A Weight-aware-based Multi-source Unsupervised Domain Adaptation Method for Human Motion Intention Recognition

Xiao-Yin Liu, Guotao Li*, Xiao-Hu Zhou, Xu Liang, Zeng-Guang Hou*, *Fellow, IEEE*

Abstract—**Accurate recognition of human motion intention (HMI) is beneficial for exoskeleton robots to improve the wearing comfort level and achieve natural human-robot interaction. A classifier trained on labeled source subjects (domains) performs poorly on unlabeled target subject since the difference in individual motor characteristics. The unsupervised domain adaptation (UDA) method has become an effective way to this problem. However, the labeled data are collected from multiple source subjects that might be different not only from the target subject but also from each other. The current UDA methods for HMI recognition ignore the difference between the each source subject, which reduces the classification accuracy. Therefore, this paper considers the differences between source subjects and develops a novel theory and algorithm for UDA to recognize HMI, where the margin disparity discrepancy (MDD) is extended to multi-source UDA theory and a novel weight-aware-based multi-source UDA algorithm (WMDD) is proposed. The source domain weight, which can be adjusted adaptively by the MDD between each source subject and target subject, is incorporated into UDA to measure the differences between source subjects. The developed multi-source UDA theory is theoretical and the generalization error on target subject is guaranteed. The theory can be transformed into an optimization problem for UDA, successfully bridging the gap between theory and algorithm. Moreover, a lightweight network is employed to guarantee the real-time of classification and the adversarial learning between feature generator and ensemble classifiers is utilized to further improve the generalization ability. The extensive experiments verify theoretical analysis and show that WMDD outperforms previous UDA methods on HMI recognition tasks.**

Index Terms—**Multi-source unsupervised domain adaptation; Human motion intention recognition; Generalization bound**

I. Introduction

This work was funded by the National Natural Science Foundation of China under (Grant 62103412, Grant U22A2056, and Grant 62373013), and the Beijing Natural Science Foundation under (Grant L222053 and L232021), and in part by sponsored by CIE-Tencent Robotics X Rhino-Bird Focused Research Program. (*Corresponding authors: Guotao Li and Zeng-Guang Hou).

Xiao-Yin Liu, Guotao Li and Xiao-Hu Zhou are with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China. (e-mail: liuxiaoyin2023@ia.ac.cn, guotao.li@ia.ac.cn, xiaohu.zhou@ia.ac.cn).

Xu Liang is with the School of Automation and Intelligence, Beijing Jiaotong University, Beijing 100044, China (e-mail: liangxu2013@ia.ac.cn).

Zeng-Guang Hou is with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with CASIA-MUST Joint Laboratory of Intelligence Science and Technology, Institute of Systems Engineering, Macau University of Science and Technology, Macao, China. (e-mail: zengguang.hou@ia.ac.cn).

Spinal cord injury, stroke and neuromotor impairment have seriously reduced the quality of human life. Exoskeleton robots, including rehabilitation robots [1]–[3] and power-assisted robots [4], [5], have become one of the important ways to treat and recover these diseases. To improve the wearing comfort level and achieve natural human-robot interaction, human motion intention (HMI) has gained great concerns in exoskeleton robots and human-robot interaction [6]–[8]. Accurate recognition of the HMI is beneficial for exoskeleton robots to improve the recovery effects.

HMI refers to the fusion of various biological signals (such as electroencephalography, electromyogram, etc.) and non-biological signals (such as speed, torque, etc.) to identify movement patterns. Li *et al.* [9] provided a systematic review of the HMI recognition research for exoskeleton robots. The current methods for HMI can be divided into two types: model-based and model-free. The model-based methods involve the kinematics model [10], the dynamic model [11], and the musculoskeletal model [12]. They establish the relationship between sensing signals and motion parameters to recognize HMI. The model-based methods are suitable for continuous HMI recognition. However, it consumes more time for the identification and calibration of parameters and is not robust when the task is complicated [12].

