Domain Adaptation for Time Series Under Feature and Label Shifts

Huan He [1] **Owen Queen** [1] **Teddy Koker** [2] **Consuelo Cuevas** [2] **Theodoros Tsiligkaridis** [2] **Marinka Zitnik** [1]

Abstract

Unsupervised domain adaptation (UDA) enables the transfer of models trained on source domains to unlabeled target domains. However, transferring complex time series models presents challenges due to the dynamic temporal structure variations across domains. This leads to feature shifts in the time and frequency representations. Additionally, the label distributions of tasks in the source and target domains can differ significantly, posing difficulties in addressing label shifts and recognizing labels unique to the target domain. Effectively transferring complex time series models remains a formidable problem. We present RAINCOAT, the first model for both closed-set and universal domain adaptation on complex time series. RAINCOAT addresses feature and label shifts by considering both temporal and frequency features, aligning them across domains, and correcting for misalignments to facilitate the detection of private labels. Additionally, RAINCOAT improves transferability by identifying label shifts in target domains. Our experiments with 5 datasets and 13 state-of-the-art UDA methods demonstrate that RAINCOAT can improve transfer learning performance by up to 16.33% and can handle both closed-set and universal domain adaptation.

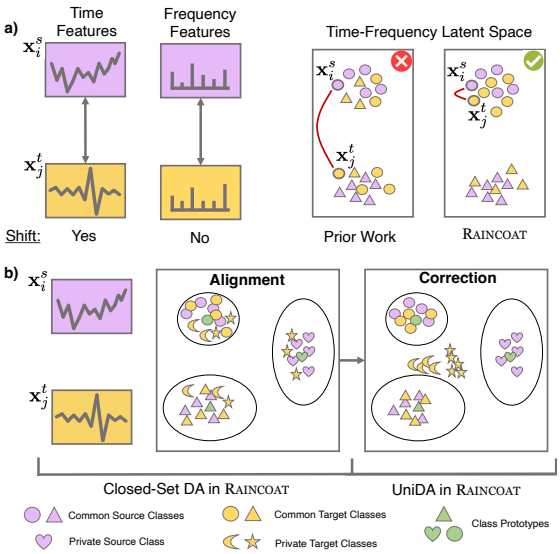


Figure 1. **a)** RAINCOAT captures domain-invariant frequency features under feature and label shifts. **b)** For Closed-Set DA, RAINCOAT aligns source and target domains for greater generalization. For Universal DA, the correction step prioritizes target-specific features to detect private target classes.

1. Introduction

Neural networks have demonstrated impressive performance on time series datasets (Ravuri et al., 2021; Lundberg et al., 2018). However, their performance deteriorates rapidly under domain shifts, making it challenging to deploy these models in real-world scenarios (Zhang et al., 2022a;b). Domain shifts occur when the test distribution is not identical to the training data, even though it is often related (Koh et al., 2021; Luo et al., 2018; Zhang et al., 2013), meaning that latent representations do not generalize to test datasets drawn from different underlying distributions, even if the differences between these distributions are minor. To overcome these challenges, domain adaptation (DA) has emerged as a set of techniques that allow adaptation to new target domains and reduce bias by leveraging unlabeled data in target domains (Ganin et al., 2016; Long et al., 2015).

Training models that can adapt to domain shifts is crucial for robust, real-world deployment. For instance, for healthcare time series, data collection methods vary widely across different clinical sites (domains) (Zhang et al., 2022c), leading to shifts in the underlying features and labels. It is preferable to train a model on a diverse dataset collected from multiple clinics rather than training and applying individual models on smaller, single-domain datasets for each clinic. Additionally, training a model that can detect unknown classes in test data, such as patients with rare diseases (Alsentzer et al., 2022), is advantageous for real-world implementa-

[1]Department of Biomedical Informatics, Harvard University [2]Artificial Intelligence Technology, MIT Lincoln Laboratory. Correspondence to: Huan He, Theodoros Tsiligkaridis, Marinka Zitnik <huan_he@hms.harvard.edu, ttsili@ll.mit.edu, marinka@hms.harvard.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

tion among end-users, such as clinicians (Tonekaboni et al., 2019). Endowing learning systems with DA capabilities can increase their reliability and expand applicability across downstream tasks.

DA is a highly complex problem due to several factors. First, models trained for robustness to domain shifts must learn highly generalizable features; however, neural networks trained using standard practices can rely on spurious correlations created by non-causal data artifacts (Geirhos et al., 2020; DeGrave et al., 2021), hindering their ability to transfer across domains. Additionally, shifts in label distributions across domains may result in *private labels*, i.e., classes that exist in the target domain but not in the source domain (Lipton et al., 2018). In unsupervised DA, a model must generalize across domains when labels from the target domain are not available during training (Long et al., 2018a; Kang et al., 2019a). Therefore, DA methods must be able to identify when a private label is encountered in the target domain without any prior supervision on detecting these unknown labels (You et al., 2019; Fu et al., 2020). Yet, that is not possible by techniques that rely on training samples that simulate predicting unknown labels. This highlights the need for time series DA methods that 1) produce *generalizable representations robust to feature and label shifts*, and 2) expand the scope of existing DA methods by supporting both *closed-set* and *universal* DA.

DA becomes even more challenging when applied to time series data. Domain shifts can occur in both the time and frequency features of time series, which can create a shift that highly perturbs time features while frequency features are relatively unchanged, or vice versa (Figure 1a). Previous time series DA methods fail to explicitly model frequency features. Further, models can fail to generalize due to shortcut learning (Brown et al., 2022), which occurs when the model focuses on time-space features while overlooking crucial underlying concepts in the frequency-space domain, leading to limited poor performance on data unseen during training. Additionally, universal DA—when no assumptions are made about the overlap between labels in the source and target domains—is an unexplored area in time series research (Figure 1b).

Present Work. We introduce RAINCOAT (fRequency-augmented AlIgN-then-Correct for dOmain Adaptation for Time series), a novel domain adaptation method for time series data that can handle both feature and label shifts (as shown in Figure 1). Our method is the first to address both closed-set and universal domain adaptation for time series and has the unique capability of handling feature and label shifts. To achieve this, we first use time and frequency-based encoders to learn time series representations, motivated by inductive bias that domain shifts can occur via both time or frequency feature shifts. We use Sinkhorn divergence for source-target feature alignment and provide both empirical evidence and theoretical justification for its superiority over other popular divergence measures. Finally, we introduce an "align-then-correct" procedure for universal DA, which first aligns the source and target domains, retrains the encoder on the target domain to correct misalignments, and then measures the difference between the aligned and corrected representations of target samples to detect unknown target classes (as shown in Figure 2). We evaluate RAINCOAT on five time-series datasets from various modalities, including human activity recognition, mechanical fault detection, and electroencephalogram prediction. Our method outperforms strong baselines by up to 9.0% for closed-set DA and 16.33% for universal DA. RAINCOAT is available at https://github.com/mims-harvard/Raincoat.

2. Related Work

General Domain Adaptation. General domain adaptation (DA), leveraging labeled source domain to predict labels on the unlabeled target domain, has a wide range of applications (Ganin and Lempitsky, 2015; Sener et al., 2016; Zhang et al., 2018; Perone et al., 2019; Ramponi and Plank, 2020). We organize DA methods into three categories: 1) *Adversarial training:* A domain discriminator is optimized to distinguish source and target domains, while a deep classification model learns transferable features indistinguishable by the domain discriminator (Hoffman et al., 2015; Tzeng et al., 2017; Motiian et al., 2017; Long et al., 2018a; Hoffman et al., 2018). 2) *Statistical divergence:* These approaches aim to extract domain invariant features by minimizing domain discrepancy in a latent feature space. Widely used measures include MMD (Rozantsev et al., 2016), correlation alignment (CORAL) (Sun and Saenko, 2016), contrastive domain discrepancy (CDD) (Kang et al., 2019a), optimal transport distance (Courty et al., 2017; Redko et al., 2019), and graph matching loss (Yan et al., 2016; Das and Lee, 2018). 3) *Self-supervision:* These general DA approaches incorporate auxiliary self-supervision training tasks. These methods learn domain-invariant features through a pretext learning task, such as data augmentation and reconstruction, for which a target objective can be computed without supervision (Kang et al., 2019b; Singh, 2021; Tang et al., 2021). In addition, reconstruction-based methods achieve alignment by carrying out source domain classification and reconstruction of target domain data or both source and target domain data (Ghifary et al., 2016; Jhuo et al., 2012). RAINCOAT sits in the category of both 2 and 3.

Domain Adaptation for Time Series. While in light of successes in computer vision, limited methods have focused on adaptation approaches for time series data. To date, few DA methods are specifically designed for time series. 1) *Adversarial training:* VRADA (Purushotham et al., 2017)

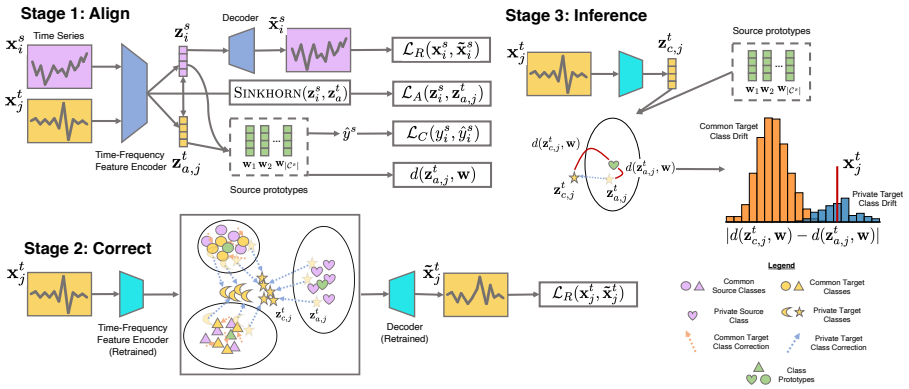


Figure 2. Illustration of the RAINCOAT method for time series DA. Details provided in-text.

builds upon a variational recurrent neural network (VRNN) and trains adversarially to capture complex temporal relationships that are domain-invariant. CoDATS (Wilson et al., 2020) builds upon VRADA but uses a convolutional neural network for the feature extractor. 2) *Statistical divergence:* SASA (Cai et al., 2021) aligns the condition distribution of the time series data by minimizing the discrepancy of the associative structure of time series variables between domains. AdvSKM (Liu and Xue, 2021a) and (Ott et al., 2022) are metric-based methods that align two domains by considering statistic divergence. 3) *Self-supervision:* DAF (Jin et al., 2022) extracts domain-invariant and domain-specific features to perform forecasts for source and target domains through a shared attention module with a reconstruction task. CLUDA (Ozyurt et al., 2022) and CLADA (Wilson et al., 2021) are two contrastive DA methods that use augmentations to extract domain invariant and contextual features for prediction. However, the above methods align features without considering the potential gap between labels from both domains. Moreover, they focus on aligning only time features while ignoring the implicit frequency feature shift (Fig. 1a). In contrast, RAINCOAT considers the frequency feature shift to mitigate both feature and label shift in DA.

Universal Domain Adaptation. Prevailing DA methods assume all labels in the target domain are also available in the source domain. This assumption, known as closed-set DA, posits that the domain gap is driven by feature shift (as opposed to label shift). However, the label overlap between the two domains is unknown in practice. Thus, assuming both feature and label shifts can cause the domain gap is more practical. In contrast to closed-set DA, universal domain adaptation (UniDA) (You et al., 2019) can account for label shift. UniDA categorizes target samples into common labels (present in both source and target domains) or private labels (present in the target domain only). UAN (You et al., 2019), CMU (Fu et al., 2020), and TNT (Chen et al., 2022a) use sample-level uncertainty criteria to measure domain transferability. Samples with lower uncertainty are preferentially selected for adversarial adaptation. However, most UniDA methods detect common samples using sample-level criteria, requiring users to specify the threshold to recognize private labels. Moreover, over-reliance on source supervision neglects discriminative representation in the target domain. DANCE (Saito et al., 2020) uses self-supervised neighborhood clustering to learn features to discriminate private labels. Similarly, DCC (Li et al., 2021a) enumerates cluster numbers of the target domain to obtain optimal cross-domain consensus clusters as common classes. Still, the consensus clusters are not robust enough due to challenging cluster assignments. MATHS (Chen et al., 2022b) detects private labels via mutual nearest-neighbor contrastive learning. In contrast, UniOT (Chang et al., 2022) uses optimal transport to detect common samples and produce representations for samples in the target domain. However, these methods use a feature encoder shared across both domains even though the source and target domains are shifted. In addition, most require fine-tuned thresholds to recognize private labels.

3. Problem Setup and Formulation

Notation. We are given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ of n multivariate time series samples where i-th sample $\mathbf{x}_i \in \mathbb{R}^{T \times d}$ contains readouts of d sensors over T time points. Without loss of generality, we consider regular time series — RAINCOAT can be used with techniques, such as Raindrop (Zhang et al., 2022b) to handle irregular time series. We use $\mathbf{x}_i$ to denote a time series (both univariate and multivariate). Each label y_i in $\mathcal{D}$ belongs to the label set $\mathcal{C}$, i.e., $y_i \in \mathcal{C}$. We use $\mathcal{D}^s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ to denote the source domain dataset with n_s labeled samples, where $\mathbf{x}_i^s$ is a source domain sample and y_i^s is the associated label. The target domain dataset is unlabeled and denoted as $\mathcal{D}^t = \{(\mathbf{x}_i^t)\}_{i=1}^{n_t}$ with n_t unlabeled samples. Source and target label sets are denoted as $\mathcal{C}^s$ and $\mathcal{C}^t$, respectively. Zero, one or more labels may be shared between source and target

domains, which we denote as $\mathcal{C}^{s,t} = \mathcal{C}^s \cap \mathcal{C}^t$. Source and target domains have samples drawn from source and target distributions, $\mathcal{D}^s \sim p_s(\mathbf{x}^s, y^s)$ and $\mathcal{D}^t \sim p_t(\mathbf{x}^t, y^t)$.

We consider two types of domain shifts: feature shift and label shift. Feature shift occurs when marginal probability distributions of $\mathbf{x}$ differ, $p_s(\mathbf{x}) \neq p_t(\mathbf{x})$, while conditional probability distributions remain constant across domains, $p_s(y|\mathbf{x}) = p_t(y|\mathbf{x})$ (Zhang et al., 2013). Label shift occurs when marginal probability distributions of y differ, $p_s(y) \neq p_t(y)$. Feature shifts may occur in time series due to, for example, differences in sensor measurement setup or length of samples. A unique property of time series is that feature shifts may occur in both time and frequency spectra. The importance of modeling shifts in both the time and frequency spectrum is discussed in later sections. Label shift may occur as either a change in the proportion of classes in either domain or as a categorical shift: both domains might contain different classes in their label sets.

Problem 3.1 (Closed-set Domain Adaptation for Time Series Classification). Given the source and target domain time series datasets, $\mathcal{D}^s$ and $\mathcal{D}^t$, whose label sets are the same, $\mathcal{C}^s = \mathcal{C}^t$, and target labels y^t are not available at train time. RAINCOAT specifies a strategy to train a classifier f on $\mathcal{D}^s$ such that f generalizes to $\mathcal{D}^t$, *i.e.*, it minimizes classification risk on $\mathcal{D}^t$: $\mathbb{E}_{\mathbf{x}_i, y_i \sim \mathcal{D}^t}\left[\mathcal{L}_C(f(\mathbf{x}_i), y_i)\right]$, where $\mathcal{L}_C$ is a classification loss function.

In a real-world application, little information may be available on the feature or label distribution of the target domain. Private labels in either the source or target domain may exist, i.e., classes present in one domain but absent in the other. Thus, it is desirable to relax the strict assumption of $\mathcal{C}^s = \mathcal{C}^t$ made by Problem 3.1. We denote source private labels as $\bar{\mathcal{C}}^s = \mathcal{C}^s \backslash \mathcal{C}^t$, target private labels as $\bar{\mathcal{C}}^t = \mathcal{C}^t \backslash \mathcal{C}^s$, and labels shared between domains as $\mathcal{C}^{s,t} = \mathcal{C}^s \cap \mathcal{C}^t$. We denote the access of samples in dataset $\mathcal{D}$ belonging to label set $\mathcal{C}$ as $\mathcal{D}[\mathcal{C}]$, e.g., samples in the target domain belonging to the common label set would be denoted as $\mathcal{D}^t[\mathcal{C}^{s,t}]$. Domains might not have common labels, $\mathcal{C}^{s,t} = \emptyset$, leading to the definition of universal DA.

