) Dataset.** This dataset has been collected by Paderborn university for the fault diagnosis application, where the vibration signals are leveraged to identify different types of incipient faults. The data has been collected under four different working conditions. Each data sample consists of a single univariate channel and 5120 data points following previous works [18, 25]. In our experiments, each working condition is considered as one domain, where we utilize five different cross-condition scenarios to evaluate the domain adaptation performance.

More details about the datasets are included in Table 1.

**Table 1: Details of the adopted datasets (C: #channels, K: #classes, L: sample length).**

Dataset	C	K	L	# training samples	# testing samples
UCIHAR	9	6	128	2300	990
SSC	1	5	3000	14280	6130
MFD	1	3	5120	7312	3604

## 4.2 Implementation Details

*Encoder Design.* In our study, we adopt the encoder architecture presented in existing works [8, 26], which is a 1-dimensional convolutional neural network composed of three layers with filter sizes of 64, 128, and 128 respectively. Each conventional layer was followed by the application of a rectified linear unit activation function and batch normalization.

*MAPU Parameters.* For the purpose of temporal masking, a masking ratio of 1/8 is utilized across all datasets in our experiments. To perform the imputation task, a single-layer recurrent neural network with a hidden dimension of 128 is employed for all datasets. In addition, our method includes a primary hyperparameter,  $\alpha$ , which is set to 0.5 for all datasets in our evaluation.

*Unified Training Scheme.* To provide a fair and valid comparison with source-free baseline methods, we adhered to their established implementations [20, 36, 37] while incorporating the same backbone network and training procedures utilized in our proposed method. In accordance with the AdaTime framework [26], all the models are trained for a total of 40 epochs, using a batch size of 32, with a learning rate of 1e-3 for UCIHAR and 1e-4 for SSC and MFD. Also, the macro F1-score (MF1) metric [26] has been used to ensure a reliable evaluation under data imbalance situations, where we report the mean and the standard deviation of three consecutive runs for each cross-domain scenario.

## 4.3 Baseline Methods

To evaluate the performance of our model, we compare it against conventional UDA approaches that assume access to source data during adaptation. These baselines are adapted from the AdaTime benchmark [26]. Additionally, we compare our model against recent source-free domain adaptation methods. To ensure fair evaluation, we re-implement all source-free baselines in our framework, while ensuring the same backbone network and training schemes. Overall the compared methods are as follows:

### Conventional UDA methods.

- Deep Domain Confusion (DDC) [31]: leverages the MMD distance to align the source and target features.
- Deep Correlation Alignment (DCORAL) [29]: aligns the second-order statistics of the source and target distributions in order to effectively minimize the shift between the two domains.
- High-order Maximum Mean Discrepancy (HoMM) [4]: aligns the high-order moments to effectively tackle the discrepancy between the two domains.

- Minimum Discrepancy Estimation for Deep Domain Adaptation (MMDA) [27]: combines the MMD and correlation alignment with entropy minimization to effectively address the domain shift issue.
- Domain-Adversarial Training of Neural Networks (DANN) [9]: leverages gradient reversal layer to adversarially train a domain discriminator network against an encoder network.
- Conditional Domain Adversarial Network (CDAN) [23]: realizes a conditional adversarial alignment by integrating task-specific knowledge with the features during the alignment step for the different domains.
- Convolutional deep adaptation for time series (CoDATS) [33]: employs adversarial training with weak supervision to enhance the adaptation performance on time series data.
- Adversarial spectral kernel matching (AdvSKM)[22]: introduces adversarial spectral kernel matching to tackle the challenges of non-stationarity and non-monotonicity present in time series data.

### Source-free methods.

- Source Hypothesis Transfer (SHOT) [20]: minimizes information maximization loss with self-supervised pseudo labels to identify target features that can be compatible with the transferred source hypothesis.
- Exploiting the intrinsic neighborhood structure (NRC) [36]: captures the intrinsic structure of the target data by forming clear clusters and encouraging label consistency among data with high local affinity.
- Attracting and dispersing (AaD) [37]: optimizes an objective of prediction consistency by treating SFDA as an unsupervised clustering problem and encouraging local neighborhood features in feature space to have similar predictions.