The model-free methods map the sensing signals into target HMI directly without parameter identification [13]–[15]. The above model-free methods can achieve good performance on specific subject but fail to perform well on cross-subjects. Due to the difference in individual motor characteristics, such as kinematic properties, most current methods require labeling a large amount of data and training specific classifiers for each individual, which is burdensome [9]. Unsupervised domain adaptation (UDA) is a common method to solve the above problem [16]–[19]. In this scenario, the source subjects (domains) data are labeled, and the target subject (domain) data are unlabeled. However, there is a discrepancy between the source and target domains. To alleviate the performance reduction caused by this discrepancy, single-source UDA (SUDA) utilizes the adversarial-based methods to align two domains [20]–[22] or explores different metric learning schemes to minimize this divergence [23]–[25].

In actual HMI recognition tasks, the labeled data are collected from multiple source subjects that might be different not only from the target subject but also from each other. However, the above UDA methods for HMI recognition don't consider the difference between the source subjects, which

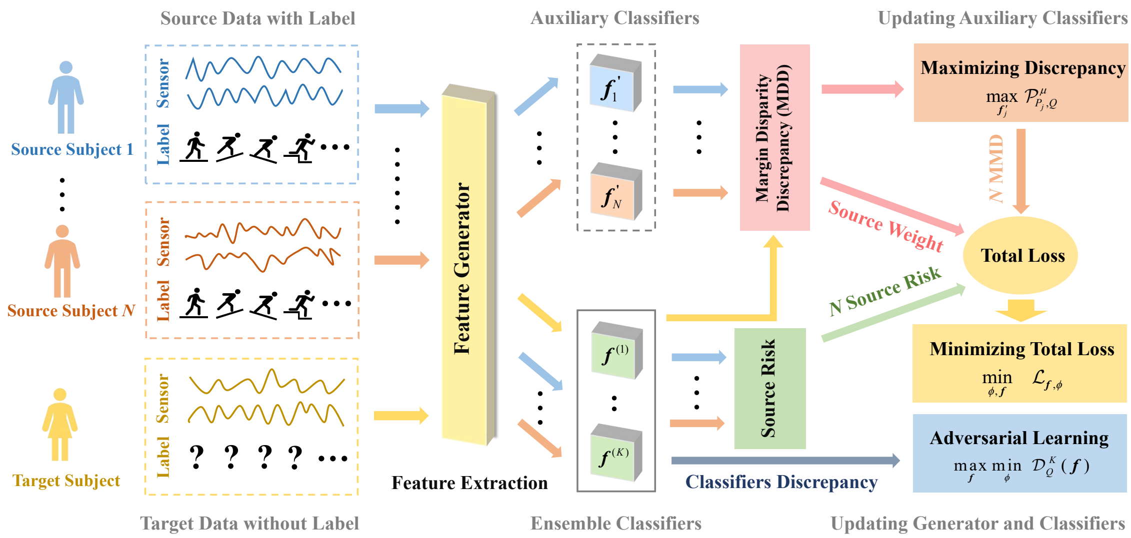

Fig. 1. The overview of the weighted multi-source unsupervised domain adaptation (WMDD) method for human motion intention recognition. The training data are from N source subjects with labels and one target subject without labels. The features of the sensor signal are extracted by feature generator ϕ. The final goal is to use feature generator and ensemble classifiers to classify the target subject intention accurately. Firstly, the j-th auxiliary classifier $\boldsymbol{f}'_j$ optimized by maximizing the discrepancy $\mathcal{P}$. Secondly, the N source domain weight $\boldsymbol{\alpha}$ is computed according to the estimated margin disparity discrepancy, and then the feature generator ϕ and K classifiers $\boldsymbol{f}$ are trained to minimize total loss $\mathcal{L}_{\boldsymbol{f},\phi}$. Then, motivated by adversarial learning, the classifiers and generator are optimized by minimizing and maximizing classifiers discrepancy.

may result in a sub-optimal solution to the problem and hinder the improvement of classifier performance [26]. Therefore, this paper aims to propose a multi-source UDA (MUDA) method, which considers the difference between the source subjects, for HMI recognition to further improve the classifying accuracy in the target subject. In MUDA field, there are following two key challenges that need to be addressed.

1) *How to accurately and effectively measure the discrepancy between source domain (subject) and target domain (subject)*?
2) Since the gap between the each source domain and target domain is different, *how to integrate these differences to improve the classification accuracy and generalization ability*?

