Time-Frequency Decoupled Cross-Scale Partial Optimal Transport for Time Series Domain Adaptation

Shubin Liu[1], **Xinyang Chen**[1]✉, **Xiucheng Li**[1], **Weili Guan**[2], **Liqiang Nie**[1]
[1]School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
[2]School of Information Science and Technology, Harbin Institute of Technology (Shenzhen)
{liushb7, chenxinyang95, nieliqiang}@gmail.com, {lixiucheng, guanweili}@hit.edu.cn

Abstract

Unsupervised domain adaptation has become an important paradigm for mitigating distribution shift in time series classification. However, existing time series domain adaptation methods typically align source and target data at a fixed time scale and simply fuse time-frequency features before uniform alignment. Furthermore, they do not consider the effect of outliers on distribution alignment. To address these limitations, we propose CSPAN (Time-Frequency Decoupled **C**ross-**S**cale **P**artial Optimal Transport **A**lignment **N**etwork) for time series domain adaptation. CSPAN first constructs segment-scale time-frequency candidates, and then uses Segment-Scale Aware Top-k Fusion to select the most suitable segment-scale feature representation. It then performs cross-domain alignment after decoupling time and frequency through Temporal Partial Optimal Transport and Frequency Partial Optimal Transport, where temporal domain alignment is guided by the transferability estimated from a probing transport plan, and frequency domain alignment is guided by the discriminability estimated by an auxiliary frequency classifier. Combining segment-scale representation selection, time-frequency decoupling, and partial distribution alignment, CSPAN enables more flexible cross-domain matching. Extensive experiments on six datasets demonstrate CSPAN's consistent superiority, achieving an average accuracy improvement of 3.16% in cross-domain scenarios. Code is available at: https://github.com/liushb7/CSPAN.

1 Introduction

Time series classification has attracted great interest and made significant progress in a wide range of applications [11, 36, 41]. However, due to distributional shift in training and testing environments, performance degradation often occurs during actual model deployment [28, 40, 8, 39]. Unsupervised domain adaptation (UDA) [13, 22], which utilizes labeled data from the source domain and unlabeled data from the target domain for joint training, transfers knowledge from the source domain to the target domain, thereby enabling the model to generalize across domains. This may be a promising solution.

Early methods for time series domain adaptation (TSDA) primarily learn domain-invariant features after modeling temporal features. More recently, an increasing number of methods model both temporal and frequency features simultaneously [15, 19, 1, 16]. However, most of these methods simply fuse (e.g., concatenate) these two types of features and then perform a uniform cross-domain alignment, ignoring the differences in the distributions of temporal and frequency features. Secondly, existing methods typically perform distribution alignment on data at the same scale, failing to fully consider the inherent differences between individuals and devices. For example, in running, the stride frequency of different individuals may vary, making cross-domain alignment on data at a

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

uniform scale inappropriate. Furthermore, most approaches enforce full batch-wise participation in cross-domain alignment, neglecting the negative effect arising from class distribution mismatch within the batch and the presence of outliers.

In light of the above motivations, we propose CSPAN (Time-Frequency Decoupled **C**ross-**S**cale **P**artial Optimal Transport **A**lignment **N**etwork) for TSDA. Firstly, to select the most suitable feature scale representation for cross-domain alignment, and considering that local details of the time series are reflected in different segments [38], we propose **Segment-Scale Aware Top-k Fusion** to fuse several dominant segment-scale feature representations for subsequent alignment. Secondly, we highlight the differences in data distribution between the temporal and frequency subspaces, resulting in suboptimal alignment after fusion. Therefore, it is necessary to perform cross-domain alignment separately in the temporal and frequency domains. Thirdly, to mitigate the impact of outliers, we utilize Partial Optimal Transport [25] for cross-domain alignment, while simultaneously leveraging the characteristics of temporal and frequency features to enhance the alignment of the two branches. Based on points two and three, we propose **Temporal Partial Optimal Transport** and **Frequency Partial Optimal Transport** for selective sample alignment in the temporal and frequency subspaces, respectively.

Main contributions are as follows:

1. We discover cross-scale distribution alignment between the source and target domains in both the temporal and frequency subspaces, breaking the constraints of previous methods that required consistent distribution alignment scales and unified alignment through time-frequency feature fusion.

2. We propose CSPAN, which provides a new approach to TSDA through three collaborative modules: Segment-Scale Aware Top-k Fusion to adaptively select appropriate segment-scale feature representations; Temporal Partial Optimal Transport to enhance alignment in the temporal subspace; Frequency Partial Optimal Transport to promote distribution alignment in the frequency subspace.

3. CSPAN achieves average accuracy improvements of 3.16% over state-of-the-art methods across a wide range of time-series datasets with cross-domain scenarios.

2 Related Work

General Unsupervised Domain Adaptation Unsupervised domain adaptation (UDA) uses labeled source domain data to predict the label of a different but related unlabeled target domain. It has a wide range of applications [43, 14]. To achieve this goal, UDA methods aim to minimize domain discrepancy, thereby reducing the upper bound of the target error [3]. Existing UDA methods can be divided into three categories: (1) Methods based on adversarial training introduce a domain discriminator to distinguish source samples from target ones, while the feature extractor learns domain-invariant representations to deceive the domain discriminator. Advanced methods include DANN [13], CDAN [23], and DIRT-T [31]. (2) Methods based on statistical divergence aim to extract transferable features by minimizing statistical domain discrepancy in the latent feature space. Commonly used methods include DAN [22], DeepCoral [32], and HoMM [6]. (3) Methods based on self-training assign pseudo-labels to unlabeled target data and select samples with high confidence to combine with source samples in the next training iteration. Commonly used methods include PFAN [7], CST [18], and AdaMatch [4]. Overall, these methods are relatively universal in design. While these methods can be applied to time series with customized feature extractors, they often yield suboptimal performance due to neglecting the unique characteristics of time series.

Unsupervised Domain Adaptation for Time Series To date, unsupervised domain adaptation methods for time series remain limited. Early works focus on modeling features in the temporal subspace. VRADA [27] and CoDATS [37] consider suitable feature extractors based on temporal structure. SASA [5] adopts LSTM [30] to capture domain-invariant associations. AdvSKM [21] adapts MMD [33] to fit time series features. CLUDA [26] learns contextual representations through contrastive learning. Recently, some studies have emphasized the necessity of modeling features simultaneously in both the temporal and frequency subspaces for unsupervised domain adaptation. RAINCOAT [15] firstly introduces frequency features into domain adaptation and aligns temporal and frequency features separately using Sinkhorn divergence [1]. ACON [19] proposes mutual learning and adversarial learning in the time-frequency subspace. SSSS-TSA [1] proposes selective cross-domain alignment between channels. TransPL [16] improves pseudo-labeling through vector-

quantized transition matrices. TDVR [42] proposes variable calibration driven by transferability. EDEN [20] expands the raw dataset using multiple explicit domains. Despite significant progress, existing methods are still limited to fixed data scales and unified alignment of time-frequency features, and do not consider the negative effect of outliers during the alignment process, thus limiting the mitigation of domain shift.

3 Preliminaries

3.1 Problem Setup

In this paper, we study unsupervised domain adaptation (UDA) for time series classification. Let $\widehat{P} = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ denote the labeled source domain and $\widehat{Q} = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ denote the unlabeled target domain, where $\mathbf{x} \in \mathbb{R}^{C \times L}$ is a time series sample with C variables and length L. The source and target samples are drawn from two different distributions P and Q, respectively, while sharing the same label space $\mathcal{Y} = \{1, \ldots, K\}$. The goal of UDA is to learn a classifier that can achieve low prediction error on the target domain by using labeled source samples and unlabeled target samples.

Consistent with previous methods, we consider both the time and frequency domains of time series [28]. Given an original time series $\mathbf{x}$, its frequency domain representation can be obtained using the Fast Fourier Transform (FFT):

$$\mathbf{v} = \mathrm{FFT}(\mathbf{x}), \tag{1}$$

where complex variable $\mathbf{v} \in \mathbb{C}^{C \times \lfloor \frac{L}{2} \rfloor}$ contains observation of C variates over $\lfloor \frac{L}{2} \rfloor$ different frequencies, and this is due to the conjugate property of the frequency domain. In this paper, we use superscripts s and t to represent the source and target domains, respectively, and subscripts T and F to represent the temporal and frequency domain representations, respectively.