Problem 3.2 (Universal Domain Adaptation (UniDA) for Time Series Classification). Given our source and target domain time series datasets, $\mathcal{D}^s$ and $\mathcal{D}^t$, where target labels y^t are unavailable at train time. RAINCOAT specifies a stratefy to train a classifier f on $\mathcal{D}^s$ such that f generalizes to $\mathcal{D}^t$, *i.e.*, it minimizes classification risk of a loss function $\mathcal{L}_C$ on samples belonging to $\mathcal{C}^{s,t}$ in $\mathcal{D}^t$: $\mathbb{E}_{\mathbf{x}_i, y_i \sim \mathcal{D}^t[\mathcal{C}^{s,t}]}\left[\mathcal{L}_C(f(\mathbf{x}_i), y_i)\right]$, while identifying samples in *private* target classes, $\mathbf{x}_i \sim \mathcal{D}^t[\bar{\mathcal{C}}^t]$, as *unknown* samples.

4. Preliminaries

Discrete Fourier Transform. Given a series sample $\mathbf{x}$ with d channels and T time points, it is transformed to the frequency space by applying the 1-dim DFT of length T to each channel and then transforming it back using the 1-dim inverse DFT, defined as:

$$\begin{aligned} \text{Forward DFT}: \quad & \mathbf{v}[m] = \textstyle\sum_{t=0}^{T-1} \mathbf{x}[t] \cdot e^{-i2\pi \frac{mt}{T}} \\ \text{Inverse DFT}: \quad & \mathbf{x}[n] = \frac{1}{T} \textstyle\sum_{t=0}^{T-1} \mathbf{v}[m] \cdot e^{i \cdot 2\pi \frac{mt}{T}} \end{aligned} \tag{1}$$

where T = number of points, n = current point index, m = current frequency index, where $m \in [0, T-1]$. We denote the extracted amplitude and phase as $\mathbf{a}$ and $\mathbf{p}$ respectively:

$$\begin{aligned} \mathbf{a}[m] &= \frac{|\mathbf{v}[m]|}{T} = \frac{\sqrt{\mathrm{Re}\left(\mathbf{v}[m]\right)^2 + \mathrm{Im}\left(\mathbf{v}[m]\right)^2}}{T} \\ \mathbf{p}[m] &= \mathrm{atan2}\left(\mathrm{Im}\left(\mathbf{v}[m]\right), \mathrm{Re}\left(\mathbf{v}[m]\right)\right) \end{aligned} \tag{2}$$

where $\mathrm{Im}(\mathbf{v}[m])$ and $\mathrm{Re}(\mathbf{v}[m])$ indicate imaginary and real parts of a complex number, and atan2 is the two-argument form of arctan.

5. RAINCOAT Approach

We start with an overview of RAINCOAT and proceed with (5.2) time-frequency encoding, (5.3) feature alignment, (5.4) unknown sample detection, and (5.5) training and inference.

5.1. Overview

RAINCOAT is an unsupervised method for closed set and universal domain adaptation in time series, addressing Problems 3.1-3.2. RAINCOAT consists of three modules: a time-frequency encoder G_{TF}, a classifier H, and an auxiliary decoder U_{TF}. Sec. 5.2 describes the encoder G_{TF}, which leverages both time and frequency features. Sec. 5.3 describes how Sinkhorn divergence is a suitable divergence measurement to align the source and target domain because frequency features may not share the same support across both domains. Sec. 5.4 motivates the correction step for UniDA. Sec. 5.5 describes how RAINCOAT detects potential unknown samples through analysis of pre- and post-correction embeddings. Finally, Sec. 5.6 provides an overview of RAINCOAT models.

5.2. Time-Frequency Feature Encoder

We begin by highlighting the significance of frequency features in DA for time series. Although various methods have been proposed to solve the time series DA problem under the assumption of feature shift, none of them explicitly address situations where changes in the frequency domain also act as an implicit feature shift. To fill this gap, RAINCOAT encodes both time and frequency features in its latent representations. The source frequency and time features are

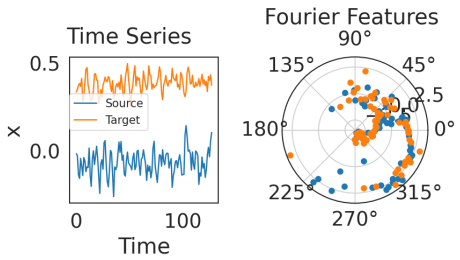


Figure 3. Left: averaged sensor readings (one channel) of the walking activity collected from two persons (source and target). *Right:* corresponding polar coordinates of Fourier features. Fourier features are more domain-invariant than time features.

denoted as $\mathbf{e}^s_{\text{F},i}$ and $\mathbf{e}^s_{\text{T},i}$, respectively, while the target frequency and time features are represented as $\mathbf{e}^t_{\text{F},i}$ and $\mathbf{e}^t_{\text{T},i}$. For simplicity, the superscript indicating the source or target domain is omitted in the rest of the text.

Shift of Frequency Features. We formalize the frequency shift of time series as another type of feature shift. For this purpose, we use the Fourier transform, with the possibility of exploring other options such as wavelets left for future work. A time series $\mathbf{x}_i$ can be represented as a combination of sinusoids, each with a specific frequency, amplitude, and phase, as explained in Sec. 4. If the conditional distributions of the labels with respect to the frequency features are equal ($p_s(y|DFT(\mathbf{x}^s)) = p_t(y|DFT(\mathbf{x}^t))$), but the domains have different frequency features ($p(DFT(\mathbf{x}^s)) \neq p(DFT(\mathbf{x}^t))$), then a frequency shift occurs.

Frequency Features Promote Domain Adaptation. Ben-David et al.; Ben-David et al. demonstrated that the performance of DA techniques is bounded by the divergence between the source and target domains, and that a small feature shift is necessary for DA techniques to be effective. However, unsupervised DA methods for time series align only time features ($\mathbf{e}^s_{\text{T},i}$ and $\mathbf{e}^t_{\text{T},i}$), leading to sub-optimal performance when the time feature shift is large. By including frequency features in the encoder G_{TF}, we can uncover potential invariant features across domains and improve transferability. For instance, Figure 3 illustrates the sensor readings of walking activity from two different individuals ($\mathbf{x}^s_i$ and $\mathbf{x}^t_i$) in the WISDM dataset (Kwapisz et al., 2011) and their corresponding Fourier features ($\mathbf{e}^s_{\text{F},i}$ and $\mathbf{e}^t_{\text{F},i}$). Using only time features would result in poor predictions in the target domain due to a significant time feature shift between $\mathbf{x}^s_i$ and $\mathbf{x}^t_i$. On the other hand, frequency features from different domains do not exhibit significant feature shifts and thus are domain invariant. This suggests that incorporating frequency features can lead to more accurate predictions in the target domain as DA aims to extract domain-invariant features. For this reason, RAINCOAT uses both time and frequency features in domain alignment.

Frequency Feature Encoder. Inspired by Fourier neural operator (FNO) (Li et al., 2021b), RAINCOAT applies convolution on low-frequency modes of the Fourier transform of $\mathbf{x}_i$. We make two modifications to improve the utility of Fourier convolution for DA: *1) Prevent Frequency Leakage:* Discrete Fourier Transform considers inputs $\mathbf{x}_i$ to be periodic. Violation of such assumption results in frequency leakage (Harris, 1978). Specifically, given two window sliced time series $\mathbf{x}^s_i$ and $\mathbf{x}^t_i$, applying DFT (1) could return perturbed and noisy $\mathbf{v}^s_i$ and $\mathbf{v}^t_i$ which may lead to noisy-biased domain alignment. To prevent aligning on noisy frequency features, RAINCOAT applies a smoothing function (cosine function) before applying DFT. *2) Consider amplitude* $\mathbf{a}_i$ *and phase* $\mathbf{p}_i$ *information:* Instead of using inverse DFT to convert $\mathbf{v}_i$ back to time-space which is an unnecessary step for frequency feature extraction, RAINCOAT extracts the polar coordinates of frequency coefficients to keep both low-level ($\mathbf{a}_i$) and high-level ($\mathbf{p}_i$) semantics. The frequency space features $\mathbf{e}_{\text{F}}$ is a concatenation $[\mathbf{a}_i; \mathbf{p}_i]$.

Now we summarize how G_{FT} encodes time-frequency feature from $\mathbf{x}_i$. Define a convolution operator "$*$" and weight matrix $\mathbf{B}$, the encoder G_{F} encodes frequency features $\mathbf{e}_{\text{F},i}$ by: *1) Smooth:* $\mathbf{x}_i = \text{Smooth}(\mathbf{x}_i)$, *2) DFT:* $\mathbf{v}_i = \text{DFT}(\mathbf{x}_i)$, *3) Convolution:* $\tilde{\mathbf{v}}_i = \mathbf{B} * \mathbf{v}_i$, *4) Transform:* $\mathbf{a}_i, \mathbf{p}_i \leftarrow \tilde{\mathbf{v}}_i$ (Use Eq. 2), *5) Extract:* $\mathbf{e}_{\text{F},i} = [\mathbf{a}_i; \mathbf{p}_i]$ The time features $\mathbf{e}_{\text{T},i}$ can be obtained using any existing time feature encoder, such as CNNs. Finally, the latent representation $\mathbf{z}_i$ is a concatenation of frequency and time features $[\mathbf{e}_{\text{F},i}; \mathbf{e}_{\text{T},i}]$. Details are in Appendix B.

5.3. Domain Alignment of Time-Frequency Features

Next, we address the question of what is the appropriate metric to align frequency features between $\mathbf{e}^s_{\text{F}}$ and $\mathbf{e}^t_{\text{F}}$. RAINCOAT represents the frequency features as the amplitude and phase, $\mathbf{e}^s_{\text{F},i} = [\mathbf{a}^s_i; \mathbf{p}^t_i], \mathbf{e}^t_{\text{F},i} = [\mathbf{a}^t_i; \mathbf{p}^t_i]$, meaning that the frequency feature shift can be represented as $p_s(\mathbf{a}^s, \mathbf{p}^s) \neq p_t(\mathbf{a}^t, \mathbf{p}^t)$.

Disjoint Support Sets for Frequency Features. An appropriate metric to align frequency features between $\mathbf{e}^s_{\text{F}}$ and $\mathbf{e}^t_{\text{F}}$ is challenging to find. Distance measures such as the total variation distance or Kullback-Leibler divergence are not suitable because they are unstable when the supports of distributions are deformed and do not metricize the convergence in law (Feydy et al., 2019), meaning that they do not effectively capture the discrepancy when $\mathbf{e}^s_{\text{F},i}$ and $\mathbf{e}^t_{\text{F},i}$ have disjoint support. The KL divergence, for example, grows unbounded ($KL(\mathbf{e}^s_{\text{F},i}||\mathbf{e}^t_{\text{F},i}) \to +\infty$) when $\mathbf{e}^t_{\text{F},i}$ and $\mathbf{e}^t_{\text{F},i}$ are far apart, leading to a degradation of alignment and early collapse. An ideal divergence measure could capture the discrepancy even if $\mathbf{e}^s_{\text{F},i}$ and $\mathbf{e}^t_{\text{F},i}$ have disjoint support ($\text{supp}(\mathbf{e}^s_{\text{F},i}) \cap \text{supp}(\mathbf{e}^t_{\text{F},i}) \approx \emptyset$).

The components of frequency features, amplitude $\mathbf{a}$ and

phase $\mathbf{p}$, have different distributions. The phase $\mathbf{p}$ has a uniform distribution over the range of polar angles, which makes it easy to measure the distance between $\mathbf{p}_i^s$ and $\mathbf{p}_i^t$, bounded in the polar coordinate system $\mathbf{p}_i \in [0, 2\pi)$. However, the amplitude $\mathbf{a}$ has a Rayleigh distribution with an unlimited scale, $\mathbf{a}_i \in [0, +\infty)$, making it difficult to measure the distance between $\mathbf{a}_i^s$ and $\mathbf{a}_i^t$ using the KL divergence. The KL divergence can not provide useful gradients when $\mathbf{a}_i^s$ are $\mathbf{a}_i^t$ are far apart. This leads to a lack of alignment when the amplitudes are far apart, as numerically verified in Figure 5 in the Appendix.

Sinkhorn Divergence. The Sinkhorn divergence is an entropy-regularized optimal transport distance that enables the comparison of distributions with disjoint supports. Another metric, maximum mean discrepancy (MMD), addresses the issue of disjoint support by considering the geometry of the distributions. However, we demonstrate that MMD has a theoretical weakness that manifests as vanishing gradients or similar artifacts. To address this, RAINCOAT aligns the source features ($\mathbf{z}_i^s$) and target features ($\mathbf{z}_i^t$) by minimizing a domain alignment loss based on Sinkhorn. Further details are provided in Appendix A.

5.4. Correction Step in RAINCOAT

In this section, we explain how the correction step helps reduce negative transfer by rejecting target unknown samples $\mathbf{x}^t \sim \mathcal{D}^t[\bar{\mathcal{C}}^t]$. The correction step updates the encoder G_{TF} and decoder U_{TF} by solving a reconstruction task on target samples $\mathbf{x}^t \sim \mathcal{D}^t$. This updated G_{TF} repositions the target features $\mathbf{z}_i^t$. The target features before and after the correction step are denoted as $\mathbf{z}_{a,i}^t$ and $\mathbf{z}_{c,i}^t$, respectively.

Motivation for Reconstructing $\mathbf{x}_i^t$. The cluster assumption (Chapelle and Zien, 2005) holds that the input data is separated into clusters and that samples within the same cluster have the same label. Based on this, we argue that preserving target discriminative features $\mathbf{z}_i^t$ is important for UniDA, because such features help generate discriminative clusters, including clusters of target unknown samples, which improves UniDA. To do this, RAINCOAT minimizes a reconstruction loss to adapt the feature encoder G_{TF} and decoder U_{TF}. The target features $\mathbf{z}_{a,i}^t$ before the correction step are generated by a shared encoder G_{TF} that aligns the source and target domains. As a result, the target features of common samples $\mathbf{x}^t \sim \mathcal{D}^t[\mathcal{C}^{s,t}]$ should change less in the latent space than those of target unknown samples $\mathbf{x}^t \sim \mathcal{D}^t[\bar{\mathcal{C}}^t]$. This indicates that the corrected encoder G_{TF} maintains the features of common target samples close to their originally assigned label while letting the features of target unknown samples diverge from their originally assigned label. RAINCOAT leverages this to detect and reject target unknown samples, which we discuss next.

5.5. Inference: Detect Target Private Samples

RAINCOAT detects target unknown samples $\mathbf{x}^t \sim \mathcal{D}^t[\bar{\mathcal{C}}^t]$ by determining the movement of target features before and after the correction step. It assumes that when the target domain contains unknown labels, the distribution of the movement will exhibit a bimodal structure.

For brevity, the feature vector $\mathbf{z}_i^t$ is used as an input to H, which consists of prototypes for each class $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_C]$. Denote the distance (cosine similarity) of $\mathbf{z}_i^t$ to its assigned prototype c as $d(\mathbf{z}_i^t, \mathbf{w}_c)$. Cosine similarity is a reasonable choice because the cross entropy (CE) loss encourages angular separation. It can be interpreted as aligning the feature vectors $\mathbf{z}_i^t$ along its assigned class prototype. The cosine similarity in the form of the dot product gives CE an intrinsic angular property, which is observed in Eq. 3 where features naturally separate in the polar coordinates with CE only. Given a target feature $\mathbf{z}_i^t$ and true label $y_i = c$, the cross entropy can be expressed as:

$$\mathcal{L}_{\text{CE}}(\hat{y}, y) = -\log \frac{\exp\left(\mathbf{w}_c^T \mathbf{z}_i^t\right)}{\sum_j \exp\left(\mathbf{w}_j^T \mathbf{z}_i^t\right)} \propto \sum_{j \neq c} \exp\left(\mathbf{w}_j^T \mathbf{z}_i^t - \mathbf{w}_c^T \mathbf{z}_i^t\right)$$

$$\propto \sum_{j \neq c} \exp\left(\|\mathbf{z}_i^t\|_2 \|\mathbf{w}_j\|_2 \cos(\theta_j) - \|\mathbf{z}_i^t\|_2 \|\mathbf{w}_c\|_2 \cos(\theta_c)\right)$$

As a result, if the target feature $\mathbf{z}_i^t$ is close to its prototypes, then $d(\mathbf{z}_i^t, \mathbf{w}_c)$ will be small, and vice versa. Then RAINCOAT measures the movement by calculating the absolute difference of target features' distance to the assigned prototype before and after correction given by $d_i^{ac} = |d(\mathbf{z}_{a,i}^t, \mathbf{w}_c) - d(\mathbf{z}_{c,i}^t, \mathbf{w}_c)|$.