## 5 RESULTS

In this section, we rigorously test our approach against state-of-the-art methods in various time series applications. We also assess the versatility of our method by combining it with different SFDA techniques. Furthermore, we compare the effectiveness of our task to other auxiliary tasks on time series data. Lastly, we examine our model's sensitivity to different importance weights and masking ratios. In our MAPU, we leverage SHOT as the base SFDA method. Nevertheless, our approach is not limited to SHOT and can be effectively integrated with other SFDA methods, as demonstrated in our versatility experiments.

### 5.1 Quantative Results

To assess the efficacy of our approach, we evaluate its performance on three different time series datasets, namely, UCIHAR, SSC, and MFD. Tables 2, 3, and 4 present results for five cross-domain scenarios in each dataset, as well as an average performance across all scenarios (AVG). The algorithms are divided into two groups: the traditional UDA methods are marked with ✗, while the source-free methods are marked with ✓.

*5.1.1 Evaluation on UCIHAR Dataset.* The results presented in Table 2 show the performance of our MAPU in five cross-subject scenarios. Our method demonstrates superior performance in three

**Table 2: Detailed results of the five UCIHAR cross-domain scenarios in terms of MF1 score.**

Algorithm	SF	2→11	12→16	9→18	6→23	7→13	AVG
DDC	✗	60.0±13.32	66.77±8.46	61.41±5.80	88.55±1.42	77.29±2.11	75.67
DCoral	✗	67.2±13.67	64.58±8.72	54.38±9.69	89.66±2.54	90.46±2.96	77.71
HoMM	✗	83.54±2.99	63.45±2.07	71.25±4.42	94.97±2.49	91.41±1.33	84.10
MMDA	✗	72.91±2.78	<b>74.64±2.88</b>	62.62±2.63	91.14±0.46	90.61±2.00	81.40
DANN	✗	98.09±1.68	62.08±1.69	70.7±11.36	85.6±15.71	<u>93.33±0.00</u>	84.97
CDAN	✗	<u>98.19±1.57</u>	61.20±3.27	71.3±14.64	96.73±0.00	<u>93.33±0.00</u>	<u>86.79</u>
CoDATS	✗	86.65±4.28	61.03±2.33	<u>80.51±8.47</u>	92.08±4.39	92.61±0.51	85.47
AdvSKM	✗	65.74±2.69	60.52±1.99	53.25±5.19	79.63±8.52	88.89±3.12	74.67
SHOT	✓	<b>100.0±0.00</b>	<u>70.76±6.22</u>	70.19±8.99	<b>98.91±1.89</b>	93.01±0.57	86.57
NRC	✓	97.02±2.82	72.18±0.59	63.10±4.84	96.41±1.33	89.13±0.54	83.57
AaD	✓	98.51±2.58	66.15±6.15	68.33±11.9	<u>98.07±1.71</u>	89.41±2.86	84.09
<b>MAPU</b>	✓	<b>100.0±0.00</b>	67.96±4.62	<b>82.77±2.54</b>	97.82±1.89	<b>99.29±1.22</b>	<b>89.57</b>