3.2 Theoretical Foundations of Optimal Transport

We adopt Optimal Transport (OT) [29] framework to align the source and target domains. Let $\mathbf{u}_s \in \mathbb{R}_+^{B_s}$ and $\mathbf{u}_t \in \mathbb{R}_+^{B_t}$ be probability vectors defined on the source and target domains, respectively, and let $\mathbf{M} \in \mathbb{R}_+^{B_s \times B_t}$ denote a cost matrix. The Kantorovich formulation [29] of OT seeks a transport plan $\mathbf{\Pi}$ that minimizes the total transport cost:

$$\mathrm{OT}(\mathbf{u}_s, \mathbf{u}_t) = \min_{\mathbf{\Pi} \in \mathcal{U}(\mathbf{u}_s, \mathbf{u}_t)} \langle \mathbf{\Pi}, \mathbf{M} \rangle, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product, and the transport polytope is defined as

$$\mathcal{U}(\mathbf{u}_s, \mathbf{u}_t) = \left\{ \mathbf{\Pi} \in \mathbb{R}_+^{B_s \times B_t} \mid \mathbf{\Pi} \mathbf{1}_{B_t} = \mathbf{u}_s,\ \mathbf{\Pi}^\top \mathbf{1}_{B_s} = \mathbf{u}_t \right\}. \tag{3}$$

Classical OT suffers from high computational costs on large-scale datasets, to improve efficiency, entropy regularization is commonly introduced, giving rise to the Sinkhorn divergence [1]:

$$\mathrm{OT}_\epsilon(\mathbf{u}_s, \mathbf{u}_t) = \min_{\mathbf{\Pi} \in \mathcal{U}(\mathbf{u}_s, \mathbf{u}_t)} \langle \mathbf{\Pi}, \mathbf{M} \rangle - \epsilon H(\mathbf{\Pi}), \tag{4}$$

where $H(\mathbf{\Pi}) = -\sum_{ij} \Pi_{ij} \log \Pi_{ij}$ is the entropy of the transport plan and ϵ controls the smoothness of the solution.

To mitigate mismatches, Partial Optimal Transport (POT) [25] relaxes the marginal constraints by allowing only a fraction of the total mass to be transported. Given a total transported mass $m \leq \min(\|\mathbf{u}_s\|_1, \|\mathbf{u}_t\|_1)$, the feasible set becomes

$$\mathcal{U}_m(\mathbf{u}_s, \mathbf{u}_t) = \left\{ \mathbf{\Pi} \in \mathbb{R}_+^{B_s \times B_t} \mid \mathbf{\Pi} \mathbf{1}_{B_t} \leq \mathbf{u}_s,\ \mathbf{\Pi}^\top \mathbf{1}_{B_s} \leq \mathbf{u}_t,\ \mathbf{1}_{B_s}^\top \mathbf{\Pi} \mathbf{1}_{B_t} = m \right\}. \tag{5}$$

In this work, we employ entropy-regularized partial optimal transport to achieve efficient and robust alignment between the source and target domains for time series data. The entropy-regularized POT problem is formulated as:

$$\mathrm{POT}_\epsilon(\mathbf{u}_s, \mathbf{u}_t) = \min_{\mathbf{\Pi} \in \mathcal{U}_m(\mathbf{u}_s, \mathbf{u}_t)} \langle \mathbf{\Pi}, \mathbf{M} \rangle - \epsilon H(\mathbf{\Pi}). \tag{6}$$

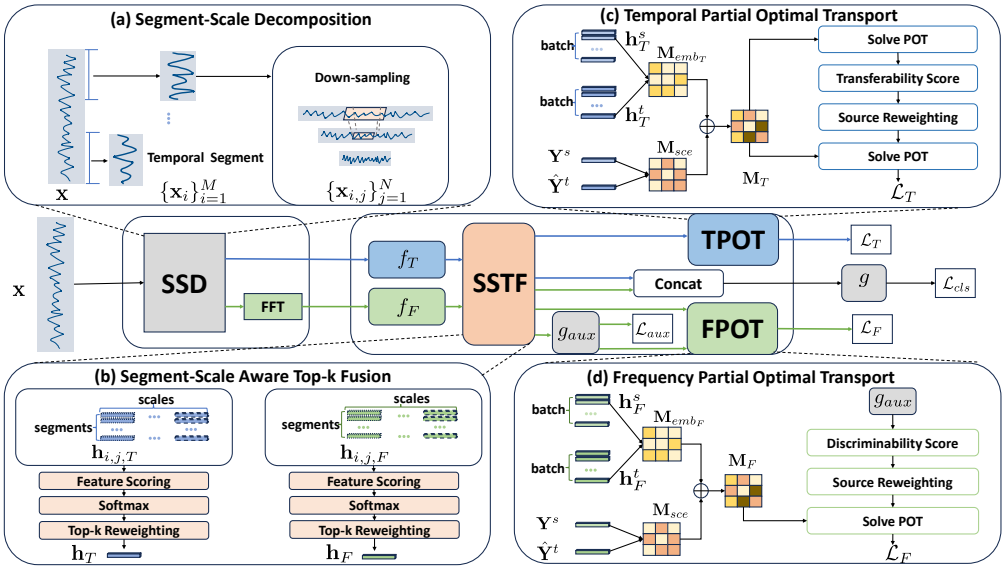


Figure 1: The architecture of CSPAN. (a) Segment-Scale Decomposition segments the input data and downsamples it to varying degrees. (b) Segment-Scale Aware Top-k Fusion extracts temporal and frequency representations through multi-segment and multi-scale, fusing the features using top-k reweighting in temporal domain and frequency domain, respectively. (c) Temporal domain partial optimal transport is performed with transferability-guided source reweighting. (d) Frequency domain partial optimal transport is performed with discriminability-guided source reweighting.

4 Approach

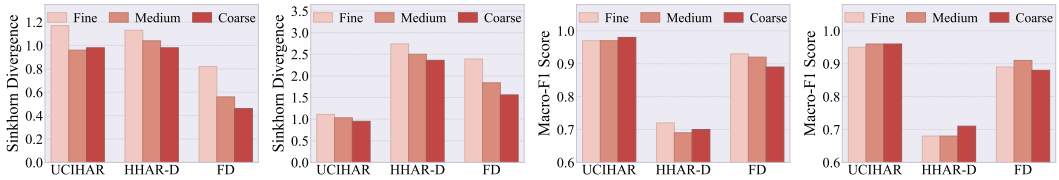
4.1 Method Overview

Figure 1 shows the overall structure of CSPAN, consisting of a temporal feature extractor f_T, a frequency feature extractor f_F, a classifier g, and a frequency auxiliary classifier g_{aux}. Specifically, **(1)** To adaptively select appropriate segment-scale feature representations, we propose Segment-Scale Aware Top-k Fusion (SSTF) in Section 4.2. **(2)** To decouple the temporal and frequency subspaces and mitigate the effect of outliers, and to utilize the transferability of temporal features, we propose Temporal Partial Optimal Transport (TPOT) in Section 4.3. **(3)** Combining the discriminability of frequency features, we propose Frequency Partial Optimal Transport (FPOT) in Section 4.4.

4.2 Segment-Scale Aware Top-k Fusion

Intuitively, coarse-scale time series typically reflect macroscopic trends, facilitating transfer but lacking discriminative detail, while fine-scale time series contain more local detail but are more difficult to transfer. Domain adaptation performance is determined by both transferability and discriminability [9]. Therefore, choosing an appropriate scale to achieve a balance between the two is expected to improve domain adaptation performance. To verify this idea, we investigate the impact of different-scale time series on domain adaptation. We use Sinkhorn Divergence (SD) [1] for cross domain alignment, and SD can also be used as an indicator to measure inter domain discrepancy, reflecting transferability. The smaller the SD, the smaller the inter domain discrepancy, indicating stronger transferability. In addition, we use Macro-F1 Score to reflect discriminability, and higher Macro-F1 Score means stronger discriminability. In Figure 2(a) and Figure 2(b), it's seen that in both the temporal and frequency subspaces, coarser scales correspond to smaller Sinkhorn Divergences, i.e., smaller inter-domain discrepancy. However, smaller inter-domain discrepancy does not necessarily mean better domain adaptation performance. As shown in Figure 2(c) and Figure 2(d), classification accuracy at coarse scales may be lower than at fine scales, meaning that transferability is enhanced at coarse scales, but discriminability may be reduced.

The above observation results indicate that if cross domain distribution alignment is only performed at a single scale, it may compromise between domain discrepancy and model performance. Therefore,



(a) SD in T-subspace (b) SD in F-subspace (c) Macro-F1 in T-subspace (d) Macro-F1 in F-subspace

Figure 2: Comparison of inter domain discrepancy (reflected by Sinkhorn Divergence) and model performance (reflected by Macro-F1 Score) at different scales.

we consider selecting data of appropriate scale to balance transferability and discriminability in the process of distribution alignment. In addition, considering that different segments of the same time series also exhibit varying degrees of details [38], we further adopt a segmentation strategy for the time series and propose Segment-Scale Aware Top-k Fusion, selectively fusing top-k dominant patterns from feature representations at different segments and scales, fully utilizing the segment-scale information of dominant patterns to achieve cross scale distribution alignment.