Next, RAINCOAT detects if there are private target samples in each class by first running a bimodal test on each group of $\mathcal{C}^s$. If the bimodal test tells us d^{ac} has two modes, it then trains a 2-mean cluster to fit the distribution of d^{ac}. For each class, after we obtain the centroid μ_1, μ_2, where $\mu_1 < \mu_2$, RAINCOAT takes μ_2 as our threshold to reject unknown target samples.

5.6. Overview of RAINCOAT Models

During alignment, RAINCOAT trains a classifier H using labeled source dataset $\mathcal{D}^s$ and a feature encoder G_{TF} and decoder U_{TF} using both $\mathcal{D}^s$ and $\mathcal{D}^t$. At the same time, it aligns target features $\mathbf{z}_i^t$ with source features $\mathbf{z}_i^s$ using Sinkhorn divergence. The overall loss function in this step has three terms. First, the sinkhorn distance $\mathcal{L}_A(\mathbf{z}_i^t, \mathbf{z}_i^s)$ urges the target features $\mathbf{z}_i^t$ to be aligned with source features $\mathbf{z}_i^s$. Second, the reconstruction loss $\mathcal{L}_R(\mathbf{x}_i^s, U_{\text{TF}}(G_{\text{TF}}(\mathbf{x}_i^s)))$ promotes learning of semantic features of $\mathcal{D}^s$. Third, the classification loss $\mathcal{L}_C(H(G_{\text{TF}}(\mathbf{x}_i^s)), y_i^s)$ guides the model to classify samples correctly. In summary, the loss in this step is defined as $\mathcal{L} = \mathcal{L}_A + \mathcal{L}_R + \mathcal{L}_C$.

In this step, target common samples could be classified cor-

Algorithm 1 Overview of RAINCOAT

Input: dataset $\mathcal{D}^s$, $\mathcal{D}^t$; epochs E_1, E_2; time-frequency feature encoder, G_{TF}, and decoder, U_{TF} (Alg. 3); prototype classifier H

Stage 1: Alignment (introduced in 5.2 , 5.3))
for E_1 epochs **do**
  Extract $\mathbf{z}_i^s, \mathbf{z}_i^t \leftarrow G_{\text{TF}}(\mathbf{x}_i^s), G_{\text{TF}}(\mathbf{x}_i^t)$
  $\mathcal{L}_A \leftarrow \text{SINKHORN}(\mathbf{z}_i^s, \mathbf{z}_i^t)$ (Alg. 2)
  $\mathcal{L}_R \leftarrow |\mathbf{x}_i^s - U_{\text{TF}}(\mathbf{z}_i^s)|$
  $\mathcal{L}_C \leftarrow CE(y_i^s, H(\mathbf{z}_i^s))$
  Update $U_{\text{TF}}, G_{\text{TF}}, H$ with $\nabla(\mathcal{L}_A + \mathcal{L}_R + \mathcal{L}_C)$
end for

Stage 2: Correction (introduced in 5.4)
Extract features: $\mathbf{z}_{a,i}^t \leftarrow G_{\text{TF}}(\mathbf{x}_i^t)$
Distance to prototypes: $\mathbf{d}_{\text{align}} \leftarrow d(\mathbf{z}_{a,i}^t, H)$
for E_2 epochs **do**
  $\mathcal{L}_R \leftarrow |\mathbf{x}_i^t - (U_{\text{TF}} \circ G_{\text{TF}})(\mathbf{x}_i^t)|$
  Update $U_{\text{TF}}, G_{\text{TF}}$ with $\nabla \mathcal{L}_R$
end for
Extract post-correction: $\mathbf{z}_{c,i}^t \leftarrow G_{\text{TF}}(\mathbf{x}_I^t)$
Re-compute: $\mathbf{d}_{\text{correct}} \leftarrow d(\mathbf{z}_{c,i}^t, H)$

Stage 3: Inference (introduced in 5.5)
$d_i^{ac} = |d(\mathbf{z}_{a,i}^t, \mathbf{w}_c) - d(\mathbf{z}_{c,i}^t, \mathbf{w}_c)|$
for c in C^s **do**
  $p \leftarrow$ Bimodal Test
  if $p < 0.05$ **then** ▷ Bimodal structure detected
    $\mu_c^{\text{common}}, \mu_c^{\text{unknown}} = \text{CLUSTER}(d^{ac}|\hat{y} = c)$
  end if
end for

rectly, and target unknown samples will be misclassified because the G_{TF} aligns all samples without considering the label shift. The correction step in RAINCOAT aims to correct such negative transfer (target unknown samples) by exploiting target-specific discriminative features by minimizing $\mathcal{L}_R(\mathbf{x}_i^t, U_{\text{TF}}(G_{\text{TF}}(\mathbf{x}_i^t)))$.

In the inference step, only the trained classifier H and feature encoder G_{TF} before and after correction are utilized. When a target samples $\mathbf{x}_i^t$ to inference is given, RAINCOAT calculates the movement using d_i^{ac} equation followed by a bimodal test and binary classification (known or unknown) is necessary. An overview of RAINCOAT is in Alg. 1; a detailed overview is in Appendix and Alg. 4.

6. Experiments

6.1. Experimental Setup

Baselines for Closed-Set DA. We consider eight closed-set DA methods. For baselines are general unsupervised DA methods: deep correlation alignment (CORAL) (Sun and Saenko, 2016), CDAN (Long et al., 2018b), decision-boundary iterative refinement training with a teacher (DIRT-T) (Shu et al., 2018), and AdaMatch (Berthelot et al., 2022). We also consider four unsupervised DA methods for time series: CODATS (Wilson et al., 2020), adversarial spectral kernel matching for unsupervised time series domain adaptation (AdvSKM) (Liu and Xue, 2021b), and CLUDA (Ozyurt et al., 2022). We additionally consider source-domain-only training (no transfer) implemented by (Ragab et al., 2022). **Baselines for Universal DA.** We consider 4 state-of-the-art methods that can reject unknown samples: include UAN (You et al., 2019), DANCE (Saito et al., 2020), OVANet (Saito and Saenko, 2021), and UniOT (Chang et al., 2022). **Datasets.** We consider five benchmark datasets from three distinct problem types: (1) human activity recognition: WISDM (Kwapisz et al., 2011), HAR (Anguita et al., 2013), HHAR (Stisen et al., 2015); (2) mechanical fault detection: Boiler (Shohet et al., 2019); and (3) EEG prediction: Sleep-EDF (Goldberger et al., 2000). Further details on datasets are given in Appendix C.1. **Setup for Closed-Set DA.** Individual, participant, or device IDs define domains in the above datasets. Following existing DA research on time series (Ozyurt et al., 2022; Wilson et al., 2020), we select ten pairs of domains to specify source $\mapsto$ target domains, except for the Boiler dataset where we consider all possible configurations (i.e., six scenarios). **Setup for Universal DA.** The WISDM dataset is the most challenging because of the considerable label shift across participants. For example, source participant 29 does not perform the activity 'jog' at all, but target participant 28 performs 'jog' 33% of the time. To this end, we consider WISDHM to examine the performance of in-dataset UniDA. In addition, HHAR and WISDM contain sensor measurements, and each has one private label ('bike' and 'jog'), making them appropriate for cross-dataset evaluation of UniDA.

Evaluation. We report accuracy and macro-F1 calculated using target test datasets. Accuracy is computed by dividing the number of correctly classified samples by the total number of samples. Macro-F1 is calculated using the unweighted mean of all the per-class F1 scores. It treats all classes equally regardless of their support values. For UniDA, the trade-off between correctly predicting common vs. private classes on the target domain is captured using H-score, defined as the harmonic mean between accuracy on common classes CA_c and accuracy on private classes CA_u, H-score $= (2\text{CA}_c\text{CA}_u)/(\text{CA}_c + \text{CA}_u)$. The H-score is high only when both CA_c and CA_u are high. **Implementation.** We adopted Adatime's implementation as a benchmarking suite for domain adaptation on time series data (Ragab et al., 2022)[1], using 1D-CNN as the encoder because it was suggested to outperform more complex networks such as Resnet and TCN, ensuring differences in performance were attributed to the adaptation algorithm.

[1]https://github.com/emadeldeen24/AdaTime

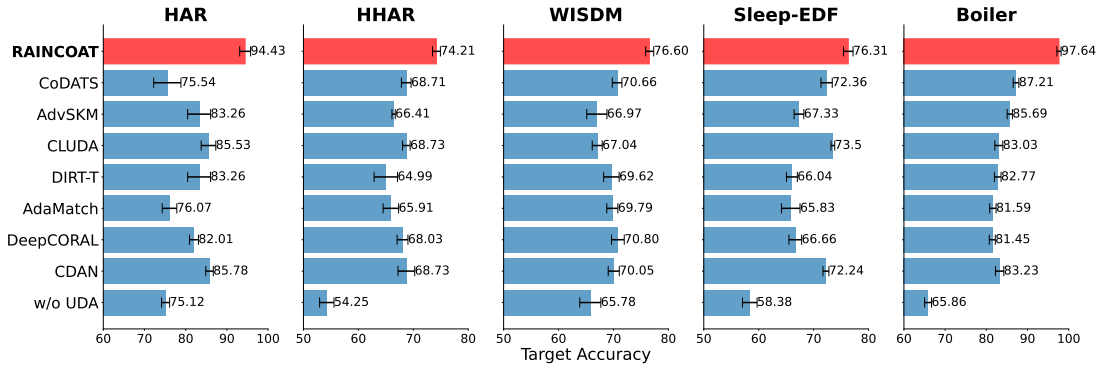


Figure 4. Average performance of multiple Closed-set DA methods across multiple datasets. RAINCOAT consistently outperforms all other methods in accuracy on test sets drawn from the target domain dataset.

Table 1. H-score of UniDA using WISDM, WISDM→HHAR, HHAR→WISDM, Shown: mean H-score over 5 independent runs. See Table 12 in Appendix for additional results.

Source ↦ Target	UAN	DANCE	OVANet	UniOT	RAINCOAT
WISDM 3 ↦ 2	0	0	0.07	0.11	**0.51**
WISDM 3 ↦ 7	0	0	0.2	0.22	**0.52**
WISDM 13 ↦ 15	0	0.14	0.33	0.36	**0.50**
WISDM 14 ↦ 19	0.24	0.28	0.31	0.28	**0.55**
WISDM 27 ↦ 28	0.07	0.07	0.23	0.35	**0.59**
WISDM 1 ↦ 0	0.41	0.39	0.38	0.40	**0.43**
WISDM 1 ↦ 3	0.46	0.49	0.45	0.43	**0.51**
WISDM 10 ↦ 11	0	0	0.34	0.41	**0.53**
WISDM 22 ↦ 17	0.13	0	0.32	0.41	**0.52**
WISDM 27 ↦ 15	0.43	0.51	0.46	0.52	**0.57**
WISDM Avg	0.17	0.19	0.31	0.35	**0.52**
WISDM Std of Avg	0.04	0.05	0.04	0.05	**0.04**
W→H 4 ↦ 0	0	0.14	0.15	0.19	**0.49**
W→H 5 ↦ 1	0.24	0.22	0.25	0.28	**0.53**
W→H 6 ↦ 2	0.14	0.12	0.20	0.25	**0.55**
W→H 7 ↦ 3	0	0.15	0.04	0.14	**0.51**
W→H 17 ↦ 4	0.35	0.28	0.41	0.45	**0.57**
W→H 18 ↦ 5	0.20	0.27	0.29	0.32	**0.47**
W→H 19 ↦ 6	0.19	0.22	0.25	0.28	**0.51**
W→H 20 ↦ 7	0.11	0.17	0.35	0.41	**0.49**
W→H 23 ↦ 8	0.21	0.28	0.47	0.51	**0.57**
W→H Avg	0.16	0.21	0.24	0.28	**0.52**
W→H Std of Avg	0.03	0.02	0.03	0.02	**0.02**
H→W 0 ↦ 4	0.23	0.28	0.33	0.37	**0.45**
H→W 1 ↦ 5	0.19	0.31	0.38	0.42	**0.47**
H→W 2 ↦ 6	0.04	0.17	0.23	0.29	**0.39**
H→W 3 ↦ 7	0.25	0.32	0.34	0.40	**0.42**
H→W 4 ↦ 17	0.31	0.39	0.41	0.40	**0.51**
H→W 5 ↦ 18	0.28	0.34	0.37	0.36	**0.48**
H→W 6 ↦ 19	0.42	0.42	0.46	0.47	**0.49**
H→W 7 ↦ 20	0.39	0.41	0.41	0.44	**0.52**
H→W 8 ↦ 23	0.19	0.28	0.32	0.35	**0.46**
H→W Avg	0.26	0.32	0.36	0.39	**0.47**
H→W Std of Avg	0.05	0.05	0.03	0.04	**0.03**

Higher H-score is better. Best performance is indicated in bold.

6.2. Results

Q1: How effective is RAINCOAT for closed-set DA? Figure 4 shows each method's average accuracy and standard deviation for selected source-target domain pairs on all datasets. Full results are given in Table 10 (accuracy) and Table 11 (Macro-F1). Overall, RAINCOAT has won 5 out of 5 tests (2 metrics in 5 datasets) and makes an average improvement of accuracy (6.77%) and Macro-F1 (9.00%) over with the strongest baseline across datasets. Specifically, RAINCOAT improves prediction accuracy by 8.65% on HAR, 5.48% on HHAR, 5.8% on WISDM, 2.81% on Sleep-EDF, and 10.43% on Boiler over the strongest baseline on each dataset respectively. In particular, RAINCOAT outperforms CLUDA, the state-of-the-art closed-set DA method for time series, by 8.23% (accuracy) and 10.00% (Macro-F1) averaged over all datasets. RAINCOAT captures and aligns time-frequency features across domains which improve knowledge transfer among time series in the presence of feature shift.

Q2: How effective is RAINCOAT for UniDA? We report the average H-score in Table 1 and the average accuracy results in Appendix 12. Results show that RAINCOAT consistently outperforms baselines and achieves state-of-the-art results on DA for time series under both feature and label shift. We note that changes in features and labels of time series data are different from other types of data, such as images, which cause a decrease in the performance of baseline models. However, RAINCOAT has a significant average improvement over the strongest baseline by 16.33% (H-score) across datasets with large gaps. This can be attributed to its time-frequency feature encoder and detection of unknown samples via discriminative features learned using the 'align-and-correct' strategy.

Ablation Studies. Next, we present the setup and results of our ablation studies discussed in Section 6.2. We study the following questions **Q1:** How effective is the time-frequency encoder? **Q2:** Will the correct step decrease the performance when there is no label shift? **Q3:** Is Sinkhorn divergence a better measurement for our time-frequency feature? We evaluate how relevant the model components are for effective DA. We perform the ablation study using WISDM since it is a more challenging dataset and present results in Table 2. When no component is used (1st row in

Table 2. Ablation analysis of RAINCOAT. Specifically, the frequency encoder, Sinhorn Alignment, and Correct Step modules are shown below. When no component is checked (first row), it refers to the source-only model. We evaluate RAINCOAT on both closed-set and universal DA and also include average accuracy across all 10 scenarios (source ↦ target domain) on the WISDM dataset.