**Table 3: Detailed results of the five SSC cross-domain scenarios in terms of MF1 score.**

Algorithm	SF	16→1	9→14	12→5	7→18	0→11	AVG
DDC	✗	55.47±1.72	63.57±1.43	55.43±2.75	67.46±1.45	54.17±1.79	59.22
DCoral	✗	55.50±1.74	63.50±1.36	55.35±2.64	67.49±1.50	53.76±1.89	59.12
HoMM	✗	55.51±1.79	63.49±1.14	55.46±2.71	67.50±1.50	53.37±2.47	59.06
MMDA	✗	62.92±0.96	<u>71.04±2.39</u>	<b>65.11±1.08</b>	<u>70.95±0.82</u>	43.23±4.31	<u>62.79</u>
DANN	✗	58.68±3.29	64.29±1.08	<u>64.65±1.83</u>	69.54±3.00	44.13±5.84	60.26
CDAN	✗	59.65±4.96	64.18±6.37	64.43±1.17	67.61±3.55	39.38±3.28	59.04
CoDATS	✗	<u>63.84±3.36</u>	63.51±6.92	52.54±5.94	66.06±2.48	46.28±5.99	58.44
AdvSKM	✗	57.83±1.42	64.76±3.00	55.73±1.42	67.58±3.64	55.19±4.19	60.21
SHOT	✓	59.07±2.14	69.93±0.46	62.11±1.62	69.74±1.22	<b>50.78±1.90</b>	62.33
NRC	✓	52.09±1.89	58.52±0.66	59.87±2.48	66.18±0.25	<u>47.55±1.72</u>	56.84
AaD	✓	57.04±2.03	65.27±1.69	61.84±1.74	67.35±1.48	44.04±2.18	59.11
<b>MAPU</b>	✓	<b>63.85±4.63</b>	<b>74.73±0.64</b>	64.08±2.21	<b>74.21±0.58</b>	43.36±5.49	<b>64.05</b>

**Table 4: Detailed results of the five MFD cross-domain scenarios in terms of MF1 score.**

Algorithm	SF	0→1	1→0	1→2	2→3	3→1	AVG
DDC	✗	74.50±5.56	48.91±6.24	89.34±2.16	96.34±3.07	<b>100.0±0.00</b>	81.82
DCoral	✗	79.03±8.83	40.83±5.01	82.71±0.76	98.01±0.67	97.73±3.93	79.66
HoMM	✗	80.80±2.46	42.31±5.90	84.28±1.32	98.61±0.08	96.28±6.45	80.46
MMDA	✗	82.44±4.47	49.35±5.02	<b>94.07±2.72</b>	<b>100.0±0.00</b>	<b>100.0±0.00</b>	85.17
DANN	✗	83.44±1.72	51.52±0.38	84.19±2.10	<u>99.95±0.09</u>	<b>100.0±0.00</b>	83.82
CDAN	✗	<u>84.97±0.62</u>	52.39±0.49	85.96±0.90	99.7±0.45	<b>100.0±0.00</b>	84.60
CoDATS	✗	67.42±13.3	49.92±13.7	<u>89.05±4.73</u>	99.21±0.79	99.92±0.14	81.10
AdvSKM	✗	76.64±4.82	43.81±6.29	83.10±2.19	98.85±0.93	<b>100.0±0.00</b>	80.48
SHOT	✓	41.99±2.78	57.00±0.09	80.70±1.49	99.48±0.31	99.95±0.05	75.82
NRC	✓	73.99±1.36	74.88±8.81	69.23±0.75	78.04±11.3	71.48±4.59	73.52
AaD	✓	71.72±3.96	<u>75.33±4.65</u>	78.31±2.26	90.07±7.02	87.45±11.7	80.58
<b>MAPU</b>	✓	<b>99.43±0.51</b>	<b>77.42±0.16</b>	85.78±7.38	99.67±0.50	<u>99.97±0.05</u>	<b>92.45</b>

of the five scenarios, achieving an overall performance of 89.57%. This exceeds the second-best source-free method by 3%. Notably,

the source-free methods (i.e., SHOT, NRC, and AaD) perform competitively with conventional unsupervised domain adaptation (UDA)

methods that utilize source data. This can be attributed to the two-stage training (i.e., pertaining and adaptation) scheme employed in the source-free methods, which focuses on optimizing the target model for the target domain without considering source performance [24]. Furthermore, our MAPU, with its temporal adaptation capability, outperforms all conventional UDA methods, surpassing the best method (i.e., CDAN) by 2.78%.