Given a time series $\mathbf{x} \in \mathbb{R}^{C \times L}$, we first divide it into M segments: $\{\mathbf{x}_i\}_{i=1}^{M}$. Then, we downsample each segment to N different degrees, obtaining $\{\mathbf{x}_{i,j}\}_{j=1}^{N}$. For the temporal domain, features are extracted using the corresponding temporal feature extractor for each segment-scale data: $\mathbf{h}_{i,j,T} = f_T^s(\mathbf{x}_{i,j})$. To select informative candidate features, we first score the features and then fuse the top-k highest-scoring candidate features as the final representation:

$$\begin{aligned} e_{i,j,T} &= q_T(\mathbf{h}_{i,j,T}), \quad \alpha_{i,j,T} = \mathrm{Softmax}(e_{i,j,T}), \\ \mathbf{h}_T &= \sum_{(i,j)\in\mathcal{K}_T} \bar{\alpha}_{i,j,T}\mathbf{h}_{i,j,T}, \end{aligned} \tag{7}$$

where $q_T(\cdot)$ is a two layer neural network used for feature scoring, $\mathcal{K}_T$ is the selected set of top-k temporal candidates, and $\bar{\alpha}_{i,j,T}$ is the renormalized weight on $\mathcal{K}_T$.

The frequency representation is obtained similarly. First, we compute the spectral representation using FFT: $\mathbf{v}_{i,j} = \mathrm{FFT}(\mathbf{x}_{i,j})$, and then obtain the frequency features $\mathbf{h}_{i,j,F} = f_F^s(\mathbf{v}_{i,j})$, where $f_F^s(\cdot)$ is the frequency encoder at the corresponding scale. After obtaining the frequency domain candidate features, we compute the frequency representation:

$$\mathbf{h}_F = \sum_{(i,j)\in\mathcal{K}_F} \bar{\alpha}_{i,j,F}\mathbf{h}_{i,j,F}, \tag{8}$$

where $\mathcal{K}_F$ is the selected set of top-k frequency candidates.

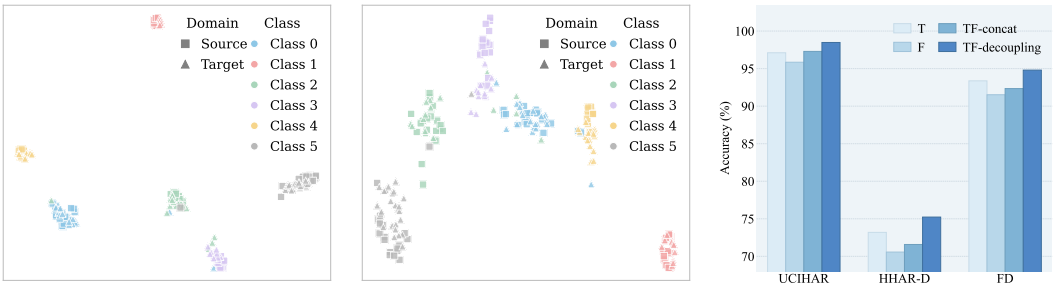
After obtaining the temporal domain and frequency domain features, we concatenate them and feed the fused representation into the classifier g for prediction, minimizing the cross-entropy loss on the labeled data in the source domain:

$$\mathcal{L}_{\mathrm{cls}} = \mathbb{E}_{(\mathbf{x}_i^s,\mathbf{y}_i^s)\sim\widehat{P}}\left[\ell\left(g(\mathrm{Concat}\left(\mathbf{h}_{T,i}^s, \mathbf{h}_{F,i}^s\right)), \mathbf{y}_i^s\right)\right], \tag{9}$$

where ℓ denotes cross-entropy loss.

4.3 Temporal Partial Optimal Transport

Within a domain, different categories often exhibit specific dominant frequencies. Models that focus on specific peak frequencies can more acutely distinguish category boundaries, showing better discriminability. When domain shift occurs, local frequency peaks in the frequency domain may shift drastically, while temporal features typically capture the overall pattern of the sequence, making them more robust across domains and exhibiting better transferability [19]. Figure 3(a) and Figure 3(b) visualize the temporal and frequency features of the cross domain scenario 5→11 in the HHAR-P dataset using t-SNE, respectively. From the figure, it's observed that there are significant differences in the distribution of temporal and frequency features. In the temporal subspace, the boundaries of the source and target domain data for each category are smaller compared to the frequency subspace,



(a) Temporal feature t-SNE (b) Frequency feature t-SNE (c) Performance in four situations

Figure 3: **(a)** t-SNE visualization of temporal features. **(b)** t-SNE visualization of frequency features. Different colors correspond to different categories, and squares and triangles correspond to source domain data and target domain data, respectively. **(c)** The classification accuracy of models in the target domain under four different cross domain alignment strategies.

indicating that the cross domain discrepancy of temporal features are smaller and have stronger transferability. However, many existing TSDA methods simply fuse (such as concatenate) these two features and perform a unified distribution alignment process, ignoring the potential negative impact of distribution differences between the two.

To validate the shortcomings of time-frequency coupling, we adopt four strategies for cross domain alignment on three datasets: using only temporal features, using only frequency features, concatenating and decoupling time-frequency features, and record the classification accuracy of the model in each case. The results are shown in Figure 3(c) . It's seen that the performance of concatenating time-frequency feature scenarios is a compromise between the first two scenarios. This indicates that simply fusing temporal and frequency features and aligning them uniformly may not be a good approach, as it does not achieve better results to meet our expectations.

Guided by these observations, and considering that existing methods usually require all samples within the source and target domain batches to participate in cross domain alignment during the training process, ignoring the potential negative impact of noise sample on the process, we decouple the cross domain alignment of time-frequency features via Temporal Partial Optimal Transport and Frequency Partial Optimal Transport, where the latter will be presented in Section 4.4. These two modules utilize the characteristics of temporal and frequency features (transferability of temporal features, discriminability of frequency features) and combine them with partial optimal transport to suppress noise sample influence and enhance cross domain alignment in temporal and frequency subspaces, respectively.

Specifically, for the temporal subspace, given mini-batches of source and target temporal features $\{\mathbf{h}^s_{T,i}\}_{i=1}^{B_s}$ and $\{\mathbf{h}^t_{T,j}\}_{j=1}^{B_t}$, we construct the temporal transport cost matrix $\mathbf{M}_T \in \mathbb{R}^{B_s \times B_t}$ by combining the embedding cost and the semantic cost [10]. The embedding cost matrix $\mathbf{M}_{\text{emb}_\text{T}}$ calculates the squared pairwise Euclidean distance between the source and target temporal features. The semantic cost matrix $\mathbf{M}_{\text{sce}}$ contains label information. The overall cost matrix is calculated as follows:

$$\begin{aligned} \mathbf{M}_T &= \eta_1 \mathbf{M}_{\text{emb}_\text{T}} + \eta_2 \mathbf{M}_{\text{sce}}, \\ (\mathbf{M}_{\text{emb}_\text{T}})_{ij} &= \|\mathbf{h}^s_{T,i} - \mathbf{h}^t_{T,j}\|_2^2, \\ (\mathbf{M}_{\text{sce}})_{ij} &= -\mathbf{y}_i^{s\top} \log(\mathbf{p}^t_j), \end{aligned} \tag{10}$$

where η_1 and η_2 are balance coefficients, $\mathbf{y}^s_i$ is the one-hot label vector of the i-th source sample, and $\mathbf{p}^t_j = g(\text{Concat}(\mathbf{h}^t_{T,j}, \mathbf{h}^t_{F,j}))$ is the predicted class probability vector of the j-th target sample.

To enhance temporal distribution alignment through transferability, we first compute a probing partial optimal transport plan using uniform source and target marginal distributions: $\mathbf{u}_s = \frac{1}{B_s}\mathbf{1}_{B_s}$ and $\mathbf{u}_t = \frac{1}{B_t}\mathbf{1}_{B_t}$. The probing plan is obtained by solving the entropy-regularized partial optimal transport problem:

$$\mathbf{\Pi}^{\text{probe}} = \underset{\mathbf{\Pi} \in \mathcal{U}_m(\mathbf{u}_s, \mathbf{u}_t)}{\arg\min} \langle \mathbf{\Pi}, \mathbf{M}_T \rangle - \epsilon H(\mathbf{\Pi}), \tag{11}$$

where $\langle \cdot, \cdot \rangle$ represents the Frobenius inner product, ϵ is the entropy regularization parameter, $H(\mathbf{\Pi}) = -\sum_{ij} \mathbf{\Pi}_{ij} \log \mathbf{\Pi}_{ij}$ is the entropy of the transport plan, and $\mathcal{U}_m(\mathbf{u}_s, \mathbf{u}_t) = \{\mathbf{\Pi} \in \mathbb{R}_+^{B_s \times B_t} \mid \mathbf{\Pi} \mathbf{1}_{B_t} \leq \mathbf{u}_s,\ \mathbf{\Pi}^\top \mathbf{1}_{B_s} \leq \mathbf{u}_t,\ \mathbf{1}_{B_s}^\top \mathbf{\Pi} \mathbf{1}_{B_t} = m\}$ is the partial transport polytope with total transported mass m.