	Element of RAINCOAT			Closed Set DA					Universal DA				
	Frequency Encoder	Sinkhorn	Correct	4 ↦ 15	7 ↦ 30	12 ↦ 17	12 ↦ 19	Avg (10 scenarios)	1 ↦ 0	10 ↦ 11	22 ↦ 17	27 ↦ 15	Avg (10 scenarios)
1				79.86	89.32	71.53	54.29	65.78	64.58	54.38	42.98	38.04	40.84
2	✓			89.72	90.12	84.34	83.87	75.22	70.84	65.04	44.81	54.39	42.97
3		✓		82.43	89.88	83.14	76.74	69.66	65.13	57.44	45.14	42.42	41.25
4	✓	✓		95.34	92.36	86.84	84.11	76.24	73.68	72.37	40.79	58.17	44.08
5	✓		✓	90.84	90.01	86.31	79.84	76.04	74.34	66.10	48.01	57.22	46.52
6	✓	✓	✓	97.91	91.28	89.80	85.00	76.60	82.57	76.36	48.16	66.42	53.51

Table 2), it refers to a source-only model. When Sinkhorn is not used (2nd,5th row in Table 2), we use MMD to align features. It can be observed that using the frequency encoder alone (2nd row) results in performance improvement (accuracy) of 9.44% for Closed-set DA and 2.33% for UniDA on average. It demonstrates the effectiveness of a frequency encoder for handling the feature shift of time series. When the frequency encoder (2nd row) is further equipped with a correction step (5th row), it verifies the effectiveness of the correction step when there is a label shift. By comparing the 5th row with the 2nd and 4th row, we find that the correction step does not lead to a performance drop for Closed-set DA. This finding indicates that RAINCOAT is suitable for resolving both feature and label shifts, even if no prior information on feature and label shifts is given. By comparing 2nd row with 4th row, we observe Sinkhorn Divergence brings consistent improvement for both Closed-set DA (1.02%) and UniDA (1.11%), which demonstrates the benefit of Sinkhorn Divergence for aligning frequency features.

We systematically investigate the role of Sinkhorn divergence in RAINCOAT to align time-frequency features. This analysis is particularly relevant because existing methods do not consider frequency features as a potential source of feature shifts. To numerically verify that Sinkhorn divergence is an appropriate divergence measurement for aligning time-frequency features, we conducted experiments by replacing the encoder with our time-frequency feature encoder for additional baselines. We select four representative and strong baselines to ensure a diverse category of adaptation methods: CoDATS, DeepCoral, AdvSKM, and CLUDA. We run Closed-Set DA experiments on HAR datasets and report the average prediction accuracy in Table 3. The results demonstrate that the time-frequency feature encoder achieves the highest accuracy when combined with Sinkhorn divergence, highlighting the effectiveness of using this method to align time-frequency features for time series. Furthermore, all methods show improved prediction accuracy when using our time-frequency encoder, indicating that leveraging and aligning both time and frequency features are crucial for domain adaptation in time series. Additional experiments on RAINCOAT's loss function and sample complexity are in Table 14 and 15 in Appendix D.

Table 3. Accuracy comparison of different domain adaptation methods with and without a time-frequency encoder on the HAR dataset. Results on 'RAINCOAT without our encoder' indicate performance when only Sinkhorn divergence is used.

Method	W/o our encoder	W/ our encoder
CoDATS	75.54	**83.67**
DeepCoral	82.01	**89.75**
AdvSKM	83.26	**89.64**
CLUDA	85.53	**90.62**
RAINCOAT	82.48	***94.43***

7. Conclusion

We introduce RAINCOAT, a domain adaptation approach for time series that addresses both feature and label shifts. RAINCOAT combines time and frequency space features, aligns them across domains, corrects misalignments, and detects label shifts. Experimental results on five datasets demonstrate RAINCOAT's effectiveness, achieving up to 6.7% improvement on closed-set domain adaptation and 16.33% improvement on universal domain adaptation.

Acknowledgements

We gratefully acknowledge the support of the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001 and awards from NIH under No. R01HD108794, Harvard Data Science Initiative, Amazon Faculty Research, Google Research Scholar Program, Bayer Early Excellence in Science, AstraZeneca Research, and Roche Alliance with Distinguished Scientists. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders. The authors declare that there are no conflict of interests.

References

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons,

Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749–760, 2018.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. In *Advances in Neural Information Processing Systems*, 2022a.

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations, ICLR*, 2022b.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

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. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2502–2511, 2018.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, 2013.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15. JMLR.org, 2015.

Angela Zhang, Lei Xing, James Zou, and Joseph C Wu. Shifting machine learning for healthcare from development to deployment and from models to data. *Nature Biomedical Engineering*, pages 1–16, 2022c.

Emily Alsentzer, Michelle M Li, Shilpa N Kobren, Undiagnosed Diseases Network, Isaac S Kohane, and Marinka Zitnik. Deep learning for diagnosing patients with rare genetic diseases. *medRxiv*, pages 2022–12, 2022.

Sana Tonekaboni, Shalmali Joshi, Melissa D. McCradden, and Anna Goldenberg. What clinicians want: Contextualizing explainable machine learning for clinical end use. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 359–380. PMLR, 09–10 Aug 2019. URL https://proceedings.mlr.press/v106/tonekaboni19a.html.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, Nov 2020. ISSN 2522-5839. doi:10.1038/s42256-020-00257-z.

Alex J. DeGrave, Joseph D. Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(77):610–619, Jul 2021. ISSN 2522-5839. doi:10.1038/s42256-021-00338-7.

Zachary Chase Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and correcting for label shift with black box predictors. *ArXiv*, abs/1802.03916, 2018.

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

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4888–4897, 2019a.

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2715–2724, 2019.

Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *European Conference on Computer Vision*, 2020.

Alexander Brown, Nenad Tomasev, Jan Freyberg, Yuan Liu, Alan Karthikesalingam, and Jessica Schrouff. Detecting

and preventing shortcut learning for fair medical ai using shortcut testing (short), 2022.

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

Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. *Advances in neural information processing systems*, 29, 2016.

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*, pages 599–607. Springer, 2018.

Christian S Perone, Pedro Ballester, Rodrigo C Barros, and Julien Cohen-Adad. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. *NeuroImage*, 194:1–11, 2019.

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

Judy Hoffman, Eric Tzeng, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4068–4076, 2015.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. Few-shot adversarial domain adaptation. *Advances in neural information processing systems*, 30, 2017.

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*, pages 1989–1998. Pmlr, 2018.

Artem Rozantsev, Mathieu Salzmann, and Pascal V. Fua. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:801–814, 2016.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV 2016 Workshops*, 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.

Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 849–858. PMLR, 2019.

Junchi Yan, Xu-Cheng Yin, Weiyao Lin, Cheng Deng, Hongyuan Zha, and Xiaokang Yang. A short survey of recent advances in graph matching. *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, 2016.

Debasmit Das and C. S. George Lee. Unsupervised domain adaptation using regularized hyper-graph matching. *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3758–3762, 2018.

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*, pages 4893–4902, 2019b.

Ankit Singh. Clda: Contrastive learning for semi-supervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:5089–5101, 2021.

Shixiang Tang, Peng Su, Dapeng Chen, and Wanli Ouyang. Gradient regularized contrastive learning for continual domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2665–2673, 2021.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European conference on computer vision*, pages 597–613. Springer, 2016.

I-Hong Jhuo, Dong Liu, DT Lee, and Shih-Fu Chang. Robust visual domain adaptation with low-rank reconstruction. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2168–2175. IEEE, 2012.

S. Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. In *ICLR*, 2017.

Garrett Wilson, Janardhan Rao Doppa, and Diane Joyce Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 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. *ArXiv*, abs/2205.03554, 2021.

Qiao Liu and Hui Xue. Adversarial spectral kernel matching for unsupervised time series domain adaptation. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2744–2750. International Joint Conferences on Artificial Intelligence Organization, 8 2021a.

Felix Ott, David Rügamer, Lucas Heublein, Bernd Bischl, and Christopher Mutschler. Domain adaptation for time-series classification to mitigate covariate shift. *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10280–10297. PMLR, 17–23 Jul 2022.

Yilmazcan Ozyurt, Stefan Feuerriegel, and Ce Zhang. Contrastive learning for unsupervised domain adaptation of time series. *ArXiv*, abs/2206.06243, 2022.

Garrett Wilson, Janardhan Rao Doppa, and Diane J. Cook. Calda: Improving multi-source time series domain adaptation with contrastive adversarial learning, 2021.

Liang Chen, Yihang Lou, Jianzhong He, Tao Bai, and Min Deng. Evidential neighborhood contrastive learning for universal domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2022a.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *ArXiv*, abs/2002.07953, 2020.

Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9752–9761, 2021a.

Liang Chen, Qianjin Du, Yihang Lou, Jianzhong He, Tao Bai, and Min Deng. Mutual nearest neighbor contrast and hybrid prototype self-training for universal domain adaptation. In *AAAI*, 2022b.

Wanxing Chang, Ye Shi, Hoang Duong Tuan, and Jingya Wang. Unified optimal transport framework for universal domain adaptation. *ArXiv*, abs/2210.17067, 2022.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando C Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando C Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, mar 2011.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=c8P9NQVtmnO.

F.J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66:51–83, 1978.

J. Feydy, T. Sejourne, F-X. Vialard, S-I. Amari, A. Trouve, and G. Peyre. Interpolating between optimal transport and mmd using sinkhorn divergences. In *AISTATS*, 2019.

Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *International Conference on Artificial Intelligence and Statistics*, 2005.

Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b.

Rui Shu, Hung Hai Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *ArXiv*, abs/1802.08735, 2018.

David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alexey Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Q5uh1Nvv5dm.

Qiao Liu and Hui Xue. Adversarial spectral kernel matching for unsupervised time series domain adaptation. pages 2744–2750, 08 2021b. doi:10.24963/ijcai.2021/378.

Mohamed Ragab, Emadeldeen Eldele, Wee Ling Tan, Chuan-Sheng Foo, Zhenghua Chen, Min Wu, Chee Keong Kwoh, and Xiaoli Li. Adatime: A benchmarking suite for domain adaptation on time series data. *arXiv preprint arXiv:2203.08321*, 2022.

Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8980–8989, 2021.

D. Anguita, Alessandro Ghio, L. Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *The European Symposium on Artificial Neural Networks*, 2013.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor S. Prentow, Mikkel Baun Kjærgaard, Anind K. Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015.

R. Shohet, M. Kandil, and J.J. McArthur. Simulated boiler data for fault detection and classification, 2019. URL https://dx.doi.org/10.21227/awav-bn36.

Ary L. Goldberger, Luis A. Nunes Amaral, L Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and Harry Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 2000.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26, 2013.

Marco Cuturi and Gabriel Peyré. A smoothed dual approach for variational wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162:707–738, 2013.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

A. Genevay, G. Peyre, and M. Cuturi. Learning generative models with sinkhorn divergences. In *AISTATS*, 2018.

Soheil Kolouri, Phillip E. Pope, Charles E. Martin, and Gustavo K. Rohde. Sliced wasserstein auto-encoders. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1xaJn05FQ.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematical Statistics*, 35:876–879, 1964.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19:297–301, 1965.

Kamisetty Ramamohan Rao and Pat Yip. *The Transform and Data Compression Handbook*. CRC Press, Inc., USA, 2000. ISBN 0849336929.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 17–23 Jul 2022.

Junsik Kim, Tae-Hyun Oh, Seokju Lee, Fei Pan, and In-So Kweon. Variational prototyping-encoder: One-shot learning with prototypical images. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9454–9462, 2019.

Min-Hung Chen, Zsolt Kira, Ghassan Al-Regib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6320–6329, 2019a.

Donghyun Kim, Yi-Hsuan Tsai, Bingbing Zhuang, Xiang Yu, Stan Sclaroff, Kate Saenko, and Manmohan Chandraker. Learning cross-modal contrastive features for video domain adaptation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13598–13607, 2021.

Jin Chen, Xinxiao Wu, Lixin Duan, and Shenghua Gao. Domain adversarial reinforcement learning for partial domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 33:539–553, 2019b.

Huan He, Yuanzhe Xi, and Joyce C Ho. Fast and accurate tensor decomposition without a high performance computing machine. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 163–170. IEEE, 2020.

Huan He, Shifan Zhao, Yuanzhe Xi, and Joyce Ho. Gda-am: On the effectiveness of solving min-imax optimization via anderson mixing. In *International Conference on Learning Representations*, 2022.

Huan He, Shifan Zhao, Yuanzhe Xi, and Joyce C Ho. Meddiff: Generating electronic health records using accelerated denoising diffusion model, 2023.

Difeng Cai, Yuliang Ji, Huan He, Qiang Ye, and Yuanzhe Xi. Autm flow: atomic unrestricted time machine for monotonic normalizing flows. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume

180 of *Proceedings of Machine Learning Research*, pages 266–274. PMLR, 01–05 Aug 2022.

Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1215–1224, 2021.

Jogendra Nath Kundu, Naveen Venkat, V. RahulM., and R. Venkatesh Babu. Universal source-free domain adaptation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4543–4552, 2020.

Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8958–8967, 2021.

Yuecong Xu, Jianfei Yang, Haozhi Cao, Keyu Wu, Min Wu, and Zhenghua Chen. Source-free video domain adaptation by learning temporal consistency for action recognition. In *European Conference on Computer Vision*, 2022.

A. Further Information on Domain Alignment of Time-Frequency Feature

We first show the distributions of Fourier amplitude and phase.

$$
\begin{aligned}
f(a,p) &= a \times f(x = a\sin p, y = a\cos p) \\
&= a \times \frac{1}{2\pi} \cdot \exp\left(-\frac{a^2\left(\sin^2\theta + \cos^2\theta\right)}{2}\right) \cdot \\
&= \frac{a}{2\pi} \cdot \exp\left(-\frac{a^2}{2}\right) \cdot \mathbb{I} \\
&= a \cdot \exp\left(-\frac{a^2}{2}\right) \cdot \mathbb{I}(a \geqslant 0) \times \frac{1}{2\pi} \cdot \\
&= \text{ Rayleigh } (a \mid 1) \cdot \mathrm{U}(p \mid 0, 2\pi) \\
&\quad (a \geqslant 0, 0 \leqslant p \leqslant 2\pi).
\end{aligned} \tag{3}
$$

We can observe the amplitude can be arbitrarily large, and thus $\mathbf{a}^s$ and $\mathbf{a}^t$ might have a disjoint set when the frequency feature shift is significant. As a result, we need to consider a measurement that can measure the distance of two arbitrary distributions.

Sinkhorn Divergence. We consider two discrete probability measures represented as sums of weighted Dirac atoms:

$$
\boldsymbol{\mu} = \sum_{i=1}^{n} \boldsymbol{\mu}_i \boldsymbol{\delta}_{\mathbf{z}_i} \text{ and } \boldsymbol{\nu} = \sum_{j=1}^{m} \boldsymbol{\nu}_j \boldsymbol{\delta}_{\mathbf{z}_j} \tag{4}
$$

Here, $\boldsymbol{\mu} \in R_+^n$ and $\boldsymbol{\nu} \in R_+^m$ are non-negative vectors of length n and m that sum up to 1. We denote their probabilistic couplings, set $\mathbf{\Pi}$ and cost matrix $\mathbf{C}$, as:

$$
\begin{gathered}
\mathbf{\Pi}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \left\{\mathbf{P} \in \mathbb{R}_+^{n \times m}, \mathbf{P}\mathbf{1}_m = \boldsymbol{\mu}, \mathbf{P}^\top \mathbf{1}_n = \boldsymbol{\nu}\right\} \\
\mathbf{C} = (\mathbf{C}_{ij}) \in \mathbb{R}_+^{n \times m}, \mathbf{C}_{ij} = \|\mathbf{z}_i - \mathbf{z}_j\|^p
\end{gathered} \tag{5}
$$

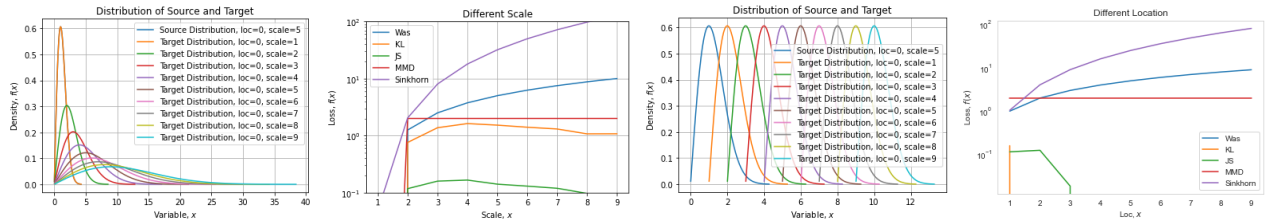
Sinkhorn divergence (Cuturi, 2013; Cuturi and Peyré, 2016) was proposed as an entropic regularization of the Wasserstein distance (Fournier and Guillin, 2013) that interpolates between the pure OT loss for $\eta = 0$ and MMD (Gretton et al., 2012) losses for $\eta \to \infty$ and offers a computationally efficient way to approximate OT costs. It thus provides a good tradeoff between (a) favorable sample complexity and unbiased gradient estimates and (b) non-flat geometry of OT (Genevay et al., 2018; Feydy et al., 2019). The Sinkhorn divergence between $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ is given by

$$
\mathcal{S}_\eta(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \{\langle \boldsymbol{C}, \mathbf{P} \rangle + \eta H(\mathbf{P})\}, \tag{6}
$$

where $H(P) = \sum_{i,j} \mathbf{P}_{ij} \log(\mathbf{P}_{ij})$ is the negative entropy and $\eta > 0$ is a regularization parameter. By making η higher, the resulting coupling matrix will be smoother, and as η goes to zero, it will be sparser, with the solution being close to the optimal transport solution. The Sinkhorn algorithm to find such a coupling matrix is efficiently provided in Alg. 2.