For challenge 1, Jensen Shannon Divergence [27], Maximum Mean Discrepancy [28] and Wasserstein Distance [29] are widely employed to measure the discrepancy between source and target domains in SUDA. Zhang *et al.* [23] proposed Margin Disparity Discrepancy (MDD) method to measure this gap, which contains more generalization bound information. Zhang *et al.* [24] aggregated the absolute margin violations in multi-class classification and proposed Multi-Class Scoring Disagreement Discrepancy based on MDD. However, the above discrepancy-based methods are based on the theory of SUDA. Further researches are required on the MUDA theory for related discrepancies.

For challenge 2, the current MUDA methods are mainly constructed based on the distribution-weighted hypothesis, where the source domain is represented by a convex combination of multi-source domains [30]. Most scholars [31]–[34] calculated the source domain weight through the estimated discrepancy between source and target domains and optimized the weighted sum loss of multiple sources to improve the accuracy of classification. However, the weight learned by the above method lacks interpretability and cannot capture the complex relationship between domains [35], and there is a gap between the divergence in theories and algorithms for MUDA methods.

To solve the above problems, we extend the margin disparity discrepancy in SUDA theory to MUDA and propose a novel weighted MUDA algorithm (WMDD) motivated by the margin disparity discrepancy [23] and distribution-weighted hypothesis [30], where the new generalization bound of MUDA is proven and the new method for determining the source domain weight is given. Moreover, this paper theoretically and empirically validates that WMDD can learn diverse features and adapt well to unknown target subject in HMI recognition tasks. Fig. 1 shows the overview of the proposed WMDD. The contributions of this paper are given below.

1) A theory of MUDA is developed based on margin disparity discrepancy and a novel weight-aware-based MUDA algorithm is proposed for HMI recognition.
2) The MDD between each source domain and target domain is estimated by auxiliary classifiers, which can adjust the source domain weight adaptively.
3) A lightweight network is designed to guarantee the real-time of classification and adversarial learning is utilized to further improve the generalization ability.

The framework of this paper is as follows: Section II introduces the related works about HMI recognition and MUDA. Section III develops the new theory of MUDA, where the generalization bound for MUDA is proven. Section IV bridges the gap between MUDA theory and algorithm, and gives the detailed training steps of proposed method WMDD. Section

V presents the performance of WMDD on HMI recognition tasks and verifies the advantages of the designed mechanism. Section VI further discusses the multi-source UDA theory and algorithm. Section VII summarizes the entire work.

II. Related Works

This section presents a brief overview of the literature in the area of human motion intention (HMI) recognition and multi-source unsupervised domain adaptation (MUDA).

A. Human Motion Intention Recognition

Model-free methods based on deep networks, which can obtain higher-level features from sensor signals without domain-specific knowledge, have achieved great success in recent years, such as CNN [36], LSTM [37], CNN-BiLSTM [38]. However, the above methods are difficult to perform well on cross-subjects due to individual differences. Domain adaption is an effective method to cross-domain (subject) problems. Zhang *et al.* [18] incorporated an unsupervised cross-subject adaptation method to predict the HMI of the target subject without labels. Zhang *et al.* [16] proposed a novel non-adversarial cross-subject adaptation method for HMI recognition. The ensemble diverse hypotheses method was designed to mitigate the cross-subject divergence [17]. However, the above methods ignore the difference between each source subject, essentially a single-source UDA method, which may cause a sub-optimal solution to the problem. Few multi-source UDA studies have focused on recognizing HMI. Therefore, this paper aims to propose a novel multi-source UDA method for HMI recognition. *To the best of our knowledge, WMDD is the first work about multi-source UDA used for HMI recognition.*

B. Multi-source Unsupervised Domain Adaptation

Domain Adaptation Theory: Ben-David *et al.* [39], [40] conducted the pioneering theoretical works in domain adaptation field. They used the $\mathcal{H}\Delta\mathcal{H}$ divergence to replace the traditional distribution discrepancies and overcame the difficulties in estimation from finite samples. Mansour *et al.* [41] extended the zero-one loss of [39] to the general loss function of binary classification and developed a generalization theory. Kuroki *et al.* [42] proposed a more tractable source-guided discrepancy by fixing the hypothesis of [41] to the ideal source minimizer. Zhang *et al.* [23] extended the theory of [40] to multiple classes by introducing margin disparity discrepancy (MDD), which can characterize the difference of the multi-class scoring hypothesis. Zhang *et al.* [24] proposed multi-class scoring disagreement divergence based on MDD [23], that can characterize element-wise disagreements of multi-class scoring hypotheses by aggregating violations of absolute margin.