The larger the sum of the i-th row of $\mathbf{\Pi}^{\text{probe}}$, the more likely the i-th source sample is to be transferable to the target-domain samples. Therefore, we use the row sum of $\mathbf{\Pi}^{\text{probe}}$ as the transferability score of the corresponding source-domain sample: $r_i = \sum_{j=1}^{B_t} \mathbf{\Pi}_{ij}^{\text{probe}}$. Then, we mix the normalized transferability distribution with the original uniform distribution:

$$\mathbf{u}_s^{\text{trans}} = (1 - \lambda_T)\mathbf{u}_s + \lambda_T \tilde{\mathbf{r}}, \tag{12}$$

where $\lambda_T \in [0, 1]$ controls the strength of transferability reweighting, and $\tilde{\mathbf{r}}$ is the normalized transferability vector.

Thus, the temporal distribution alignment loss is given by the following entropy-regularized partial optimal transport cost:

$$\mathcal{L}_T = \min_{\mathbf{\Pi} \in \mathcal{U}_m(\mathbf{u}_s^{\text{trans}}, \mathbf{u}_t)} \langle \mathbf{\Pi}, \mathbf{M}_T \rangle - \epsilon H(\mathbf{\Pi}). \tag{13}$$

4.4 Frequency Partial Optimal Transport

Unlike the temporal domain, frequency domain features exhibit better discriminability. Based on this, we propose Frequency Partial Optimal Transport, which introduces an auxiliary frequency classifier to estimate the discriminability of source domain samples and guide distribution alignment in the frequency domain.

Specifically, given the source and target frequency features: $\{\mathbf{h}_{F,i}^s\}_{i=1}^{B_s}$ and $\{\mathbf{h}_{F,j}^t\}_{j=1}^{B_t}$, we construct the frequency domain transport cost matrix $\mathbf{M}_F$ in a similar manner to the temporal domain. To evaluate discriminability, we introduce an auxiliary frequency classifier $g_{aux}(\cdot)$. For each source frequency feature, we calculate $\mathbf{p}_{F,i}^s = g_{aux}(\mathbf{h}_{F,i}^s)$, and g_{aux} is trained using the following loss:

$$\mathcal{L}_{\text{aux}} = \mathbb{E}_{(\mathbf{x}_i^s, \mathbf{y}_i^s) \sim \widehat{P}} \left[-\mathbf{y}_i^{s\top} \log(\mathbf{p}_{F,i}^s) \right]. \tag{14}$$

Discriminability scores are defined based on normalized prediction entropy:

$$H(\mathbf{p}_{F,i}^s) = -\sum_{c=1}^{K} p_{F,i,c}^s \log(p_{F,i,c}^s), \quad d_i = 1 - \frac{H(\mathbf{p}_{F,i}^s)}{\log K}, \tag{15}$$

where K is the number of classes. A larger d_i value indicates that the auxiliary frequency classifier is more confident in classifying the source sample, i.e., the sample contains stronger class discriminative information. After obtaining the normalized discriminability score, we mix the discriminability distribution and the uniform distribution:

$$\mathbf{u}_s^{\text{disc}} = (1 - \lambda_F)\mathbf{u}_s + \lambda_F \tilde{\mathbf{d}}, \tag{16}$$

where $\lambda_F \in [0, 1]$ controls the strength of the discriminability reweighting, and $\tilde{\mathbf{d}}$ is the normalized discriminability vector.

The frequency alignment loss is given by the transport cost of the entropy-regularized partial optimal transport problem as follows:

$$\mathcal{L}_F = \min_{\mathbf{\Pi} \in \mathcal{U}_m(\mathbf{u}_s^{\text{disc}}, \mathbf{u}_t)} \langle \mathbf{\Pi}, \mathbf{M}_F \rangle - \epsilon H(\mathbf{\Pi}). \tag{17}$$

4.5 Overall Training Objective

During training process, our method trains the temporal feature extractor f_T, the frequency feature extractor f_F, and the classifier g by minimizing the supervised classification loss $\mathcal{L}_{\text{cls}}$ in the labeled source domain, trains the auxiliary frequency domain classifier g_{aux} by minimizing the auxiliary classification loss $\mathcal{L}_{\text{aux}}$ of the frequency features in the source domain, and learns domain-invariant features by minimizing the partial optimal transport loss $\mathcal{L}_T$ in the temporal domain and $\mathcal{L}_F$ in the frequency domain, respectively. The overall training objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_T + \mathcal{L}_F + \beta \mathcal{L}_{\text{aux}}, \tag{18}$$

where β is a trade-off hyperparameter.

5 Experiments

5.1 Setup

Datasets We conduct extensive experiments using six benchmark datasets: (1) UCIHAR [2], HHAR-P [28], WISDM [17], and HHAR-D [12] for sensor-based human activity recognition. (2) FD [28] for machine fault diagnosis. (3) EMG [24] for gesture recognition. For each dataset, we evaluate 10 source-target pairs by referring to existing time series data domain partitioning methods [15, 4, 1]. Further processing details are provided in Appendix A.

Baselines (1) We report the performance of the model without UDA (Source-only) in the temporal domain to show the overall contribution of UDA methods. (2) We implement the following state-of-the-art baselines for TSDA: CODATS [37], AdvSKM [21], CLUDA [26], RAINCOAT [15], ACON [19], TransPL [16], SSSS-TSA [1] and TDVR [42]. (3) We additionally implement general UDA methods: CDAN [23], DeepCoral [32], AdaMatch [4], HoMM [6] and DIRT-T [31].

Implementation We adopt the implementation of AdaTime [28] as the benchmarking suites for domain adaptation on time series data. We use 1D-CNN as temporal feature extractor base and complex-valued linear as frequency feature extractor base. Evaluation is performed using classification accuracy and Macro-F1 score computed on the target test set. Each experiment is repeated 5 times with different random seeds. Detailed model architectures and hyperparameters are included in Appendix B.

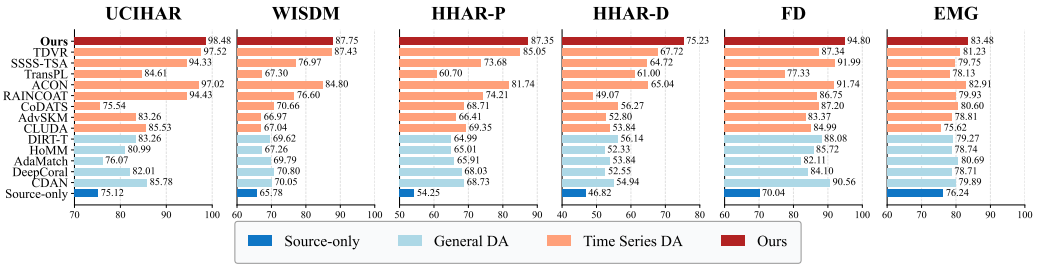


Figure 4: Average Accuracy (%) across multiple datasets.

Table 1: Accuracy (%) on EMG for unsupervised domain adaptation.

Method	0→1	0→2	0→3	1→2	1→3	2→0	2→1	2→3	3→1	3→2	Avg
Source-only	84.94	74.38	73.38	74.38	73.88	73.88	82.16	73.69	79.38	72.31	76.24
CDAN	87.84	76.63	77.63	77.44	81.63	73.94	87.10	75.13	83.98	77.63	79.89
DeepCoral	87.50	76.44	76.19	77.63	77.63	74.69	84.72	75.50	81.93	74.88	78.71
AdaMatch	89.03	75.94	79.38	76.94	80.00	76.31	89.94	**81.31**	84.26	73.81	80.69
HoMM	87.61	76.50	75.75	77.00	77.94	73.94	84.89	75.88	82.61	75.31	78.74
DIRT-T	89.77	75.25	78.69	75.88	80.06	70.63	84.77	77.69	83.30	76.69	79.27
CLUDA	78.18	75.00	76.75	74.75	74.19	75.94	79.43	70.00	76.88	75.13	75.62
AdvSKM	86.42	75.94	76.25	77.25	78.00	74.88	85.06	77.25	81.76	75.31	78.81
CoDATS	88.24	77.44	78.31	78.44	81.81	73.75	86.65	78.88	84.43	78.06	80.60
RAINCOAT	89.60	77.00	78.56	78.25	83.13	73.06	85.68	76.88	83.13	74.00	79.93
ACON	**92.50**	79.06	81.75	80.13	83.13	**77.94**	**90.91**	79.75	85.11	78.88	82.91
TransPL	86.37	76.48	75.88	77.13	77.26	74.21	83.30	75.67	81.36	73.66	78.13
SSSS-TSA	88.37	76.44	77.52	76.94	81.79	73.22	86.83	75.31	82.84	78.24	79.75
TDVR	90.21	76.03	80.13	78.22	82.13	76.31	90.23	80.24	83.78	75.13	81.23
Ours	92.15	**80.56**	**82.81**	**81.19**	**84.63**	77.75	90.28	80.13	**86.08**	**79.19**	**83.48**

5.2 Results

Figure 4 reports the average accuracy of 10 source-target domain pairs for 6 datasets and 3 tasks. The method proposed in this article consistently outperforms state-of-the-art baselines and has achieved

significant benefits on several datasets. Specifically, our model improved prediction accuracy by 2.70% on the HHAR-P dataset, 11.09% on the HHAR-D dataset, and 3.05% on the FD dataset, respectively. Due to page limitations, we report the results for 10 selected source-target domain pairs with metric accuracy on the EMG dataset in Table 1. Complete accuracy results for each dataset are shown in Table 9-13. The average Macro-F1 score is shown in Figure 9.