Optimal transport losses have appealing geometric properties, but it takes $O(n^3 \log n)$ to compute. On the other hand, discrepancy metrics such as MMD are geometry-aware and can scale up to large batches with a low sample complexity. But we realize that measuring the discrepancy of frequency features using Sinkhorn has a stronger Gradient than MMD. Specifically, consider MMD with an RBF kernel, the gradient of MMD w.r.t. a particular sample $\mathbf{z}^\mathbf{s}$ is $\nabla_{\mathbf{z}^s} D_{MMD}(\mathbf{Z}^s, \mathbf{Z}^t) = \frac{1}{N^2} \sum_j k\left(\mathbf{z}_i^s, \mathbf{z}_j^s\right) \frac{\mathbf{z}_j^s - \mathbf{z}_i^s}{\sigma^2} - \frac{2}{NM} \sum_j k\left(\mathbf{z}_i^s, \mathbf{z}_j^t\right) \frac{\mathbf{z}_j^t - \mathbf{z}_i^s}{\sigma^2}$. When minimizing MMD, the first term is a repulsive term between the samples from $p(\mathbf{z}^s)$, and the second term is an attractive term between the samples from $p(\mathbf{z}^s)$ and $p(\mathbf{z}^t)$. The L2 norm of the term between two samples $\mathbf{z}^s$ and $\mathbf{z}^t$ is small if $\|\mathbf{z}^s - \mathbf{z}^t\|_2$ is either too small or too large. This is saying if $p(\mathbf{z}^s)$ is far away from $p(\mathbf{z}^t)$, the model will not receive strong gradients (bounded by a small norm). From another viewpoint, (Feydy et al., 2019) demonstrated that the norm of MMD strongly relies on the smoothness of the reference measure and tends to have vanishing gradients when points of the measures' support are disjoint. Now let's look at the gradients of Sinkhorn. Denote a Lipschitz cost function as $\mathrm{C}(\mathbf{z}^s, \mathbf{z}^t)$. For $\eta > 0$, the associated Gibbs kernel is defined through

$$
k_\eta : (\mathbf{z}^s, \mathbf{z}^t) \in \mathcal{Z}^s \times \mathcal{Z}^t \mapsto \exp(-\mathrm{C}(\mathbf{z}^s, \mathbf{z}^t)/\eta)
$$



(a) (b) (c) (d)

Figure 5. (a) Rayleigh distributions with different scales (b) JSD and KLD (c) Rayleigh distributions with different locations (d) JSD and KLD. The Mean Squared Error (MSE) measure exhibits a rapid increase and explodes when there is a significant location shift between distributions. As shown in Figure b, using the Kullback-Leibler (KL) divergence helps mitigate this issue, but it is still not bounded. Furthermore, as depicted in Figure c, both Jensen-Shannon Divergence (JSD) and KL divergence struggle to provide meaningful gradients when there is a substantial location shift. This observation is consistent with the fact that JSD cannot offer usable gradients when distributions are supported on non-overlapping domains, as explained in (Kolouri et al., 2019). The Wasserstein distance demonstrates a linear relationship with the shift, but it also lacks a bound.

(Feydy et al., 2019) show that the Sinkhorn divergence gradient w.r.t a particular sample $\mathbf{z_i^s}$ is largely determined by the magnitude of:

$$\eta \left(\log(\exp(-C(\mathbf{z}_i^s, \mathbf{z}_j^s))/\eta)) - \log(\exp(-C(\mathbf{z}_i^s, \mathbf{z}_j^t))/\eta))\right). \tag{7}$$

Different from MMD, the cost function $C(\mathbf{z}^s, \mathbf{z}^t)$ replaces the Euclidean distance with an absolute distance $|\mathbf{z}_i^s - \mathbf{z}_j^t|$. Then, the gradient is always strong regardless of the closeness between $\mathbf{z}_i^s$ and $\mathbf{z}_j^t$. To numerically verify this claim, we compare the magnitude of the gradients of different shifts in Figure 5. It shows that Sinkhorn has stronger gradients than alternative approaches.

Algorithm 2 Simplified illustration of computation of Sinkhorn Divergence (Sinkhorn, 1964)

function SINKHORN DIVERGENCE(z^s, z^t)
  $a, b \leftarrow \mathbf{1}_n/n, \mathbf{1}_n/n$
  $\boldsymbol{C} \leftarrow \|z^s - z^t\|^p$
  $\boldsymbol{K} \leftarrow \exp^{(-C/\eta)}$
  for $j \leftarrow 1$ to J **do**
    $\boldsymbol{a}^{(j)} \leftarrow \boldsymbol{\mu} \oslash \boldsymbol{K}\boldsymbol{b}^{(j-1)}; \boldsymbol{b}^{(j)} \leftarrow \boldsymbol{\nu} \oslash \boldsymbol{K}^\top \boldsymbol{a}^{(j-1)}$
  end for
  $\mathcal{L}_{align} \leftarrow \sum \boldsymbol{C}\text{diag}(\boldsymbol{a}^{(j)})\boldsymbol{K}\text{diag}(\boldsymbol{b}^{(j-1)})$
  return $\mathcal{L}_{align}$
end function

B. Details on Neural Networks RAINCOAT Algorithm

Encoder. An aspect that has not been adequately emphasized is the composition of a practical time series, which consists of a blend of numerous oscillations at various frequencies, potentially even infinite. In order to address this, RAINCOAT takes into account both time and frequency characteristics during encoding. We use DFT in our work and leave other approaches for future work. To mitigate the issue of frequency leakage, RAINCOAT incorporates a smoothing process on the input $\mathbf{x}_i$. The selection of a suitable smoothing function presents several options, yet the distinctions between them are often negligible in practical applications. In our approach, we utilize either the cosine or Hann window w as a smoothing function, commonly known as tapering functions. These functions are designed using a raised cosine with optimized non-zero endpoints to minimize the impact of nearby side lobes. It is defined as:

$$w[n] = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) \quad 0 \le n \le N-1 \tag{8}$$

Following the smoothing process, the smoothed signal $\mathbf{x}_i$ undergoes the Discrete Fourier Transform (DFT), resulting in the transformed vector $\mathbf{v}_i$. However, the direct implementation of DFT, as shown in Equation (1), can be computationally

inefficient for long signals. To address this, we can leverage the fast Fourier transform (FFT) algorithm (Cooley and Tukey, 1965) to scale up the computation efficiently. An important property of the Fourier domain representation of real signals is the Hermitian property: $\mathbf{v}[m] = \mathbf{v}[-m]$. This property implies that we can save memory by storing only the one-sided representation containing positive frequencies. By doing so, we can reduce memory requirements by half. For a comprehensive understanding of DFT, please refer to Rao and Yip (2000).

Next, RAINCOAT applies a convolution operator specifically on the **low-frequency modes** of $\mathbf{v}_i$, which aligns with existing approaches in neural network-based frequency analysis. The rationale behind this step is that by focusing on low-frequency modes, the operator smooths out high-frequency details that often exhibit less structure compared to their low-frequency counterparts. This process helps to preserve the low-rank structure of signals, facilitating alignment. Unlike previous works such as (Li et al., 2021b; Zhou et al., 2022), RAINCOAT does not incorporate an additional linear transform, as this step is employed to preserve time-space features. Instead, RAINCOAT adopts a time feature encoder.

Subsequently, we extract the amplitude and phase from the output of the convolutional layer, as we have observed that representing these features in polar coordinates (amplitude-phase representation) tends to be more domain-invariant and introduces a useful inductive bias into the model.

Finally, the extracted frequency features are concatenated with the time features. Various approaches, such as manifold alignment and self-attention, can be explored for feature fusion in future work. However, we defer these investigations to our future research.

The process of frequency-space feature extraction is as follows: given a time series $\mathbf{x}$, it is first multiplied by the Hann window function (8) to mitigate frequency leakage. Subsequently, a convolution operation is applied to the smoothed signal. This results in the frequency-space feature $e_{\mathcal{F}}$, which is obtained by concatenating the results using (2). For the extraction of time-space features, any appropriate network architecture can be utilized. In this work, we adopt a Convolutional Neural Network (CNN) to ensure a fair comparison with existing studies. The pseudocode for the time-frequency feature extraction is presented in Algorithm 3.

Fourier Neural Operator. Fourier neural operator (FNO) (Li et al., 2021b) performs temporal predictions by combining the Fourier transform with neural networks. Define a convolution operator "$*$" and weight matrix $\mathbf{B}$, the Fourier layer in FNO can be summarized as:

(1) DFT	$\mathbf{v} = [\mathrm{DFT}(\mathbf{x})]$
(2) Frequency Convolution	$\tilde{\mathbf{e}}_{\mathcal{F}} = \mathbf{B} * \mathbf{v}$
(3) IDFT	$\tilde{\mathbf{x}} = [\mathrm{IDFT}(\tilde{\mathbf{e}}_{\mathcal{F}})]$

FNO then adds the output of the Fourier layer with the bias term (a linear transformation) and applies the activation function. RAINCOAT differs a lot from FNO. The only shared component is the frequency convolution, as we mentioned previously.

Decoder. In order to acquire discriminative features, RAINCOAT employs a decoder that is trained through a reconstruction task. Given a latent representation $\mathbf{z}_i$ obtained from either the source or target, we decompose it into frequency and time features. By performing separate reconstructions on both $\mathbf{e}_{\mathrm{F}}$ and $\mathbf{e}_{\mathrm{T}}$, we can easily reconstruct the original signal $\mathbf{x}$. For the frequency feature reconstruction, we apply the inverse Discrete Fourier Transform (DFT) on $\mathbf{e}_{\mathrm{F}}$, while for the time feature reconstruction, a standard deconvolution network is utilized on $\mathbf{e}_{\mathrm{T}}$. These reconstructed frequency and time components are then combined to form $\hat{\mathbf{x}}$, an approximation of the original signal. To train the reconstruction task, we employ the L1 loss function.

Prototypical Classifier. In RAINCOAT, we employ a prototypical classifier, inspired by the work of Kim et al. (2019). The normalized feature vector $\mathbf{z}$ serves as the input to the classifier H. The classifier H consists of weight vectors $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_C]$, where C denotes the number of classes. These weight vectors can be interpreted as estimated prototypes for each class. By utilizing the prototypical classifier, RAINCOAT aims to classify input samples based on their similarity to the estimated prototypes. The feature vector $\mathbf{z}$ is compared to each weight vector $\mathbf{w}_i$, and the class with the closest prototype is assigned as the predicted class label. This approach leverages the discriminative power of the prototypes to facilitate accurate classification.

Algorithm 3 Time-Frequency Feature Encoder and Decoder, Domain Alignment via Sinkhorn Divergence

function TIME-FREQ ENCODER $G_{\text{TF}}(x)$
  $x \leftarrow$ smooth(x), as shown in (8)
  $\mathbf{v}_{\text{F}} \leftarrow DFT(x)$
  $\leftarrow$ SPEC-CONV($\mathbf{v}_{\text{F}}$)
  $\mathbf{a}, \mathbf{p} \leftarrow \mathbf{v}_{\text{F}}|, atan2(Im(x_{\text{F}}), Re(x_{\text{F}}))$
  $\mathbf{e}_{\text{F}} \leftarrow \text{CONCAT}(\mathbf{a}, \mathbf{p})$
  $\mathbf{e}_{\text{F}} \leftarrow \text{TIME-CONV}(x)$
  $\mathbf{z} \leftarrow \text{CONCAT}(\mathbf{e}_{\text{F}}, \mathbf{e}_{\text{F}})$
  return $\mathbf{z}$
end function

function TIME-FREQ DECODER $\mathcal{U}_{\mathcal{TF}}(z)$
  $e_{\mathcal{T}}, e_{\mathcal{F}} \leftarrow z$
  $\bar{x}_{\mathcal{T}}, \bar{x}_{\mathcal{F}} \leftarrow \text{CONVTRAN1D}(e_{\mathcal{T}}), \text{IFFT}(e_{\mathcal{F}})$
  $\bar{x} \leftarrow \bar{x}_{\mathcal{T}} + \bar{x}_{\mathcal{F}},$
  return $\bar{x}$
end function

C. Additional Experimental Results

C.1. Dataset Details

We evaluate the performance of RAINCOAT on five benchmark datasets, each with its own characteristics. The datasets we consider are as follows:

(1) WISDM (Kwapisz et al., 2011): This dataset consists of 3-axis accelerometer measurements obtained from 30 participants. The measurements are collected at a frequency of 20 Hz. To predict the activity (label) of each participant during specific time segments, we utilize non-overlapping segments of 128-time steps. The dataset includes six activity labels: walking, jogging, sitting, standing, walking upstairs, and walking downstairs.

(2) Boiler (Shohet et al., 2019): The dataset comprises sensor data from three boilers recorded between March 24, 2014, and November 30, 2016. Each boiler is treated as a separate domain. The objective of this task is to detect mechanical faults specifically related to the blowdown valve of each boiler.

(3) HAR (Anguita et al., 2013): This dataset contains measurements from a 3-axis accelerometer, 3-axis gyroscope, and 3-axis body acceleration. The data is collected from 30 participants at a sampling rate of 50 Hz. Similar to the WISDM dataset, we use non-overlapping segments of 128-time steps for classification. The goal is to classify the time series into six activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying down.

(4) HHAR (Stisen et al., 2015): This dataset consists of 3-axis accelerometer measurements from 30 participants. The measurements are captured at a frequency of 50 Hz. Non-overlapping segments of 128-time steps are used for classification purposes. The dataset includes six activity labels: biking, sitting, standing, walking, walking upstairs, and walking downstairs.

(5) Sleep-EDF (Goldberger et al., 2000): This dataset contains electroencephalography (EEG) readings from 20 healthy individuals. The objective is to classify the EEG readings into five sleep stages: wake (W), non-rapid eye movement stages (N1, N2, N3), and rapid eye movement (REM). In line with prior research, we focus on the Fpz-Cz channel for our analysis.

For detailed statistics regarding each dataset, please refer to Table 4. These datasets cover a range of applications and challenges, allowing us to evaluate the effectiveness and robustness of RAINCOAT across various domains.

C.2. Experimental Details

In this section, we provide implementation details of RAINCOAT and the baseline methods. The implementation was done in PyTorch, based on the code available at here. The experiments were conducted on a NVIDIA GeForce RTX 3090 graphics card.