Multi-source Unsupervised Domain Adaptation Algorithm: Existing MUDA algorithms focused on aligning the distribution of each pair of source and target to reduce their domain shift by minimizing the combined discrepancy between source and target domains. Wen *et al.* [43] proved that the target loss is bounded by the weighted source loss and weighted discrepancy distance, and optimized networks by minimizing source loss and discrepancy. Yao *et al.* [44] quantified the importance of different source domains and aligned the source and target distributions by minimizing maximum mean discrepancy. Liu *et al.* [45] integrated multi-source domains into a single domain and aligned this distribution with the target distribution. Chen *et al.* [46] optimized the networks through minimizing the source loss and the Hellinger distance between source and target domains. Chen *et al.* [47] optimized the mixing weights and the network by minimizing the source mixture loss and the Pearson divergence.

Our work is different from the above ones in several key aspects. 1) The source domain weight is adaptively adjusted through the margin disparity discrepancy estimated by auxiliary classifiers instead of optimized constantly by solving unconstrained problem [47], which reduces the computation cost and enhances algorithm stability. 2) The theory and algorithm of WMDD are based on each single source distribution instead of the whole multi-source class-conditionals [44] or the whole multi-source joint distributions [46]. WMDD fully considers the differences between each source subject, which is beneficial for improving classification accuracy. 3) The distribution disparity metric is margin disparity discrepancy that contains more generalization bound information, rather than maximum mean discrepancy [44], the Hellinger distance [46], or Pearson divergence [47]. 4) A lightweight network is employed to guarantee the real-time of classification, instead of using multiple complex networks [43], [45] that is difficult to guarantee the real-time.

III. Theoretical Analysis of Multi-source Unsupervised Domain Adaptation

In this section, we give the basic notations and generalization bound for multi-source UDA based on the theory of single-source UDA. The theoretical analysis follows the previous work [23], where margin disparity discrepancy was used to measure the generalization bound.

A. Problem Formulation

Fig. 2 compares single-source unsupervised domain adaptation (SUDA) and multi-source unsupervised domain adaptation (MUDA). In MUDA, we consider $\mathcal{C}$ class, N source domains $\{\mathcal{D}_{sj}\}_{j=1}^{N}$ and one target domain $\mathcal{D}_t$ problem. There are two different but related distributions over $\mathcal{X}\times\mathcal{Y}$, namely the j-th source P_j and target Q, where $\mathcal{X}$ denotes input space and $\mathcal{Y}$ denotes output space, which is $\{0,1\}$ in binary classification. The learner is trained on labeled data $\mathcal{D}_{sj}=\{(\boldsymbol{x}_i^{sj},y_i^{sj})\}_{i=1}^{|\mathcal{D}_{sj}|}=(\boldsymbol{X}_{sj},\boldsymbol{Y}_{sj})$ sampled from source distribution P_j and unlabeled data $\mathcal{D}_t=\{\boldsymbol{x}_i^{t}\}_{i=1}^{|\mathcal{D}_t|}=\boldsymbol{X}_t$ sampled from target distribution Q. The goal of MUDA is to learn the hypothesis space $\mathcal{H}$ of labeling function $h:\mathcal{X}\rightarrow\mathcal{Y}$ to minimize expected target error

$$\mathcal{L}_Q(h)=\mathbb{E}_{(\boldsymbol{x},y)\sim Q}L\left[h(\boldsymbol{x}),y\right], \tag{1}$$

where L is the loss function. Ben-David *et al.* [40] used zero one loss of the form $\mathbb{I}[h(\boldsymbol{x})\neq y]$ to represent L, where $\mathbb{I}$ is the indicator function. Following [24], we consider the hypothesis