5.3 Analysis

Ablation Study We conduct a comprehensive ablation study in Table 2. We investigate the effectiveness of three modules of CSPAN and analyze the interaction between them. CSPAN's three modules show performance gains when added individually (Rows 2-4) and demonstrate mutual promotion in pairwise combinations (Rows 5-7). Full integration (Row 8) achieves optimal performance.

Table 2: Ablation study: Average Accuracy (%) on UCIHAR, HHAR-D and FD.

	SSTF	TPOT	FPOT	CSPAN	UCIHAR	HHAR-D	FD	Average
1	-	-	-	-	96.21	62.81	82.44	78.15
2	✓	-	-	-	96.76	64.12	85.47	82.12
3	-	✓	-	-	97.02	69.38	90.91	85.77
4	-	-	✓	-	96.43	67.19	86.15	83.26
5	✓	✓	-	-	98.04	74.94	93.30	88.76
6	✓	-	✓	-	96.89	69.08	87.43	84.47
7	-	✓	✓	-	97.91	71.40	91.77	87.03
8	-	-	-	✓	**98.48**	**75.23**	**94.80**	**89.50**

Computational Cost We compare the accuracy and training time of CSPAN with several recent methods on the HHAR-D dataset. Detailed results and analysis are provided in Appendix C.1. As shown in Table 5, CSPAN achieves a good balance between efficiency and performance.

Further Module Study To further investigate how each module affects model performance, we conduct additional studies on the three proposed modules to verify their effectiveness. Specifically, we examine the complementarity of temporal segment and multi-scale processing in SSTF, validate the effectiveness of TPOT via consecutive POT losses and entropy of transport matrices, and evaluate FPOT by comparing target accuracy and conditional entropy with and without discriminability enhancement. Detailed results and analysis are provided in Appendix C.2–C.4.

Sensitivity Analysis We perform sensitivity analysis on the trade-off parameters β for the frequency-domain auxiliary loss $\mathcal{L}_{\text{aux}}$, segment-scale selection parameter top-k, and transported mass m for partial optimal transport. Detailed results and analysis are provided in Appendix C.5.

6 Conclusion

In this paper, we propose CSPAN, a novel framework for TSDA based on time-frequency decoupled cross-scale partial optimal transport. CSPAN first breaks through the limitation of fixed-scale alignment by constructing segment-scale aware representations, enabling the model to capture transferable patterns at different temporal resolutions. Specifically, Segment-Scale Aware Top-k Fusion is introduced to adaptively select dominant segment-scale features; Temporal Partial Optimal Transport is proposed to enhance temporal-domain alignment by estimating source-side transferability from a probing transport plan; Frequency Partial Optimal Transport is designed to promote frequency-domain alignment with guidance from an auxiliary frequency classifier. Extensive experiments on multiple benchmark datasets demonstrate that CSPAN achieves consistent improvements over state-of-the-art TSDA methods.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (62676119, 62236003), Shenzhen College Stability Support Plan (GXWD20231130151329002), Shenzhen Science and Technology Program(KQTD20240729102207002, KQTD2025120109171700), Guangdong Science and Technology Program (2025B0101130003), Guangdong Provincial Key Laboratory of Artificial General Intelligence and Large Models (2026B1212010005).

References

[1] Nauman Ahad, Mark A. Davenport, and Eva L Dyer. Time series domain adaptation via channel-selective representation alignment. *TMLR*, 2025.

[2] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Esann*, 2013.

[3] Shai Ben-David, John Blitzer, Koby Crammer, and Alex Kulesza. A theory of learning from different domains. *Machine Learning*, 2010.

[4] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *ICLR*, 2021.

[5] Ruichu Cai, Jiawei Chen, Zijian Li, Wei Chen, Keli Zhang, Junjian Ye, Zhuozhang Li, Xiaoyan Yang, and Zhenjie Zhang. Time series domain adaptation via sparse associative structure alignment. In *AAAI*, 2021.

[6] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. Homm: Higher-order moment matching for unsupervised domain adaptation. In *AAAI*, 2020.

[7] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*, 2019.

[8] Xinyang Chen, Sinan Wan, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In *NeurIPS*, 2019.

[9] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, 2019.

[10] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, 2018.

[11] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 2019.

[12] Jean-Christophe Gagnon-Audet, Kartik Ahuja, Mohammad-Javad Darvishi-Bayazi, Pooneh Mousavi, Guillaume Dumas, and Irina Rish. Woods: Benchmarks for out-of-distribution generalization in time series. In *TMLR*, 2023.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[14] Xiaoqing Guo, Chen Yang, Baopu Li, and Yixuan Yuan. Metacorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation. In *CVPR*, 2021.

[15] Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. Domain adaptation for time series under feature and label shifts. In *ICML*, 2023.

[16] Jaeho Kim and Seulki Lee. Transpl: Vq-code transition matrices for pseudo-labeling of time series unsupervised domain adaptation. In *ICML*, 2025.

[17] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *ACM SIGKDD*, 2011.

[18] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. In *NeurIPS*, 2021.

[19] Mingyang Liu, Xinyang Chen, Yang Shu, Xiucheng Li, Weili Guan, and Liqiang Nie. Boosting transferability and discriminability for time series domain adaptation. In *NeurIPS*, 2024.

[20] Mingyang Liu, Xinyang Chen, Xiucheng Li, Weili Guan, and Liqiang Nie. Breakthrough sensor-limited single view: Towards implicit temporal dynamics for time series domain adaptation. In *NeurIPS*, 2025.

[21] Qiao Liu and Hui Xue. Adversarial spectral kernel matching for unsupervisedtime series domain adaptation. In *IJCAI*, 2021.

[22] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features withdeep adaptation networks. In *ICML*, 2015.

[23] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

[24] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. Out-of-distribution representation learning for time series classification. In *ICLR*, 2023.

[25] Khai Nguyen, Dang Nguyen, The-Anh Vu-Le, Tung Pham, and Nhat Ho. Improving mini-batch optimal transport via partial transportation. In *ICML*, 2022.

[26] Yilmazcan Ozyurt, Stefan Feuerriegel, and Ce Zhang. Contrastive learning for unsupervised domain adaptation of time series. In *ICLR*, 2022.

[27] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. In *ICLR*, 2021.

[28] 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. *ACM TKDD*, 2023.

[29] Bruno Levyand Erica Schwindt. Notions of optimal transport theory and how to implement them on a computer. In *arXiv preprint*, 2017.

[30] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. Convolutional lstmnetwork: A machine learning approach for precipitation nowcasting. In *NeurIPS*, 2015.

[31] Rui Shu, Hung H. Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *ICLR*, 2018.

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

[33] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. In *arXiv preprint*, 2014.

[34] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma andJames Y Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *ICLR*, 2024.

[35] Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *ICLR*, 2025.

[36] Yunshi Wen, Tengfei Ma, Tsui-Wei Weng, Lam M Nguyen, and Anak Agung Juliu. Abstracted shapes as tokens-a generalizable and interpretable model for time-series classification. In *NeurIPS*, 2024.

[37] Garrett Wilson, Janardhan Rao Doppa, and Diane J. Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *ACM SIGKDD*, 2020.

[38] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, , and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICML*, 2022.

[39] Tengxue Zhang, Yang Shu, Xinyang Chen, Yifei Long, Chenjuan Guo, and Bin Yang. Assessing pre-trained models for transfer learning through distribution of spectral components. In *AAAI*, 2025.