Algorithm 4 Detailed overview of RAINCOAT

Input: dataset $\mathcal{D}_s$, $\mathcal{D}_t$; epochs E_1, E_2
Initialization: Parameter Γ for Time-Frequency Feature Encoder $G_{\mathcal{TF}}$, Φ for Time-Frequency Feature Decoder $U_{\mathcal{TF}}$, weight vectors $W = [w_1, w_2, \cdots, w_{C^s}]$ for prototypical classifier,.
Stage 1: Align, introduced in Section 5.2 5.3
for $e \leftarrow 1$ to E_1 **do**
  while $\mathcal{D}_t$ not exhausted **do**
    Sample x^s, y^s from $\mathcal{D}_s$, x^t from $\mathcal{D}_t$
    Extract: $z^s \leftarrow G_{\mathcal{TF}}(x^s)$ (use Algorithm 3)
    Extract: $z^t \leftarrow G_{\mathcal{TF}}(x^t)$ (use Algorithm 3)
    Reconstruct $\bar{x}^s \leftarrow \mathcal{U}_{\mathcal{TF}}(z^s)$
    Compute:$\mathcal{L}_{align} = \text{SINKHORN}(z^s, z^t, \epsilon)$ ▷ in algorithm 3
    Compute:$\mathcal{L}_{recon} = |x^s - \bar{x}^s|$
    Predict: $\hat{y_s} = \text{CLASSIFIER}(z^s)$
    Compute $\mathcal{L}_{cls} = CE(y^s, \hat{y^s})$
    $\mathcal{L}_{total} = \mathcal{L}_{recon} + \mathcal{L}_{align} + \mathcal{L}_{cls}$
    Update Γ, Φ, W with $\nabla \mathcal{L}_{total}$
  end while
end for

Stage 2: Correct, introduced in Sec. 5.4
Compute distance to prototypes before correct:
  $d_{align} = \frac{\mathbf{Z}^s \cdot W}{\|\mathbf{Z}^s\| \|\mathbf{W}\|}$
for $e \leftarrow 1$ to E_2 **do**
  while $\mathcal{D}_t$ not exhausted **do**
    Sample x^t from $\mathcal{D}_t$
    Extract: $z^t \leftarrow G_{\mathcal{TF}}(x^t)$ (use Algorithm 3)
    Reconstruct $\bar{x}^t \leftarrow \mathcal{U}_{\mathcal{TF}}(z^t)$
    Compute:$\mathcal{L}_{recon} = |x^s - \bar{x}^s|$
    Update Γ, Φ with $\nabla \mathcal{L}_{recon}$
  end while
end for
Compute distance to prototypes after correct:
  $d(\mathbf{Z}, \mathbf{W}) = \frac{\mathbf{Z}^s \cdot W}{\|\mathbf{Z}^s\| \|\mathbf{W}\|}$

Stage 3: Inference, introduced in Sec. 5.5
Compute drift during correct:
  $drift = |d_{correct} - d_{align}|$
for $c \leftarrow 1$ to C **do**
  Compute DIP statistic: $dip = \text{DIP}(\{drift\}_{y=c})$
  if $dip < 0.05$ **then** ▷ Two modes detected
    $\mu_c^{common}, \mu_c^{private} = \text{K-MEANS}(\{drift\}_{y=c})$
  end if
end for

Table 4. Summary of datasets.

Dataset	#Subjects	#Channels	Length	# Class	#Train	# Test
HAR	30	9	128	6	2,300	990
HHAR	9	3	128	6	12,716	5,218
WISDM	30	3	128	6	1,350	720
Sleep-EDF	20	1	3,000	5	14,280	6,310
Boiler	3	20	36	2	160,719	107,400

Method	Epoch	Batch Size	Learning rate
CoDATS	50	32	$1e-3$
AdvSKM	50	32	$5e-1$
CLUDA	50	32	$1e-2$
DIRT-T	50	32	$5e-4$
AdaMatch	50	32	$3e-3$
DeepCoral	50	32	$5e-3$
CDAN	50	32	$1e-2$
RAINCOAT	50	32	$5e-4$

Table 5. Experimental details for HAR dataset.

To ensure fair comparisons, we carefully selected the appropriate encoder and scale across all methods. This consideration was applied to all our comparisons. For the extraction of time-space features, we utilized a 1D-convolutional neural network (CNN) as the encoder. This configuration was kept consistent across all methods to ensure a fair comparison, where differences in prediction performance could be attributed to the adaptation algorithm itself. The implementation of the 1D-CNN architecture was adapted from a recently published benchmark codebase in the literature (Ragab et al., 2022), which has also been employed by others (Ozyurt et al., 2022). The 1D-CNN architecture consists of three blocks, each consisting of a 1D convolutional layer, followed by a 1D batch normalization layer, a rectified linear unit (ReLU) function for non-linearity, and finally, a 1D max-pooling layer. Extensive benchmark evaluations have demonstrated that the 1D-CNN consistently outperforms more complex backbone networks, such as 1D-Resnet-18 and TCN, hence our choice of the 1D-CNN encoder.

During model training, we employed the Adam optimizer for all methods, with carefully tuned learning rates specific to each method. The hyperparameters of Adam were selected after conducting a grid search on source validation datasets, exploring a range of learning rates from 1×10^{-4} to 1×10^{-1}. The learning rates were chosen to optimize the performance of each method.

Key hyperparameters for RAINCOAT are reported in Tables 5, 6, 7, 8, and 9. The Fourier Frequency modes used for HAR, EEG, HHAR, WISDM, and Boiler datasets are 64, 200, 64, 64, and 10, respectively. For the regularization term used in the Sinkhorn divergence, we consistently used a value of 1×10^{-3} across all datasets and experiments. Additional hyperparameters can be found in the codes.

By providing these implementation details and hyperparameter values, we ensure transparency and reproducibility of the experiments conducted with RAINCOAT.

C.3. t-SNE Visualizations of Learned Representations for Closed-Set DA

We present t-SNE plots of the learned representations using HAR in Figures 6 and 7, which serve as strong evidence of the effectiveness of RAINCOAT for domain adaptation. The t-SNE plots provide visual representations of the feature distributions in both the source and target domains.

The plots clearly depict distinct clusters of data points, corresponding to different activity types, in both the source and target domains. This observation indicates that RAINCOAT successfully preserves the underlying structure of the data, even in the presence of differences in sensor configurations and other domain-specific factors. The distinct clusters in the t-SNE plots validate the ability of our method to capture and discriminate between different activities.

Hyperparameter	Epoch	Batch Size	Learning rate
CoDATS	50	128	$1e-2$
AdvSKM	50	128	$5e-4$
CLUDA	50	128	$5e-4$
DIRT-T	50	128	$5e-4$
AdaMatch	50	128	$5e-4$
DeepCoral	50	128	$5e-4$
CDAN	50	128	$1e-3$
RAINCOAT	50	128	$1e-3$

Table 6. Experimental details for EEG dataset.

Hyperparameter	Epoch	Batch Size	Learning rate
CoDATS	50	64	$1e-3$
AdvSKM	50	64	$3e-4$
CLUDA	50	64	$1e-3$
DIRT-T	50	64	$1e-3$
AdaMatch	50	64	$2e-3$
DeepCoral	50	64	$5e-2$
CDAN	50	64	$1e-3$
RAINCOAT	50	64	$1e-3$

Table 7. Experimental details for WISDM dataset

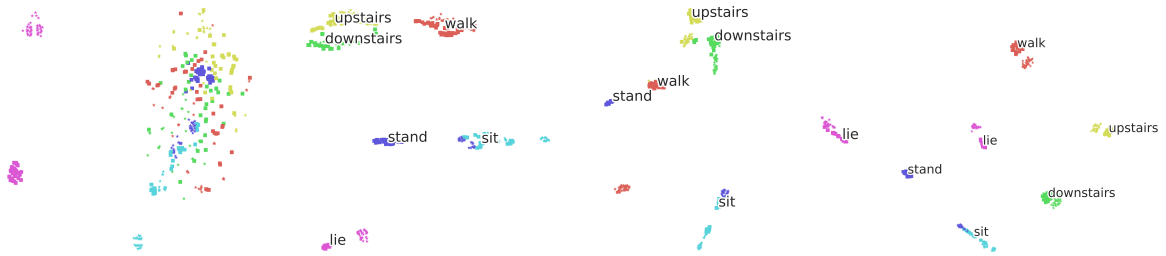
Hyperparameter	Epoch	Batch Size	Learning rate
CoDATS	50	32	$1e-3$
AdvSKM	50	32	$3e-4$
CLUDA	50	32	$1e-3$
DIRT-T	50	32	$1e-3$
AdaMatch	50	32	$3e-3$
DeepCoral	50	32	$5e-4$
CDAN	50	32	$1e-3$
RAINCOAT	50	32	$1e-3$

Table 8. Experimental details for HHAR dataset.

Hyperparameter	Epoch	Batch Size	Learning rate
CoDATS	30	32	$5e-4$
AdvSKM	30	32	$1e-3$
CLUDA	30	32	$1e-3$
DIRT-T	30	32	$1e-3$
AdaMatch	30	32	$3e-3$
DeepCoral	30	32	$5e-4$
CDAN	30	32	$1e-3$
RAINCOAT	50	32	$1e-3$

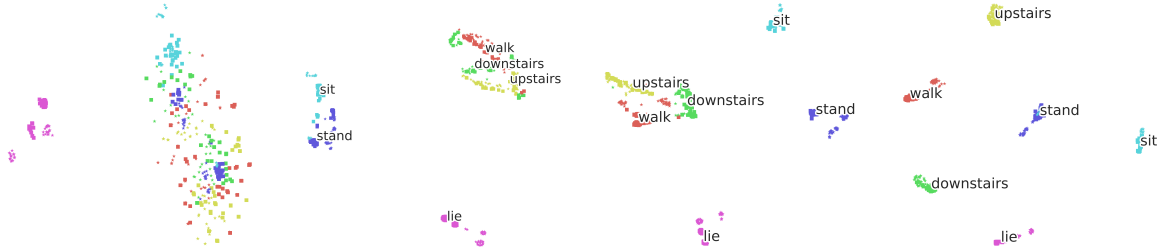
Table 9. Experimental details for Boiler dataset.

Additionally, the t-SNE plots reveal that the clusters in the target domain are generally more tightly grouped and better separated compared to those in the source domain. This suggests that RAINCOAT effectively adapts the model to the target domain, leading to improved performance and more accurate predictions. These findings demonstrate the efficacy of RAINCOAT for domain adaptation and highlight its potential for a wide range of applications, including robotics, healthcare, and sports performance analysis.



(a) HAR 2- HAR 11 (b) DeepCoral (c) CLUDA (d) RAINCOAT

Figure 6. For the HAR dataset of adapting from source 2 to target 11, we generated T-SNE plots of learned embeddings for three different methods. Figure 6(a) depicts the T-SNE visualization of datasets of source and target domains. Figure 6(b), 6(c), and 6(d) represent a different method respectively, and the plots are arranged from left to right (DeepCoral, CLUDA, and RAINCOAT respectively). In each plot, each color corresponds to a different activity label. The square markers represent embeddings of source samples, while the star markers represent embeddings of target samples. These T-SNE plots provide a visual representation of the learned embeddings and demonstrate the effectiveness of the different methods in adapting to the target domain.



(a) HAR 7- HAR 13 (b) DeepCoral (c) CLUDA (d) RAINCOAT

Figure 7. For the HAR dataset of adapting from source 7 to target 13, we generated T-SNE plots of learned embeddings for three different methods. In each plot, each color corresponds to a different activity label. The square markers represent source samples, while the star markers represent target samples. These T-SNE plots provide a visual representation of the learned embeddings and demonstrate the effectiveness of the different methods in adapting to the target domain.

C.4. Full Results of Closed-Set DA

Comprehensive tables presenting the results for Closed-Set Domain Adaptation (DA) experiments are provided in two separate tables, accuracy and macro-F1. Table 10 showcases the accuracy scores, while Table 11 displays the Macro-F1 scores. Upon analyzing the tables, it becomes evident that RAINCOAT consistently outperforms the baseline methods across all datasets in terms of both accuracy and Macro-F1 scores. This demonstrates the superiority of RAINCOAT in effectively adapting to the target domain and achieving improved performance compared to the baseline approaches.

C.5. Full Results of UniDA

Detailed tables containing the results for Universal Domain Adaptation (UniDA) experiments are provided in two separate tables, accuracy and H-scores. Table 12 presents the accuracy scores, while Table 13 displays the H-scores. It is worth noting that accuracy alone is not an appropriate metric for evaluating UniDA since it does not fully reflect the ability to detect target unknown samples. Accuracy can be misleading due to class imbalance issues, resulting in high or low scores without effectively capturing the capability of detecting unknown samples. We conducts three UniDA settings including

Table 10. Prediction accuracy for each dataset between various subjects. Shown: mean Accuracy over 5 independent runs.

Source ↦ Target	w/o UDA	CDAN	DeepCORAL	AdaMatch	DIRT-T	CLUDA	AdvSKM	CoDATS	RAINCOAT
HAR 2 ↦ 11	76.56	85.42	90.63	75.00	80.21	81.77	98.96	68.23	**100**
HAR 6 ↦ 23	67.36	87.50	84.38	80.20	74.31	92.01	88.54	74.31	**95.83**
HAR 7 ↦ 13	83.68	92.01	87.50	85.76	82.99	99.31	92.71	77.43	**100**
HAR 9 ↦ 18	24.65	58.86	46.88	56.59	59.03	67.71	74.65	63.89	**75.69**
HAR 12 ↦ 16	61.11	66.67	65.28	49.65	67.01	65.28	69.44	66.32	**86.52**
HAR 13 ↦ 19	88.89	96.52	95.49	94.79	99.30	94.44	93.05	94.09	**100**
HAR 18 ↦ 21	100	100	100	100	98.61	98.96	100	99.65	**100**
HAR 20 ↦ 6	94.10	95.13	95.49	84.37	92.36	**97.22**	85.41	70.49	93.41
HAR 23 ↦ 13	71.18	82.64	69.79	68.75	74.72	72.92	79.51	56.25	**86.52**
HAR 24 ↦ 12	83.68	93.40	87.50	70.83	94.27	**99.31**	96.87	82.81	93.75
HAR Avg	75.12	85.78	82.01	76.07	83.26	85.53	83.26	75.54	**94.43**
HAR Std of Avg	0.98	**0.91**	1.09	1.77	2.78	1.78	2.79	3.31	1.32
HHAR 0 ↦ 2	64.51	76.19	84.23	84.78	77.83	79.84	78.94	79.61	**87.72**
HHAR 1 ↦ 6	70.63	92.57	90.14	92.31	88.54	**93.40**	87.91	90.90	93.33
HHAR 2 ↦ 4	45.42	52.57	47.08	54.50	50.69	45.90	52.57	60.07	**63.75**
HHAR 4 ↦ 0	32.81	29.09	28.13	36.45	32.22	38.84	33.49	21.80	**46.46**
HHAR 4 ↦ 5	78.32	97.27	90.49	78.45	93.16	94.08	92.64	97.66	**98.05**
HHAR 5 ↦ 1	90.63	96.16	89.91	94.20	91.86	95.57	92.71	97.66	**98.25**
HHAR 5 ↦ 2	25.67	35.04	38.39	41.96	38.62	33.93	36.53	41.44	**42.63**
HHAR 7 ↦ 2	32.37	37.05	34.45	37.65	38.10	37.80	39.95	38.54	**43.32**
HHAR 7 ↦ 5	39.26	75.26	55.73	63.80	72.46	75.26	65.49	58.15	**84.17**
HHAR 8 ↦ 4	62.92	96.11	76.88	64.69	65.83	96.11	83.75	**97.01**	93.75
HHAR Avg	54.25	68.73	68.03	65.91	64.99	68.73	66.41	68.71	**74.21**
HHAR Std of Avg	1.31	1.52	0.99	1.41	2.13	0.69	**0.30**	0.88	0.72
WISDM 2 ↦ 32	81.16	**89.37**	87.92	74.39	77.78	73.91	70.83	77.29	79.71
WISDM 4 ↦ 15	79.86	65.97	62.50	78.47	70.83	67.36	95.85	70.83	**97.91**
WISDM 7 ↦ 30	89.32	84.79	91.26	89.64	90.61	86.40	93.85	83.20	**91.28**
WISDM 12 ↦17	71.53	70.48	79.86	73.26	70.20	65.97	77.08	70.17	**89.80**
WISDM 12 ↦19	54.29	51.01	51.77	55.30	51.51	49.24	47.47	47.47	**85.00**
WISDM 18 ↦20	83.74	88.62	64.23	75.20	85.36	83.74	81.30	76.01	**92.23**
WISDM 20 ↦30	67.96	77.02	81.88	74.76	71.84	72.49	21.28	82.85	**91.66**
WISDM 21 ↦31	21.29	46.58	54.62	31.32	54.41	49.97	44.45	52.61	**59.09**
WISDM 25 ↦29	26.11	44.33	53.89	57.78	60.04	35.00	74.79	53.89	**82.97**
WISDM 26 ↦2	82.52	83.33	77.44	87.20	66.46	**86.47**	74.95	83.29	83.50
WISDM Avg	65.78	70.05	70.80	69.79	69.62	67.04	66.97	70.66	**76.60**
WISDM Std of Avg	1.92	1.01	1.16	1.01	1.41	0.91	1.84	0.88	**0.73**
Sleep-EDF 0 ↦ 11	55.60	68.94	57.22	63.86	65.88	57.87	56.51	69.53	**74.41**
Sleep-EDF 2 ↦ 5	60.03	69.53	60.41	72.39	72.85	71.86	65.62	71.83	**73.76**
Sleep-EDF 12 ↦ 5	72.01	78.45	75.00	72.09	78.97	79.39	76.49	79.28	**79.81**
Sleep-EDF 7 ↦ 18	53.91	73.18	65.82	71.61	74.34	74.49	60.93	73.19	**75.32**
Sleep-EDF 16 ↦ 1	40.21	74.53	69.53	57.86	**81.82**	75.83	72.96	75.32	78.64
Sleep-EDF 9 ↦ 14	75.00	80.14	82.22	82.55	86.14	86.32	76.75	81.64	**87.17**
Sleep-EDF 4 ↦ 12	48.76	67.08	64.97	48.17	68.48	66.53	66.14	71.68	**69.86**
Sleep-EDF 10 ↦ 7	67.86	74.35	76.05	60.41	75.05	75.23	74.31	73.31	**77.23**
Sleep-EDF 6 ↦ 3	75.20	80.99	78.38	78.12	83.66	81.96	78.90	83.59	**84.58**
Sleep-EDF 8 ↦ 10	35.21	55.16	36.79	51.25	46.01	**65.70**	44.76	44.22	62.35
Sleep-EDF Avg	58.38	72.24	66.66	65.83	66.04	73.50	67.33	72.36	**76.31**
Sleep-EDF Std of Avg	1.33	**0.54**	1.16	1.69	0.99	0.34	0.89	1.03	0.87
Boiler 1 ↦ 2	57.09	67.93	67.13	67.42	68.13	68.93	72.43	75.74	**98.06**
Boiler 1 ↦ 3	74.54	94.98	93.32	94.02	94.88	95.36	96.14	97.32	**99.57**
Boiler 2 ↦ 1	73.14	85.96	84.32	84.32	87.76	88.74	89.32	90.23	**97.33**
Boiler 2 ↦ 3	66.09	93.32	91.53	92.89	92.62	91.31	91.53	92.89	**93.18**
Boiler 3 ↦ 1	74.99	93.89	92.43	93.01	93.14	93.92	94.77	95.32	**98.1**
Boiler 3 ↦ 2	61.31	63.32	60.39	57.93	60.43	60.43	70.62	72.32	**99.57**
Boiler Avg	65.86	83.23	81.45	81.59	82.77	83.03	85.69	87.21	**97.64**
Boiler Std of Avg	0.84	1.02	0.73	0.78	0.81	0.97	0.64	0.69	**0.51**