**5.1.2 Evaluation on SSC Dataset.** The results of the sleep stage classification task, as presented in Table 3, demonstrate the superior performance of our proposed method, MAPU, over other baseline methods. Our MAPU performs best in three out of the five cross-domain scenarios, with an overall performance of 64.05%. This is higher than the best source-free method, SHOT, and the best conventional UDA method, with an improvement of 1.72% and 1.27% respectively. It is worth noting that source-free methods that rely on features clustering, i.e., NRC and AaD, perform poorly on the SSC dataset due to its class-imbalanced nature. However, our MAPU, with its temporal adaptation capability, is able to handle such imbalance and outperform all source-free methods with a maximum improvement of 4.8% in scenario 16  $\rightarrow$  1.

**5.1.3 Evaluation on MFD Dataset.** The results of the Machine Fault Diagnosis (MFD) task, presented in Table 4, showcase the superior performance of our MAPU when compared to all other baselines. With an average performance of 92.45%, MAPU exceeds the second-best method by a large margin of 7.85%. Additionally, MAPU significantly outperforms baseline methods in the hard transfer tasks (i.e., 0  $\rightarrow$  1 and 1  $\rightarrow$  0), reaching a 14.46% improvement in the latter scenario, while performing competitively with other baseline methods in the easy transfer tasks (i.e., 2  $\rightarrow$  3 and 3  $\rightarrow$  1). Compared to source-free methods, our MAPU achieves the best performance in all cross-domains, surpassing the second-best source-free method, AaD, by 11.87%.

It is worth noting that the performance improvement of our method is relatively large in the MFD dataset compared to other datasets. This is mainly attributed to two reasons. First, the MFD dataset has the longest sequence length among all other datasets, thus, the adaptation of temporal information is more prominent and necessary. Second, unlike other datasets, this dataset has a limited number of classes, i.e., 3 classes, and thus, failing to correctly classify one class can significantly harm the performance.

## 5.2 Ablation Study on Auxiliary Tasks

To demonstrate the effectiveness of our proposed temporal imputation auxiliary task, we conducted evaluations using various auxiliary tasks, including rotation prediction [21] and jigsaw puzzle [2]. We chose three different SFDA backbones, SHOT, NRC, and AaD, for the auxiliary tasks to eliminate the bias to a specific SFDA method. Table 5 shows the average performance of five cross-domain scenarios for each dataset. The results show that our temporal imputation task consistently outperforms the other tasks across all datasets, even when combined with different SFDA backbones. Meanwhile, the baseline tasks, including rotation and jigsaw, not only exhibit limited improvement but also consistently harm the performance in many cases across various datasets. This indicates the inadequacy of these tasks for time series data and highlights the importance of

**Table 5: Comparing the temporal imputation task with conventional auxiliary tasks for time series adaptation.**

Task	UCIHAR	SSC	MFD
SHOT	86.57	62.33	75.82
SHOT+ Rotation	86.78	60.33	84.98
SHOT + Jigsaw	87.83	62.11	85.74
<b>SHOT + Temporal</b>	<b>89.57</b>	<b>64.05</b>	<b>92.45</b>
NRC	83.57	56.84	73.52
NRC+ Rotation	71.62	56.75	72.02
NRC + Jigsaw	70.58	56.91	74.68
<b>NRC + Temporal</b>	<b>86.05</b>	<b>58.78</b>	<b>76.34</b>
AaD	84.09	59.11	80.58
AaD + Rotation	71.52	59.00	84.18
AaD + Jigsaw	83.72	59.17	85.31
<b>AaD + Temporal</b>	<b>87.00</b>	<b>64.05</b>	<b>91.11</b>

considering temporal dynamics to the adaptation performance, as demonstrated by the superior performance of our MAPU approach.