[40] Weiyang Zhang, Xinyang Chen, Xiucheng Li, Kehai Chen, Weili Guan, and Liqiang Nie. Unified transferability metrics for time series foundation models. In *NeurIPS*, 2025.

[41] Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 2023.

[42] Kexuan Zhou, Jiajing Geng, Boliang Hao, and Bailing Zhang. Transferability-driven variable recalibration for multivariate time series domain adaptation. In *ECAI*, 2025.

[43] Yang Zou, Zhiding Yu, B.V.K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.

A Dataset

A.1 Detailed Statistics

We conduct extensive experiments using a wide range of time series datasets. The detailed statistics for each dataset is included in Table 3. For UCIHAR, HHAR-P, WISDM and FD datasets, we use the processed versions released by AdaTime [28]. For HHAR-D datasets, we use the processed versions released by WOODS [12]. For EMG dataset, we use the processed version released by DIVERSIFY [24].

Table 3: Summary of datasets.

Dataset	Subjects	Channels	Length	Class	Total
UCIHAR	30	9	128	6	3290
WISDM	30	3	128	6	2070
HHAR-P	9	3	128	6	17934
HHAR-D	5	6	500	6	13674
FD	4	1	5120	3	10916
EMG	4	8	200	6	6883

A.2 Dataset Processing

For each dataset, the source and target domain data are randomly split into 80% as the training set and 20% as the test set. We follow Adatime [28] and use Z-score normalization for both the training and testing sets, using the following equation:

$$x_i^{\text{normalize}} = \frac{x_i - x^{\text{mean}}}{x^{\text{std}}}, \quad i = 1, 2, \ldots, N \tag{19}$$

where $N = N_s$ for the source domain data and $N = N_t$ for the target domain data. Note that both the training and testing splits are normalized based on the training set statistics only.

B Experimental Details

B.1 Model Architecture

Due to the fact that the method proposed in this paper involves the extraction of features at multiple scales, and data at different scales exhibit different lengths, simply using a single CNN or MLP to extract temporal and frequency features separately may not be appropriate. Recent foundational models tailored for multi-scale in time series analysis generally employ separate feed-forward layers per scale followed by feature fusion [34, 35]. Consistent with the architecture of the TSDA benchmark's architectures [28] that are based on 1D-CNN in temporal subspace and MLP in frequency subspace, we design corresponding temporal domain 1D-CNN and frequency domain MLP for different scales of data to extract features at different scales in a targeted manner.

B.2 Key Hyperparameters

We give some key hyperparameters under different data sets in Table 4. Among them, the batch size is uniformly set to 32, the default number of segments M and the number of scales N are 2 and 3 respectively. The number of iterations and learning rate take different values according to the size of the data set. For the selection of mass m, we refer to the approach in [25]. For different source domain-target domain pairs, we select the most appropriate transmission mass at intervals of 0.1 between 0.1 and 1.0. The transferability and discriminablity weighting parameters are set to 0.5 by default.

Table 4: Key hyperparameters for different datasets.

Hyperparameter	UCIHAR	HHAR-P	WISDM	HHAR-D	EMG	FD
Iteration	300	1200	300	1000	1000	1000
Batch Size	32	32	32	32	32	32
Learning Rate	0.01	0.001	0.01	0.001	0.001	0.01
Top-k	1	1	1	1	1	1
Regularization Parameter ϵ	0.5	0.5	0.5	0.5	0.5	0.2
Transferability Parameter λ_T	0.5	0.5	0.5	0.5	0.5	0.5
Discriminability Parameter λ_F	0.5	0.5	0.5	0.5	0.5	0.5

C Further Analysis

C.1 Computational Cost

Table 5 reports the training time and accuracy on the HHAR-D dataset. Despite the additional computational overhead introduced by segment-scale modeling, CSPAN achieves a state-of-the-art accuracy of 75.23%, significantly outperforming TDVR, ACON and RAINCOAT. Simultaneously, its training time remains competitive, faster than RAINCOAT and TDVR, and only slightly slower than ACON. These results show that CSPAN achieves a good balance between efficiency and performance.

Table 5: Training time and performance on HHAR-D dataset.

HHAR-D	RAINCOAT	ACON	TDVR	Ours
Accuracy(%)	49.07	65.04	67.72	75.23
Training Time	2.23h	1.17h	1.74h	1.53h

C.2 Further Analysis on Segment-Scale Aware Top-k Fusion

To further demonstrate the effectiveness of Segment-Scale Aware Top-k Fusion, we conduct additional experiments. We study and compare the performance of the model on the HHAR-D dataset under four different strategies: direct use of raw input data (R), only segmentation processing (S), only multi-scale downsampling (D), simultaneous segmentation and multi-scale processing (S+D), while keeping all other conditions consistent.

As shown in Table 6, it's seen that both segmentation and multi-scale downsampling improves the performance of the model compared to directly processing the raw data, and the best results is achieved when both processing methods are performed simultaneously. This indicates that these two processing methods complement each other in terms of functionality and jointly contribute to the effectiveness of Segment-Scale Aware Top-k Fusion.

In addition, in order to visually experience the cross scale transport phenomenon mentioned in Section 4.2, Figure 5 shows the proportion of temporal features at various scales for different classes in the cross-domain scene 1→6 on the HHAR-P dataset. In the experiment, the topk parameter is set to 1. It's seen that for class 2, the proportion of coarse-scale features is 100% in the source domain, but only 10% in the target domain, while the proportion of medium-scale features reaches 90%. This indicates that cross-scale transmission occurs in the temporal domain, which verifies the idea in our first module. Similar cross-scale transport phenomena also exist in the frequency domain, as shown in Figure 10.

C.3 Further Analysis on Temporal Partial Optimal Transport

The performance of the model after time-frequency coupling may reach a compromise between the temporal domain and frequency domain, which has been discussed in Section 4.3. We are now further studying how Temporal Partial Optimal Transport promotes cross domain alignment. In TPOT, we solve two POT problems. The probing matrix obtained from the first POT problem is transformed into transferability scores by summing rows, and the source domain weights are adjusted for the

Table 6: The impact of four different strategies on HHAR-D dataset.

HHAR-D	Accuracy(%)
R	70.16
S	72.57
D	73.82
S+D	75.23

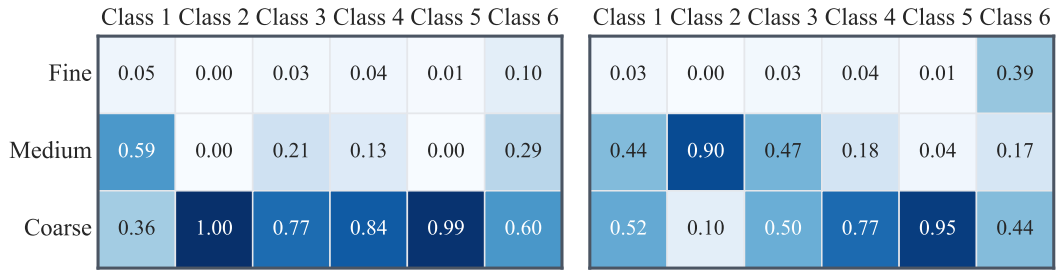


Figure 5: Visualization of temporal feature proportions at different scales for each category across scenario 1→6 in the HHAR-P dataset. **(Left)** source domain. **(Right)** target domain.

second POT problem. We study and compare the losses of two consecutive POT problems and the entropy of two transport matrices in the cross-domain scene 9→18 on the UCIHAR dataset. The results are shown in Table 7. It's seen that the entropy of the matrix obtained from the second POT has decreased compared to the first time. This is because after adjusting the source domain weights through the transferability score, the model identify the more easily transferable samples in the source domain and increase their weights, making the transmission mass set clearer for these samples and the transport plan clearer. Due to the stronger transferability of these samples, smaller differences from the target domain, and lower transmission costs, the loss of the second POT problem is ultimately lower, which means that enhancing transferability does indeed promote cross domain alignment.

Table 7: Comparison of loss and transport matrix entropy for two POTs in the cross-domain scene 9→18 on the UCIHAR dataset.

UCIHAR	POT loss	Entropy of transport matrix
First	0.232	5.180
Second	0.214	4.862

C.4 Further Analysis on Frequency Partial Optimal Transport

To verify the effectiveness of Frequency Partial Optimal Transport, we compare the model performance and predicted conditional entropy of the target domain before and after discriminability reweighting in the cross-domain scene 12→19 on the WISDM dataset. The experimental results are shown in Table 8. The experimental data show that after the enhancement of discriminability, the classification accuracy of the model in the target domain is improved, and the predicted conditional entropy is reduced. This indicates that after discriminability reweighting of the source domain, highly discriminative source domain samples transfer their rich discriminative information to the target domain, promoting the alignment of frequency features between the source and target domains and improving the discriminability on the target domain.