Higher is better. Best value in bold.

WISDM→WISDM, WISDM→HHAR, HHAR→WISDM. It can be observed that RAINCOAT consistently outperforms the baseline methods across all three UniDA settings considered in this work. The superiority of RAINCOAT is demonstrated in its ability to effectively handle the challenges associated with Universal Domain Adaptation and achieve improved performance compared to the baseline approaches.

C.6. Ablation Studies

Investigation of Loss Weights. To account for the different magnitudes of the loss terms in RAINCOAT, we employ weight balancing to ensure that the magnitudes of the loss terms are roughly comparable. We represent the overall loss as

Table 11. Macro-F1 for each dataset between various subjects. Shown: mean Accuracy over 5 independent runs.

Source ↦ Target	w/o UDA	CDAN	DeepCORAL	AdaMatch	DIRT-T	CLUDA	AdvSKM	CoDATS	RAINCOAT
HAR 2 ↦ 11	0.69	0.85	0.91	0.73	0.81	0.81	0.99	0.66	**1.00**
HAR 6 ↦ 23	0.63	0.88	0.81	0.81	0.68	0.92	0.87	0.71	**0.96**
HAR 7 ↦ 13	0.84	0.91	0.87	0.86	0.82	0.99	0.92	0.78	**1.00**
HAR 9 ↦ 18	0.17	0.61	0.44	0.55	0.58	0.67	0.73	0.60	**0.76**
HAR 12 ↦ 16	0.58	0.64	0.65	0.48	0.62	0.64	0.68	0.64	**0.86**
HAR 13 ↦ 19	0.91	0.97	0.95	0.94	0.99	0.94	0.93	0.93	**1.00**
HAR 18 ↦ 21	1.00	1.00	1.00	1.00	0.98	0.99	1.00	0.99	**1.00**
HAR 20 ↦ 6	0.94	0.95	0.95	0.84	0.92	**0.98**	0.84	0.65	0.94
HAR 23 ↦ 13	0.71	0.82	0.70	0.67	0.74	0.71	0.77	0.54	**0.86**
HAR 24 ↦ 12	0.84	0.92	0.88	0.70	0.93	**0.99**	0.96	0.81	0.94
HAR Avg	0.73	0.86	0.82	0.76	0.81	0.86	0.87	0.72	**0.93**
HAR Std of Avg	0.024	0.014	0.015	0.011	0.032	0.005	0.010	0.04	**0.005**
HHAR 0 ↦ 2	0.60	0.70	0.86	0.83	0.76	0.82	0.72	0.73	**0.87**
HHAR 1 ↦ 6	0.64	0.93	0.91	0.93	0.86	**0.94**	0.88	0.90	0.93
HHAR 2 ↦ 4	0.32	0.52	0.45	0.46	0.51	0.44	0.44	0.46	**0.59**
HHAR 4 ↦ 0	0.29	0.27	0.26	0.32	0.30	0.40	0.33	0.20	**0.45**
HHAR 4 ↦ 5	0.78	0.98	0.90	0.76	0.93	0.94	0.93	0.96	**0.98**
HHAR 5 ↦ 1	0.90	0.98	0.90	0.94	0.90	0.96	0.92	0.94	**0.98**
HHAR 5 ↦ 2	0.19	0.35	0.36	0.40	0.36	0.37	0.35	0.41	**0.41**
HHAR 7 ↦ 2	0.31	0.32	0.32	0.37	0.34	0.36	0.41	0.36	**0.44**
HHAR 7 ↦ 5	0.36	0.76	0.50	0.60	0.73	0.65	0.64	0.59	**0.86**
HHAR 8 ↦ 4	0.58	0.97	0.73	0.61	0.64	0.84	0.83	**0.95**	0.94
HHAR Avg	0.5	0.68	0.62	0.62	0.64	0.67	0.65	0.63	**0.75**
HHAR Std of Avg	0.022	0.013	0.007	0.013	0.023	0.008	**0.003**	0.006	0.004
WISDM 2 ↦ 32	0.68	**0.72**	0.71	0.59	0.65	0.64	0.61	0.66	0.68
WISDM 4 ↦ 15	0.52	0.44	0.42	0.54	0.41	0.61	0.55	0.41	**0.98**
WISDM 7 ↦ 30	0.77	0.70	0.85	0.76	0.78	0.81	0.84	0.75	**0.86**
WISDM 12 ↦17	0.53	0.50	0.67	0.67	0.56	0.59	0.53	0.62	**0.72**
WISDM 12 ↦19	0.36	0.31	0.35	0.38	0.39	0.41	0.35	0.37	**0.78**
WISDM 18 ↦20	0.81	0.87	0.63	0.66	0.67	0.70	0.71	0.76	**0.92**
WISDM 20 ↦30	0.56	0.64	0.67	0.54	0.65	0.70	0.61	0.72	**0.87**
WISDM 21 ↦31	0.10	0.31	0.27	0.16	0.28	0.27	0.28	0.30	**0.43**
WISDM 25 ↦29	0.15	0.23	0.25	0.24	0.21	0.26	0.28	0.30	**0.44**
WISDM 26 ↦2	0.69	0.71	0.64	0.74	0.54	0.75	0.55	0.70	**0.75**
WISDM Avg	0.52	0.54	0.52	0.54	0.54	0.57	0.55	0.56	**0.74**
WISDM Std of Avg	0.031	0.020	**0.006**	0.015	0.012	0.029	0.013	0.014	0.010
Sleep-EDF 0 ↦ 11	0.48	0.54	0.50	0.52	0.53	0.47	0.48	0.50	**0.54**
Sleep-EDF 2 ↦ 5	0.47	0.62	0.53	0.62	0.63	0.66	0.59	0.53	**0.65**
Sleep-EDF 12 ↦ 5	0.59	0.68	0.65	0.66	0.67	0.69	0.64	0.66	**0.70**
Sleep-EDF 7 ↦ 18	0.53	0.69	0.62	0.59	0.71	0.71	0.60	0.61	**0.72**
Sleep-EDF 16 ↦ 1	0.43	0.62	0.58	0.48	0.66	0.67	0.63	0.58	**0.70**
Sleep-EDF 9 ↦ 14	0.61	0.68	0.71	0.67	0.75	0.72	0.68	0.71	**0.76**
Sleep-EDF 4 ↦ 12	0.42	0.59	0.59	0.37	0.59	0.55	0.59	0.58	**0.62**
Sleep-EDF 10 ↦ 7	0.58	0.67	0.72	0.37	0.68	0.71	0.72	0.71	**0.73**
Sleep-EDF 6 ↦ 3	0.67	0.73	0.70	0.62	0.75	0.73	0.72	0.70	**0.75**
Sleep-EDF 8 ↦ 10	0.41	0.43	0.36	0.46	0.39	**0.65**	0.46	0.38	0.61
Sleep-EDF Avg	0.52	0.63	0.60	0.54	0.64	0.65	0.61	0.60	**0.68**
Sleep-EDF Std of Avg	0.026	0.005	0.015	0.004	0.005	0.007	**0.003**	0.012	0.008
Boiler 1 ↦ 2	0.52	0.63	0.63	0.64	0.65	0.68	0.73	0.73	**0.98**
Boiler 1 ↦ 3	0.74	0.95	0.93	0.94	0.95	0.95	0.96	0.97	**0.98**
Boiler 2 ↦ 1	0.70	0.81	0.83	0.83	0.85	0.86	0.88	0.91	**0.97**
Boiler 2 ↦ 3	0.60	0.91	0.90	0.91	0.91	0.90	0.90	0.91	**0.91**
Boiler 3 ↦ 1	0.70	0.94	0.90	0.93	0.92	0.94	0.94	0.95	**0.97**
Boiler 3 ↦ 2	0.55	0.59	0.60	0.54	0.61	0.58	0.69	0.70	**0.99**
Boiler Avg	0.635	0.80	0.80	0.80	0.82	0.82	0.85	0.86	**0.97**
Boiler Std of Avg	0.008	0.010	0.007	0.008	0.010	0.006	0.007	0.005	**0.005**

Higher value indicates better performance. Best value in bold.

$L = \lambda_1 \cdot L_1 + \lambda_2 \cdot L_2 + \lambda_3 \cdot L_3$. The weights λ_1, λ_2, and λ_3 are normalized such that their sum is equal to 1:

$$\lambda_1 = a/(a+b+c); \lambda_2 = b/(a+b+c); \lambda_3 = c/(a+b+c),$$

where a, b, and c are non-negative constants representing the desired relative importance of each loss term. To determine the optimal values of the weights λ for each dataset, we perform a grid search using an independent source-target transfer scenario. Subsequently, we conduct experiments using the obtained weights across all transfer scenarios. The results of these experiments are presented in Table 14, which displays the average prediction accuracy for the target domains in the HAR dataset (closed-set DA). By employing weight balancing and optimizing the weights, we ensure that each loss term contributes appropriately to the overall objective of RAINCOAT. This allows us to achieve better performance and more

Table 12. Accuracy of UniDA using WISDM, WISDM→HHAR, HHAR→WISDM, Shown: mean Accuracy over 5 independent runs. Closed-Set DA baselines are colored in blue.

Source ↦ Target	No. Tar Private Class	CLUDA	UAN	DANCE	OVANet	UniOT	RAINCOAT (A)	RAINCOAT (with A&C)
WISDM 3 ↦ 2	1	31.71	8.04	8.53	25.61	26.78	28.05	**28.05**
WISDM 3 ↦ 7	1	23.96	8.19	8.33	**34.38**	30.31	25.92	25.92
WISDM 13 ↦ 15	2	54.58	9.85	14.58	10.42	16.46	58.33	**64.58**
WISDM 14 ↦ 19	2	30.30	39.03	44.00	42.42	40.32	46.21	**53.78**
WISDM 27 ↦ 28	2	8.98	6.94	6.74	7.87	10.98	22.92	**53.70**
WISDM 1 ↦ 0	2	71.05	70.34	75.71	74.29	73.14	73.68	**82.57**
WISDM 1 ↦ 3	3	0.00	32.85	38.46	**61.54**	36.31	11.54	35.54
WISDM 10 ↦ 11	4	60.52	31.80	30.26	35.53	39.35	72.37	**76.36**
WISDM 22 ↦ 17	4	26.32	27.87	23.68	40.79	38.31	40.79	**48.16**
WISDM 27 ↦ 15	4	56.25	22.18	27.08	60.42	52.34	58.17	**66.42**
WISDM Avg		36.37	25.71	27.70	33.28	36.43	44.08	**53.51**
WISDM Std of Avg		1.05	2.09	1.95	**0.97**	1.25	1.06	1.41
W→H 4 ↦ 0	1	32.43	24.5	30.73	35.24	36.51	34.32	**44.14**
W→H 5 ↦ 1	1	20.32	31.0	15.32	26.31	28.14	27.94	**35.65**
W→H 6 ↦ 2	1	60.32	34.7	32.32	40.35	48.94	65.12	**69.01**
W→H 7 ↦ 3	1	51.84	21.10	36.84	39.46	50.35	55.10	**60.88**
W→H 17 ↦ 4	1	12.31	24.50	15.94	25.31	26.32	24.98	**28.41**
W→H 18 ↦ 5	1	35.85	26.60	29.65	36.14	33.46	35.70	**40.76**
W→H 19 ↦ 6	1	46.39	32.75	38.13	47.98	49.32	50.17	**54.76**
W→H 20 ↦ 7	1	62.32	39.83	42.90	58.11	60.31	64.98	**64.98**
W→H 23 ↦ 8	1	53.76	32.71	40.87	58.32	52.47	60.71	**62.84**
W→H Avg		37.55	29.74	39.06	40.80	42.87	46.55	**51.35**
W→H Std of Avg		**1.04**	1.38	1.98	1.65	1.74	1.31	1.22
H→W 0 ↦ 4	1	59.32	55.30	61.94	63.14	64.07	62.98	**64.84**
H→W 1 ↦ 5	1	56.17	50.33	58.10	60.14	61.46	60.94	**62.85**
H→W 2 ↦ 6	1	50.44	49.85	52.51	54.84	56.15	55.95	**57.11**
H→W 3 ↦ 7	1	52.21	53.01	55.91	55.71	58.91	56.42	**60.95**
H→W 4 ↦ 17	1	39.87	37.04	41.39	41.01	42.50	41.94	**44.95**
H→W 5 ↦ 18	1	47.72	47.80	50.35	51.87	52.22	49.95	**51.27**
H→W 6 ↦ 19	1	44.50	43.09	46.19	44.08	45.93	46.05	**51.86**
H→W 7 ↦ 20	1	50.92	54.01	59.85	61.35	61.06	47.00	**62.59**
H→W 8 ↦ 23	1	44.50	42.06	43.66	48.14	49.71	47.77	**52.64**
H→W Avg		44.47	48.05	52.22	53.36	54.67	52.11	**56.57**
H→W Std of Avg		1.31	1.39	1.21	0.94	1.05	**0.97**	1.08

Higher accuracy is better. Best value in bold.

effective adaptation in various transfer scenarios. The results in Table 14 demonstrate the impact of weight balancing and highlight the average prediction accuracy attained for the target domains.