## 5.3 Model Analysis

**5.3.1 Versatility Analysis.** This study investigates the effectiveness of incorporating temporal information into other SFDA methods. To achieve that, we evaluated the performance of three different SFDA methods when used in conjunction with our proposed temporal imputation task on the UCIHAR, SSC, and MFD datasets. Figure 5 shows the average performance of five cross-domain scenarios in each dataset. Our results indicate a significant improvement in performance across all tested datasets through the integration of our temporal imputation task. For instance, on the UCIHAR dataset, we saw a notable 3% boost in performance for the NRC and AaD methods. On the UCIHAR dataset, the NRC and AaD methods all experienced a performance boost of approximately 3% upon integration with our temporal imputation task. The improvements are consistent across the SSC and MFD datasets, demonstrating our approach's effectiveness in providing temporal adaptation capability to existing SFDA methods that are mainly proposed for visual applications.

**5.3.2 Sensitivity Analysis.** This study evaluates the sensitivity of our temporal imputation component to the relative weight  $\alpha$  when integrated with other SFDA methods, as illustrated in Figure 6. The results indicate that our model's performance is relatively stable across a range of values for the  $\alpha$  parameter. Particularly, the highest MF1 score achieved was 89.77, while the lowest accuracy was 87.75, with a difference of only 2%. This observed stability may be attributed to the imputation process being carried out on the feature space rather than the input space. As such, the feature space provides a more abstract representation of the data, making the imputation process free of the variations present in the input space.

**5.3.3 Impact of Masking level.** Here, we systematically examine the impact of the masking ratio on adaptation performance in the context of imputation tasks. Specifically, we employed three different masking ratios (12.5%, 25%, and 50%) and evaluated the



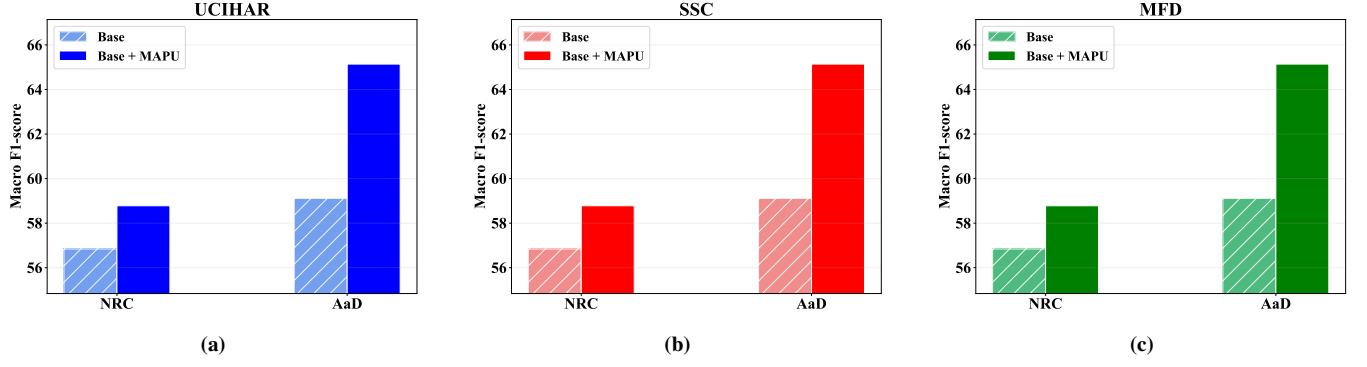


Figure 5: Intergrating temporal imputation with existing SFDA methods among the three datasets.

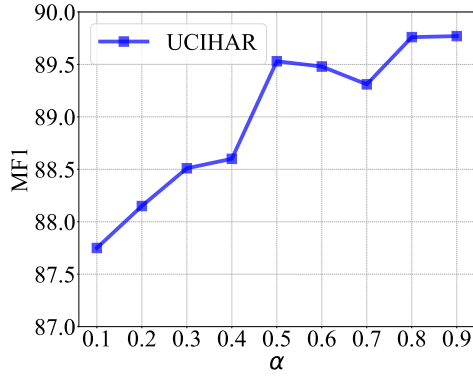


Figure 6: Analysis of adaptation performance with varying relative weight for the temporal imputation component  $\alpha$ .