C.5 Sensitivity Analysis

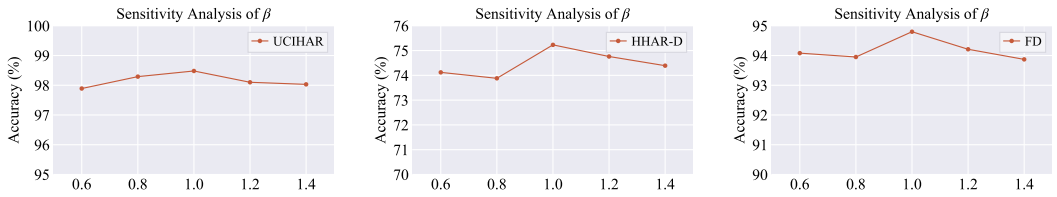
Sensitivity Analysis of β We investigate CSPAN's sensitivity to the trade-off parameters β for the frequency-domain auxiliary loss $\mathcal{L}_{\text{aux}}$. As shown in Figure 6, CSPAN remains stable within $0.6 \sim 1.4$, with the optimal average performance at the value of 1.

Table 8: Comparison of accuracy and conditional entropy before and after discriminability enhancement in the cross-domain scene 12→19 on the WISDM dataset.

WISDM	Accuracy(%)	Conditional entropy
Before	95.31	0.242
After	98.44	0.168

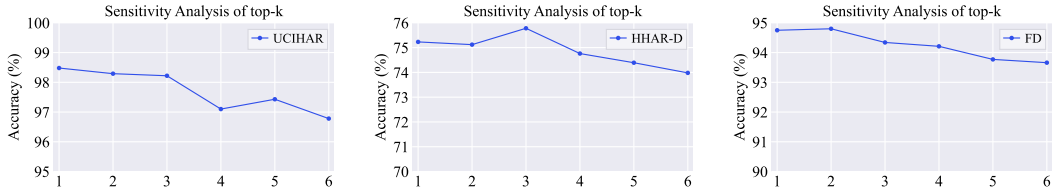
Sensitivity Analysis of top-k Different top-k values determine the number of segment scale features involved in transmission. As shown in Figure 7, when the top-k value is within a certain range (such as 1∼3), the dominant patterns with higher weights participate in distribution alignment, which can achieve better results in cross domain transmission. However, when the top-k value is too large, it is easy to introduce some noise features into the alignment process, which can have some negative effects. Therefore, in the experiment, we mainly set top-k to 1, which means that we usually choose the most representative dominant features for cross scale transmission.

Sensitivity Analysis of m Figure 8 shows the impact of different transport masses m on model performance. We conducted experiments under cross-domain scenarios using the following configurations: 23→13 for the UCIHAR dataset, 3→4 for the HHAR-D dataset, and 3→0 for the FD dataset. When m is within a certain range (such as $0.6 \sim 0.9$), the best results are achieved. When m is too small, there are too few samples involved in transmission, and the distribution alignment is not sufficient, so the performance is not good enough. It is worth noting that when using full mass distribution alignment (Optimal Transport, corresponding to $m = 1.0$), there is a significant difference in performance compared to partial optimal transport within the range of $0.6 \sim 0.9$. This indicates that partial optimal transport is able to alleviate the effect of outliers to a certain extent and enhance the alignment effect of the model.



(a) UCIHAR (b) HHAR-D (c) FD

Figure 6: Sensitivity analysis of the trade-off parameter β.

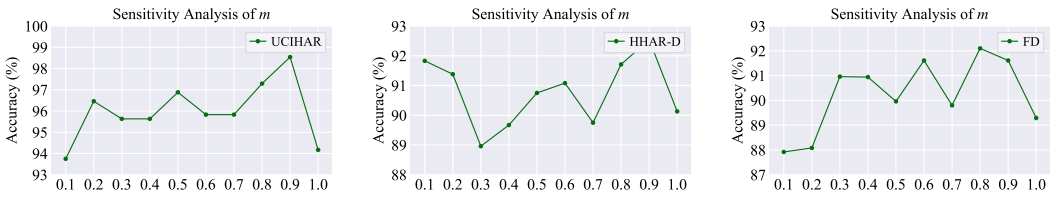


(a) UCIHAR (b) HHAR-D (c) FD

Figure 7: Sensitivity Analysis of the parameter top-k.

D Limitations

Although CSPAN achieves partial optimal transport alignment across scales in both time and frequency domains and demonstrates excellent performance on multiple datasets, it also faces a common problem of domain adaptation algorithms: it is still not stable enough on time series data with large variance and severe class imbalance, which is an urgent problem to be solved in the future.



(a) UCIHAR (b) HHAR-D (c) FD

Figure 8: Sensitivity Analysis of the transported mass m.

E Broader Impacts

We investigate unsupervised domain adaptive methods for time series classification by exploring cross-scale partial optimal transport alignment for time-frequency decoupling. This paper aims to advance the field of time series analysis and the real-world deployment of time series applications without any negative social impact.

F Full Results

We present the remaining experimental results in this appendix, including frequency-domain cross scale transmission visualization, Average Macro-F1 Score across multiple datasets, and accuracy tables under different cross domain scenarios on each dataset.

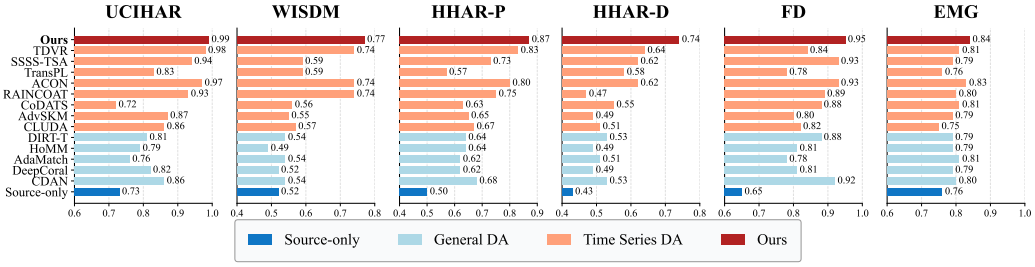


Figure 9: Average Macro-F1 Score across multiple datasets.



Figure 10: Visualization of frequency feature proportions at different scales for each category across scenario 1→6 in the HHAR-P dataset. **(Left)** source domain. **(Right)** target domain.

Table 9: Accuracy (%) on UCIHAR for unsupervised domain adaptation.

Method	**2→11**	**6→23**	**7→13**	**9→18**	**12→16**	**13→19**	**18→21**	**20→6**	**23→13**	**24→12**	**Avg**
Source-only	76.56	67.36	83.68	24.65	61.11	88.89	100.0	94.10	71.18	83.68	75.12
CDAN	85.42	87.50	92.01	58.86	66.67	96.52	100.0	95.13	82.64	93.40	85.78
DeepCoral	90.63	84.38	87.50	46.88	65.28	95.49	100.0	95.49	69.79	87.50	82.01
AdaMatch	75.00	80.20	85.76	56.59	49.65	94.79	100.0	84.37	68.75	70.83	76.07
HoMM	74.06	82.71	81.88	73.96	70.21	96.67	98.75	73.33	77.71	80.63	80.99
DIRT-T	80.21	74.31	82.99	59.03	67.01	99.30	98.61	92.36	74.72	94.27	83.26
CLUDA	81.77	92.01	99.31	67.71	65.28	94.44	98.96	97.22	72.92	99.31	85.53
AdvSKM	98.96	88.54	92.71	74.65	69.44	93.05	100.0	85.41	79.51	96.87	83.26
CoDATS	68.23	74.31	77.43	63.89	66.32	94.09	99.65	70.49	56.25	82.81	75.54
RAINCOAT	100.0	95.83	**100.0**	75.69	86.52	100.0	100.0	93.41	86.52	93.75	94.43
ACON	100.0	96.25	99.16	91.66	85.63	100.0	100.0	97.50	100.0	100.0	97.02
TransPL	85.94	87.29	89.58	70.63	85.41	86.67	94.38	71.76	82.50	92.08	84.61
SSSS-TSA	100.0	95.63	94.79	85.42	88.13	95.42	100.0	93.13	93.96	96.88	94.33
TDVR	100.0	**100.0**	98.96	86.46	91.04	100.0	100.0	98.75	**100.0**	**100.0**	97.52
Ours	**100.0**	99.58	99.58	**94.79**	**91.46**	**100.0**	**100.0**	**99.79**	99.79	99.79	**98.48**

Table 10: Accuracy (%) on WISDM for unsupervised domain adaptation.