Investigation of Sample Complexity. We conducted an investigation into the impact of varying the amount of labeled source data on the performance of RAINCOAT, along with several baseline methods. Specifically, we examined different proportions of labeled source data relative to the total source data (30%, 50%, 70%, and 100%) and evaluated the performance using prediction accuracy and F1 score on the target domain. The results of these experiments are presented in Table 15.The results demonstrate that RAINCOAT consistently outperforms the baseline methods across all sample sizes. Even when only a limited amount of labeled source data is available, RAINCOAT still achieves competitive performance, showcasing its robustness and effectiveness in scenarios with varying amounts of labeled source data. These findings provide valuable insights into the practical use of RAINCOAT, particularly in real-world situations where obtaining labeled data can be challenging or resource-intensive. The ability of RAINCOAT to leverage limited labeled source data and still achieve superior performance highlights its potential for practical applications and its capability to adapt well in settings where labeled data may be scarce.

D. Additional Discussion

We first describe the importance of describing the application scenarios and the necessity of the task cannot be ignored. the goal of RAINCOAT is to enhance the generalization of a machine learning model to an unlabeled target domain. The presence of feature and label shift between the source and target domains can lead to a decrease in model performance and accuracy. This emphasizes the need for Domain Adaptation techniques to improve the generalization and robustness of machine learning models in real-world scenarios. RAINCOAT addresses both closed-set and universal domain adaptation, catering to different application scenarios and requirements. In closed-set domain adaptation, the focus is on adapting the model to a specific target domain while considering a fixed set of known classes or labels. On the other hand, universal

Table 13. H-Score of UniDA using WISDM, WISDM→HHAR, HHAR→WISDM, Shown: mean Accuracy over 5 independent runs.

Source ↦ Target	No. Tar Private Class	UAN	DANCE	OVANet	UniOT	RAINCOAT
WISDM 3 ↦ 2	1	0	0	0.07	0.11	**0.51**
WISDM 3 ↦ 7	1	0	0	0.2	0.22	**0.52**
WISDM 13 ↦ 15	2	0	0.14	0.33	0.36	**0.50**
WISDM 14 ↦ 19	2	0.24	0.28	0.31	0.28	**0.55**
WISDM 27 ↦ 28	2	0.07	0.07	0.23	0.35	**0.59**
WISDM 1 ↦ 0	2	0.41	0.39	0.38	0.40	**0.43**
WISDM 1 ↦ 3	3	0.46	0.49	0.45	0.43	**0.51**
WISDM 10 ↦ 11	4	0	0	0.34	0.41	**0.53**
WISDM 22 ↦ 17	4	0.13	0	0.32	0.41	**0.52**
WISDM 27 ↦ 15	4	0.43	0.51	0.46	0.52	**0.57**
WISDM Avg		0.17	0.19	0.31	0.35	**0.52**
WISDM Std of Avg		0.04	0.05	0.04	0.05	**0.04**
W→H 4 ↦ 0	1	0	0.14	0.15	0.19	**0.49**
W→H 5 ↦ 1	1	0.24	0.22	0.25	0.28	**0.53**
W→H 6 ↦ 2	1	0.14	0.12	0.20	0.25	**0.55**
W→H 7 ↦ 3	1	0	0.15	0.04	0.14	**0.51**
W→H 17 ↦ 4	1	0.35	0.28	0.41	0.45	**0.57**
W→H 18 ↦ 5	1	0.20	0.27	0.29	0.32	**0.47**
W→H 19 ↦ 6	1	0.19	0.22	0.25	0.28	**0.51**
W→H 20 ↦ 7	1	0.11	0.17	0.35	0.41	**0.49**
W→H 23 ↦ 8	1	0.21	0.28	0.47	0.51	**0.57**
W→H Avg		0.16	0.21	0.24	0.28	**0.52**
W→H Std of Avg		0.03	0.02	0.03	0.02	**0.02**
H→W 0 ↦ 4	1	0.23	0.28	0.33	0.37	**0.45**
H→W 1 ↦ 5	1	0.19	0.31	0.38	0.42	**0.47**
H→W 2 ↦ 6	1	0.04	0.17	0.23	0.29	**0.39**
H→W 3 ↦ 7	1	0.25	0.32	0.34	0.40	**0.42**
H→W 4 ↦ 17	1	0.31	0.39	0.41	0.40	**0.51**
H→W 5 ↦ 18	1	0.28	0.34	0.37	0.36	**0.48**
H→W 6 ↦ 19	1	0.42	0.42	0.46	0.47	**0.49**
H→W 7 ↦ 20	1	0.39	0.41	0.41	0.44	**0.52**
H→W 8 ↦ 23	1	0.19	0.28	0.32	0.35	**0.46**
H→W Avg		0.26	0.32	0.36	0.39	**0.47**
H→W Std of Avg		0.05	0.05	0.03	0.04	**0.03**

Higher H-Score is better. Best value in bold.

domain adaptation expands the scope by handling the more challenging task of adapting to an unlabeled target domain that may contain unknown or novel classes. By addressing both closed-set and universal domain adaptation, RAINCOAT provides a versatile framework that can be applied in a wide range of scenarios

Closed-Set Domain Adaptation (Closed-set DA). Closed-set DA is the problem of adapting a machine learning model trained on a labeled source domain to perform well on an unlabeled target domain where the set of classes is known in advance. Mitigating the feature shift is a common goal in this problem, where the distribution of features in the source domain differs from that in the target domain. Below are several examples of applications in different domains where Closed-Set Domain Adaptation is necessary:

- Consider a time series classification task that aims to classify human activity based on accelerometer data. The distribution of features in accelerometer data collected during a weekday morning commute (source domain) may differ from that collected during a weekend hike (target domain). In this case, feature shift can occur due to changes in the distribution of features related to the user's movement patterns, such as walking speed, stride length, and acceleration profiles.
- Consider a speech recognition task; the acoustic features of speech signals may vary between different recording environments or speakers. For instance, a speech recognition model trained on speech data recorded in a quiet room (source domain) may have a different distribution of acoustic features when applied to speech data recorded in a noisy environment (target domain). In this case, feature shifts can occur due to changes in the distribution of features related to the background environment.

Universal Domain Adaptation (UniDA). In real-world applications, little information may be available on the feature or label distribution of the target domain. Private labels in either the source or target domain may exist, i.e., classes present in one domain but absent in the other. This means feature and label shifts exist between source and target domains. Universal Domain Adaptation refers to the problem of adapting a machine learning model to perform well on a target domain under

Table 14. Investigation on loss weights a, b, and c for UniDA on WISDM using 1D-CNN as encoder.

a for Cross Entropy	b for Sinkhorn	c for Reconstruction	Accuracy
1	0.1	0.9	79.82
1	0.2	0.8	80.54
1	0.4	0.6	84.75
1	0.6	0.4	86.37
1	0.8	0.2	88.66
1	**1**	**0.2**	**94.26**
1	1	0	92.66

Table 15. Comparison of accuracy and F1 score on the HAR dataset for different domain adaptation methods with varying percentages of available source samples.

	Accuracy				F1 Score			
% of $\mathcal{D}^s$	CDAN	DIRT-T	CLUDA	RAINCOAT	CDAN	DIRT-T	CLUDA	RAINCOAT
30%	67.46±0.67	69.28±2.16	73.17±1.53	**73.67±1.48**	0.63±0.016	0.64±0.015	0.69±0.007	**0.69±0.007**
50%	72.20±0.65	71.75±2.57	76.86±1.75	**77.75±1.56**	0.67±0.012	0.68±0.011	0.73±0.006	**0.75±0.007**
70%	79.66±0.72	78.75±1.57	80.86±1.35	**83.75±1.56**	0.77±0.015	0.76±0.016	0.79±0.005	**0.81±0.004**
100%	85.78±0.91	83.26±2.18	85.53±1.78	**91.43±1.32**	0.85±0.014	0.81±0.015	0.86±0.005	**0.91±0.005**

both feature and label shifts. UniDA allows machine learning models to generalize to new and diverse domains, improving their overall robustness and applicability in real-world scenarios. For example,

- Consider a time series classification task to identify driving behaviors based on data collected from a car's sensors. The labels (e.g., aggressive driving, normal driving, or cautious driving) may vary between different drivers, depending on drivers' driving style and data labeling methods. For example, the data from one driver (source domain) may record only aggressive and normal driving. In contrast, data from another driver (target domain) may record only normal and cautious driving due to differences in driving behaviors. In this case, the label shift can occur due to changes in the distribution of labels related to the driving habitats.
- Consider another time series EHR classification task where the goal is the prediction of hospital readmission. In the source domain, the labels could be defined as readmitted within 30 days, while in the target domain, the labels could be defined as readmitted within either 30 days or 60 days. This means the target domain has a different set of labels than the source domain, which could cause a label shift. In this case, the label shift could make it difficult for a machine learning model trained on the source domain to generalize well on the target domain.

In each of these examples, the feature and label shift between the source and target domains can decrease model performance and accuracy, highlighting the need for designing Domain Adaptation techniques to improve the generalization and robustness of machine learning models. Therefore, domain adaptation techniques like RAINCOAT could be applied to align both feature and label shifts between the two domains and improve the model's generalization performance on the target domain.

D.1. The Use of Frequency Features

It is important to consider the nature of the time series data when deciding whether frequency domain features are beneficial. In some cases, using frequency domain features may offer limited value, particularly when the data exhibits non-periodic or non-stationary patterns. For instance, in a time series dataset with a 2-way classification problem where both classes are driven by distinct temporal patterns at the same frequency rate, frequency features might not be as informative as time-based features. However, it is worth noting that RAINCOAT is specifically designed to jointly model both time and frequency features, allowing the model to prioritize learning time features when frequency features are less informative. Thus, in RAINCOAT, the potentially adverse effects of frequency features can be minimized due to the careful design of the time-frequency encoder.

It is essential to consider specific scenarios where the domain gap between the source and target domains is solely due to

Table 16. The comparison of accuracy (%) with other approaches on UCF-HMDB.

Dataset	HDMB to UCF	UCF to HDMB
DANN	76.4	75.3
TA3N	81.8	78.3
RAINCOAT	78.2	77.2

frequency changes. For example, in situations where the same data is collected using different experimental platforms that are calibrated differently in the source and target domains, methods that do not utilize frequency features may perform poorly. In RAINCOAT, we adopt a simple approach by concatenating time and frequency features to ensure fair comparisons with baseline methods. However, it is worth exploring whether incorporating a transformer architecture could further improve the performance of RAINCOAT over the existing time-frequency encoder. Transformers have demonstrated their effectiveness in capturing both local and global dependencies in sequential data, and their application to the time-frequency encoder in RAINCOAT could potentially yield additional improvements.

To summarize, while frequency domain features may offer limited value in certain scenarios, RAINCOAT is designed to handle such cases by allowing the model to prioritize time features when frequency features are less informative. Additionally, RAINCOAT offers flexibility in adapting to scenarios where frequency changes contribute more to the feature shifts. The exploration of transformer architectures within RAINCOAT presents an interesting direction for future research, as it may bring further improvements to the performance and adaptability of the model.

D.2. Extension to Video Domain Adaptation

In terms of interesting extensions, exploring the application of RAINCOAT to video data can provide a more comprehensive evaluation and broaden its scope. Videos are more complex data types compared to simple time series, as they incorporate both spatial and temporal features. This complexity introduces additional challenges, such as varying visual styles, lighting conditions, and camera viewpoints, which can significantly impact the performance of machine learning models.

One relevant work in the field is the Temporal Attentive Adversarial Adaptation Network (TA3N) developed by Chen et al. (Chen et al., 2019a). TA3N addresses video domain adaptation by simultaneously aligning and learning temporal dynamics without relying on sophisticated domain adaptation methods. It explicitly attends to temporal dynamics using domain discrepancy for effective domain alignment. Another notable framework is the unified framework for video domain adaptation presented by Kim et al. (Kim et al., 2021), which focuses on regularizing cross-modal and cross-domain feature representations, as well as feature spaces.

To evaluate RAINCOAT in the context of video domain adaptation, we conducted experiments on the publicly available benchmark dataset based on the UCF-HMDB benchmark, as assembled by Chen et al. (Chen et al., 2019b). This benchmark dataset consists of an overlapped subset of the original UCF and HMDB datasets, containing 3209 videos across 12 classes. We utilized the source code provided by the authors of TA3N (Chen et al., 2019a) and directly quoted the performance reported in their work. The results, shown in Table 16, highlight the effectiveness of RAINCOAT on the UCF-HMDB dataset, providing promising outcomes and serving as a solid foundation for further research in video domain adaptation.

Exploring video domain adaptation within the framework of RAINCOAT opens up new possibilities for addressing real-world challenges and enhancing the generalization and adaptability of machine learning models in video analysis tasks. This extension enables the consideration of both spatial and temporal features, contributing to more robust and accurate model performance in video domains. We plan to explore the efficacy of our RAINCOAT on video domain adaptation in future work by considering efficient feature extraction through tensor decomposition and acceleration algorithms (He et al., 2020; 2022; 2023; Cai et al., 2022).

D.3. Extension to Source-Free Domain Adaptation

Source-free domain adaptation attracts increasing attention because, in many real-world scenarios, collecting labeled data from the source domain may be expensive, time-consuming, or even impossible (Liu et al., 2021; Kundu et al., 2020; Yang et al., 2021; Xu et al., 2022). In such cases, source-free domain adaptation allows leveraging a pre-trained model from a different source domain to adapt to a target domain to adapt without using labeled data from the source domain. It is a

challenging and critical problem in machine learning, especially in computer vision tasks, where the source domain and target domain data have no overlap. SFDA aims to improve the performance of a model on a target domain with no access to any labeled data from the source domain. For example, (Liu et al., 2021) proposed a method that leverages the structure of the image to learn domain-invariant features for the target domain via pixel-and patch-level optimization objectives tailored for semantic segmentation. Another approach to SFDA is generalized SFDA (G-SFDA) (Yang et al., 2021), which aims to handle the more challenging case where the target domain contains multiple domains. G-SFDA proposed a method using a structural clustering algorithm to group the target domain data into clusters based on their feature similarity. They then trained a model on each cluster to handle the domain shift. Universal source-free domain adaptation (USFDA) (Kundu et al., 2020) is another variation of SFDA. It utilizes a novel instance-level weighting mechanism, source similarity metric (SSM), to handle both feature and label shifts. Recently, ATCoN (Xu et al., 2022) is proposed to address Source-Free Video Domain Adaptation by learning temporal consistency, guaranteed by two novel consistency objectives, namely feature consistency and source prediction consistency, performed across local temporal features.

Indeed, one can extend RAINCOAT to Source Free Domain Adaptation (SF-DA) by only modifying the pre-training stage. During the pre-training stage of RAINCOAT, the encoder G_{TF} is trained to learn well-separated, compact clusters of source domain data. This can be achieved by enforcing intra-class compactness and inter-class separability through negative classes, such as Triplet Loss. By doing so, the pre-trained model is better equipped for source-free deployment without prior knowledge of upcoming feature or label shifts. Once a pre-trained model is obtained, it can be adapted to a target domain using the two-stage algorithm proposed in RAINCOAT. In the first stage, the model encounters unlabeled target domain samples and obtains a target feature vector denoted as $\mathbf{z}^t_{before}$. The correction step then updates the encoder G_{TF} and decoder D_{TF} by solving a reconstruction task on target samples, which repositions the target features $\mathbf{z}^t_{before}$ into $\mathbf{z}^t_{after}$. According to the cluster assumption that input data is separated into clusters with samples within the same cluster having the same label, the corrected encoder maintains the features of common target samples close to their originally assigned label while allowing the features of target unknown samples to diverge from their assigned label. RAINCOAT can leverage this finding during deployment by detecting target unknown samples based on the movement of target features before and after the correction step. It assumes that if the distribution of the movement exhibits a bimodal structure, indicating the presence of unknown labels, it can easily detect private samples by training a 2-mean cluster while keeping common samples to their original assigned label.