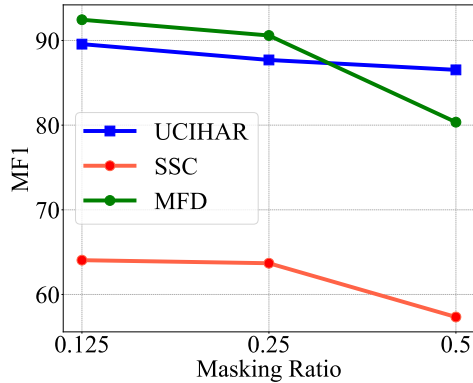


Figure 7: Effect of temporal masking ratio on the adaptation performance for the three datasets.

performance on the three benchmark datasets. The results, shown in Figure 7, reveal a clear trend of improved performance with lower masking ratios. Notably, the best performance was achieved with a masking ratio of 12.5% across all datasets. These findings suggest that excessive masking may negatively impact the adaptation performance in the imputation task.

## 6 CONCLUSION

This paper introduced MASK And impUte (MAPU), a novel method for source-free domain adaptation on time series data. The proposed method addressed the challenge of temporal consistency in time series data by proposing a temporal imputation task to recover the original signal in the feature space rather than the input space. MAPU is the first method to explicitly account for temporal dependency in a source-free manner for time series data. The effectiveness of MAPU is demonstrated through extensive experiments on three real-world datasets, achieving significant gains over the existing methods. This work highlights the potential of MAPU in addressing the domain-shift problem while preserving data privacy in time series applications.

## 7 ACKNOWLEDGMENTS

This work was supported by the Agency of Science Technology and Research under its AME Programmatic (Grant No. A20H6b0151) and its Career Development Award (Grant No. C210112046).

## REFERENCES

- [1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. In *European Symposium on Artificial Neural Networks*.
- [2] Silvia Bucci, Antonio D'Innocente, Yujun Liao, Fabio M Carlucci, Barbara Caputo, and Tatiana Tommasi. 2021. Self-supervised learning across domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 5516–5528.
- [3] Ruichu Cai, Jiawei Chen, Zijian Li, Wei Chen, Keli Zhang, Junjian Ye, Zhuozhang Li, Xiaoyan Yang, and Zhenjie Zhang. 2021. Time Series Domain Adaptation via Sparse Associative Structure Alignment. In *AAAI*.
- [4] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. 2020. HoMM: Higher-order Moment Matching for Unsupervised Domain Adaptation. *AAAI* (2020).
- [5] Tong Chu, Yahao Liu, Jinhong Deng, Wen Li, and Lixin Duan. 2022. Denoised Maximum Classifier Discrepancy for Source-Free Unsupervised Domain Adaptation. In *AAAI*.
- [6] Tong Chu, Yahao Liu, Jinhong Deng, Wen Li, and Lixin Duan. 2022. Denoised Maximum Classifier Discrepancy for Source-Free Unsupervised Domain Adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36, 472–480.
- [7] Emadelddeen Eldele, Zhenghua Chen, Chengyu Liu, Min Wu, Chee-Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2021. An Attention-based Deep Learning Approach for Sleep Stage Classification with Single-Channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* (2021).
- [8] Emadelddeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2021. Time-Series Representation Learning via Temporal and Contextual Contrasting. In *IJCAI*.