Method	**2→32**	**4→15**	**7→30**	**12→7**	**12→19**	**18→20**	**20→30**	**21→31**	**25→29**	**26→2**	**Avg**
Source-only	81.16	79.86	89.32	71.53	54.29	83.74	67.96	21.29	26.11	82.52	65.78
CDAN	89.37	65.97	84.79	70.48	51.01	88.62	77.02	46.58	44.33	83.33	70.05
DeepCoral	87.92	62.50	91.26	79.86	51.77	64.23	81.88	54.62	53.89	77.44	70.80
AdaMatch	74.39	78.47	89.64	73.26	55.30	75.20	74.76	31.32	57.78	87.20	69.79
HoMM	77.10	74.58	78.64	68.13	50.61	71.22	72.82	56.39	57.00	66.10	67.26
DIRT-T	77.78	70.83	90.61	70.20	51.51	85.36	71.84	54.41	60.04	66.46	69.62
CLUDA	73.91	67.36	86.40	65.97	49.24	83.74	72.49	49.97	35.00	86.47	67.04
AdvSKM	70.83	95.85	93.85	77.08	47.47	81.30	21.28	44.45	74.79	74.95	66.97
CoDATS	77.29	70.83	83.20	70.17	47.47	76.01	82.85	52.61	53.89	83.29	70.66
RAINCOAT	79.71	**97.91**	91.28	89.80	85.00	**92.23**	91.66	59.09	**82.97**	83.50	76.60
ACON	89.86	86.25	98.06	98.13	77.73	83.66	91.26	63.61	60.00	99.51	84.80
TransPL	72.19	66.25	91.88	77.08	81.41	70.63	77.71	55.00	14.38	66.56	67.30
SSSS-TSA	80.63	68.13	95.00	76.88	60.47	79.69	93.94	71.25	60.63	83.13	76.97
TDVR	**90.31**	88.75	**98.96**	**99.17**	70.47	90.00	**93.96**	74.06	68.63	**100.0**	87.43
Ours	89.69	93.75	92.71	86.25	**94.69**	89.38	87.29	**76.25**	71.25	96.25	**87.75**

Table 11: Accuracy (%) on HHAR-P for unsupervised domain adaptation.

Method	**0→2**	**1→6**	**2→4**	**4→0**	**4→5**	**5→1**	**5→2**	**7→2**	**7→5**	**8→4**	**Avg**
Source-only	64.51	70.63	45.42	32.81	78.32	90.63	25.67	32.37	39.26	62.92	54.25
CDAN	76.19	92.57	52.57	29.09	97.27	96.16	35.04	37.05	75.26	96.11	68.73
DeepCoral	84.23	90.14	47.08	28.13	90.49	89.91	38.39	34.45	55.73	76.88	68.03
AdaMatch	84.78	92.31	54.50	36.45	78.45	94.20	41.96	37.65	63.80	64.69	65.91
HoMM	75.67	90.79	52.83	36.61	87.66	90.78	37.23	37.32	61.29	79.88	65.01
DIRT-T	77.83	88.54	50.69	32.22	93.16	91.86	38.62	38.10	72.46	65.83	64.99
CLUDA	79.84	93.40	45.90	38.84	94.08	95.57	33.93	37.80	77.57	96.52	69.35
AdvSKM	78.94	87.91	52.57	33.49	92.64	92.71	36.53	39.95	65.49	83.75	66.41
CoDATS	79.61	90.90	60.07	21.80	97.66	97.66	41.44	38.54	58.15	**97.01**	68.71
RAINCOAT	87.72	93.33	63.75	46.46	98.05	98.25	42.63	43.32	84.17	93.75	74.21
ACON	86.65	93.45	79.01	53.53	97.15	98.32	65.80	65.71	88.59	89.17	81.74
TransPL	60.49	60.17	60.46	43.04	74.96	90.82	50.40	47.50	58.44	60.75	60.70
SSSS-TSA	82.50	89.50	67.63	47.23	89.06	93.55	48.66	59.96	80.39	78.38	73.69
TDVR	**96.12**	**93.79**	81.83	70.09	98.01	**98.78**	**77.37**	66.21	90.80	73.05	85.05
Ours	85.71	93.54	**91.96**	**73.53**	**98.20**	98.09	69.38	**75.13**	**94.38**	93.58	**87.35**

Table 12: Accuracy (%) on FD for unsupervised domain adaptation.

Method	**0→1**	**0→2**	**0→3**	**1→0**	**1→2**	**2→0**	**2→1**	**2→3**	**3→0**	**3→2**	**Avg**
Source-only	62.21	53.71	62.41	63.91	73.95	64.08	93.17	95.54	57.08	74.31	70.04
CDAN	91.29	71.83	90.13	96.50	90.09	83.10	99.38	99.98	95.40	87.95	90.56
DeepCoral	75.54	71.79	76.03	89.13	83.55	76.34	98.84	98.55	87.50	83.71	84.10
AdaMatch	67.81	55.38	62.88	92.21	98.57	79.08	89.96	90.40	87.23	97.57	82.11
HoMM	81.54	71.63	78.17	89.89	84.78	76.03	98.71	99.55	90.94	85.96	85.72
DIRT-T	75.94	70.85	76.36	**98.10**	90.27	81.92	100.0	99.98	97.06	90.29	88.08
CLUDA	90.47	**82.63**	88.68	89.06	92.23	61.92	93.91	90.80	82.01	78.17	84.99
AdvSKM	74.71	66.05	73.30	87.86	86.29	76.85	98.66	99.38	84.89	85.74	83.37
CoDATS	81.79	73.26	83.15	89.22	88.68	81.43	99.89	100.0	85.47	89.00	87.20
RAINCOAT	85.18	79.40	89.04	78.84	90.11	81.43	95.18	96.81	77.39	94.08	86.75
ACON	86.52	69.00	86.96	97.92	99.80	84.29	98.62	98.93	**97.72**	97.66	91.74
TransPL	65.94	68.50	67.77	62.88	89.26	66.14	98.55	97.90	69.35	87.05	77.33
SSSS-TSA	**96.45**	83.75	92.46	92.86	89.44	**84.75**	100.0	**100.0**	93.19	87.05	92.00
TDVR	79.58	71.58	76.92	95.89	85.47	83.97	**100.0**	99.42	93.64	89.96	87.34
Ours	93.37	**86.65**	**94.82**	96.18	**99.91**	84.12	98.66	99.15	95.18	**100.0**	**94.80**

Table 13: Accuracy (%) on HHAR-D for unsupervised domain adaptation.

Method	**0→1**	**0→2**	**0→3**	**0→4**	**1→0**	**1→3**	**1→4**	**2→1**	**3→4**	**4→1**	**Avg**
Source-only	65.48	33.59	31.71	39.79	34.69	44.83	49.54	38.17	86.17	44.23	46.82
CDAN	69.86	48.28	38.22	48.42	48.75	60.48	51.33	47.84	87.33	48.89	54.94
DeepCoral	68.94	42.88	40.67	47.96	35.63	55.31	56.21	44.71	87.25	45.96	52.55
AdaMatch	71.78	39.60	39.74	47.50	52.50	55.48	58.33	46.49	85.83	41.15	53.84
HoMM	69.66	40.51	39.16	50.42	35.94	55.02	57.13	42.36	86.79	46.35	52.33
DIRT-T	68.37	42.14	47.21	52.92	41.25	54.81	55.63	46.73	92.25	54.81	56.14
CLUDA	71.78	39.60	39.74	47.50	52.50	55.48	58.33	46.49	85.83	41.15	53.84
AdvSKM	67.93	40.71	40.19	47.33	37.19	55.65	59.54	42.69	87.46	49.33	52.80
CoDATS	72.50	43.35	50.79	45.50	58.44	62.24	54.54	40.14	89.63	45.53	56.27
RAINCOAT	74.47	36.52	48.82	35.29	51.25	41.49	41.50	34.28	88.58	38.46	49.07
ACON	77.50	61.36	54.69	65.46	69.38	71.30	62.13	50.10	93.63	44.86	65.04
TransPL	75.43	55.74	53.68	57.71	66.56	52.33	55.96	48.41	94.54	49.66	61.00
SSSS-TSA	**84.66**	42.97	40.67	48.71	69.38	66.78	70.92	61.20	93.54	64.04	64.29
TDVR	79.95	55.29	58.27	52.88	**80.63**	71.66	68.75	52.07	**99.29**	58.41	67.72
Ours	81.06	**71.79**	**65.24**	**68.00**	74.69	**78.13**	**79.25**	**71.30**	92.54	**70.29**	**75.23**

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We include detailed information in Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are included in Appendix D.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theory assumptions are included in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include the detailed experimental settings in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at the anonymous link: https://anonymous.4open.science/r/CSPAN-35FB/.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We include detailed dataset descriptions in Appendix A and additional experimental mental settings in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results are included in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the experiments in this paper are conducted on a single NVIDIA GeForce RTX 4090 with 24GiB of memory.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: In every respect in the paper, we follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts is included in Appendix D.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All data, models, and code in the paper respect the license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.