- [9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *JMLR* (2016).
- [10] Ary L. Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet Components of a New Research Resource for Complex Physiologic Signals. *Circulation* (2000).
- [11] Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. 2022. Domain Adaptation for Time Series Forecasting via Attention Sharing. In *ICML*.
- [12] Youngeun Kim, Donghyeon Cho, Kyeongtak Han, Priyadarshini Panda, and Sungeun Hong. 2021. Domain Adaptation Without Source Data. *IEEE Transactions on Artificial Intelligence* 2 (2021), 508–518.
- [13] Divya Kothandaraman, Rohan Chandra, and Dinesh Manocha. 2021. SS-SFDA : Self-Supervised Source-Free Domain Adaptation for Road Segmentation in Hazardous Environments. 2021 *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (2021), 3042–3052.
- [14] Jogendra Nath Kundu, Suvaansh Bhambri, Akshay Kulkarni, Hiran Sarkar, Varun Jampani, and R Venkatesh Babu. 2022. Concurrent subsidiary supervision for unsupervised source-free domain adaptation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*. Springer, 177–194.
- [15] Jogendra Nath Kundu, Akshay R Kulkarni, Suvaansh Bhambri, Deepesh Mehta, Shreyas Anand Kulkarni, Varun Jampani, and Venkatesh Babu Radhakrishnan. 2022. Balancing discriminability and transferability for source-free domain adaptation. In *International Conference on Machine Learning*. PMLR, 11710–11728.
- [16] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. 2020. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4544–4553.
- [17] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Nambodiri. 2021. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 615–625.
- [18] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. 2016. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM Society European Conference*, Vol. 3.
- [19] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. 2020. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9641–9650.
- [20] Jian Liang, D. Hu, and Jiashi Feng. 2020. Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation. In *ICML*.
- [21] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. 2021. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 11 (2021), 8602–8617.
- [22] Qiao Liu and Hui Xue. 2021. Adversarial Spectral Kernel Matching for Unsupervised Time Series Domain Adaptation. In *IJCAI*.
- [23] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. 2018. Conditional Adversarial Domain Adaptation. In *NeurIPS*.
- [24] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingui Tan. 2021. Source-free Domain Adaptation via Avatar Prototype Generation and Adaptation. In *International Joint Conference on Artificial Intelligence*.
- [25] Mohamed Ragab, Emadeldeen Eldele, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. 2022. Self-supervised Autoregressive Domain Adaptation for Time Series Data. *IEEE Transactions on Neural Networks and Learning Systems* (2022).
- [26] Mohamed Ragab, Emadeldeen Eldele, Wee Ling Tan, Chuan-Sheng Foo, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. 2022. ADATIME: A Benchmarking Suite for Domain Adaptation on Time Series Data. *arXiv preprint arXiv:2203.08321* (2022).
- [27] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. 2020. On Minimum Discrepancy Estimation for Deep Domain Adaptation. *Domain Adaptation for Visual Understanding* (2020).
- [28] Roshni Sahoo, Divya Shanmugam, and John V. Gutttag. 2020. Unsupervised Domain Adaptation in the Absence of Source Data. *ArXiv abs/2007.10233* (2020).
- [29] Baochen Sun, Jiashi Feng, and Kate Saenko. 2017. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*. Springer, 153–171.
- [30] Sana Tonekaboni, Shalmali Joshi, Kieran Campbell, David K Duvenaud, and Anna Goldenberg. 2020. What went wrong and when? Instance-wise feature importance for time-series black-box models. *Advances in Neural Information Processing Systems* 33 (2020), 799–809.
- [31] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).
- [32] Garrett Wilson and Diane J. Cook. 2020. A Survey of Unsupervised Deep Domain Adaptation. *ACM Trans. Intell. Syst. Technol.* 11, 5, Article 51 (jul 2020), 46 pages.
- [33] Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. 2020. Multi-Source Deep Domain Adaptation with Weak Supervision for Time-Series Sensor Data. In *SIGKDD*.
- [34] Garrett Wilson, Janardhan Rao Doppa, and Diane J. Cook. 2021. CALDA: Improving Multi-Source Time Series Domain Adaptation with Contrastive Adversarial Learning. <https://doi.org/10.48550/ARXIV.2109.14778>
- [35] Haifeng Xia, Handong Zhao, and Zhengming Ding. 2021. Adaptive Adversarial Network for Source-free Domain Adaptation. 2021 *IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 8990–8999.
- [36] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. 2021. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in neural information processing systems* 34 (2021), 29393–29405.
- [37] Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, et al. 2022. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Advances in Neural Information Processing Systems*.
- [38] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. 2021. Deep Subdomain Adaptation Network for Image Classification. *IEEE Transactions on Neural Networks and Learning Systems* (2021).