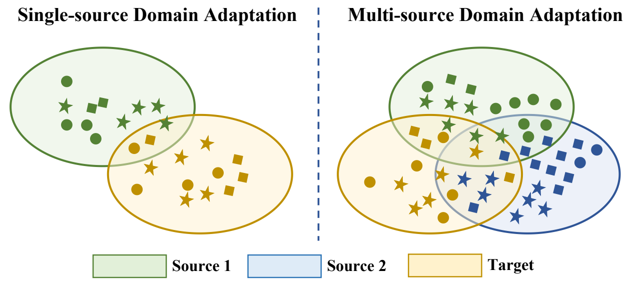

Fig. 2. The comparison of single-source unsupervised domain adaptation (SUDA) and multi-source unsupervised domain adaptation (MUDA). The source and target distributions of SUDA are not matched well. For MUDA, the target distribution hardly matches all subject distributions, and the discrepancy between the target distribution and each subject distribution might be different. Different shapes represent different data categories.

space $\mathcal{F}$ that contains scoring function $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|} = \mathbb{R}^{\mathcal{C}}$, where the outputs denote the prediction confidence. Then the *labeling* function can be induced

$$h_{\boldsymbol{f}}(\boldsymbol{x}) = \arg\max_{y\in\mathcal{Y}} f_y(\boldsymbol{x}), \tag{2}$$

where f_y denotes the y-th component of vector function $\boldsymbol{f}$. Zhang *et al.* [23] defined the *margin* of a hypothesis $\boldsymbol{f}$ at a labeled sample (x, y) as $\omega_{\boldsymbol{f}}(\boldsymbol{x}, y) = [f_y(\boldsymbol{x}) - \max_{y'\neq y} f_{y'}(\boldsymbol{x})]/2$. Then, the margin loss of $\boldsymbol{f}$ can be denoted as

$$\mathcal{L}_P^{\mu}(\boldsymbol{f}) = \mathbb{E}_{(\boldsymbol{x},y)\sim P}\Big\{\mathbb{W}_{\mu}\big[\omega_{\boldsymbol{f}}(\boldsymbol{x}, y)\big]\Big\}, \tag{3}$$

where $\mathbb{W}_{\mu}$ is ramp loss defined as

$$\mathbb{W}_{\mu}(v) := \begin{cases} 0, & \mu \leq v \\ 1 - v/\mu, & 0 < v < \mu. \\ 1, & v \leq 0 \end{cases} \tag{4}$$

Margin disparity discrepancy (MDD) [23], which contains more generalization information, is used to measure the distribution divergence between P and Q, defined as

$$d_{\boldsymbol{f}}^{\mu}(P, Q) = \sup_{\boldsymbol{f}'\in\mathcal{F}} \Big\{\mathbb{E}_Q\big[\mathbb{W}_{\mu}(\omega_{\boldsymbol{f}'})\big] - \mathbb{E}_P\big[\mathbb{W}_{\mu}(\omega_{\boldsymbol{f}'})\big]\Big\}, \tag{5}$$

where $\mathbb{E}_Q[\mathbb{W}_{\mu}(\omega_{\boldsymbol{f}'})] = \mathbb{E}_{\boldsymbol{x}\sim Q}\{\mathbb{W}_{\mu}[\omega_{\boldsymbol{f}'}(\boldsymbol{x}, h_{\boldsymbol{f}}(\boldsymbol{x}))]\}$. Thus the gap between the j-th source distribution P_j and target distribution Q for MUDA can be defined as $d_{\boldsymbol{f}}^{\mu}(P_j, Q)$. For single-source UDA, we have the following cross-domain generalization bound between source distribution P and target distribution Q:

Theorem 1: Fix $\mu > 0$. For any scoring function $\boldsymbol{f} \in \mathcal{F}$,

$$\mathcal{L}_Q(h_{\boldsymbol{f}}) \leq \mathcal{L}_P^{\mu}(\boldsymbol{f}) + d_{\boldsymbol{f}}^{\mu}(P, Q) + \lambda, \tag{6}$$

where the constant λ is the ideal combined margin loss, defined as $\lambda = \min_{\boldsymbol{f}\in\mathcal{F}}\{\mathcal{L}_P^{\mu}(\boldsymbol{f}) + \mathcal{L}_Q^{\mu}(\boldsymbol{f})\}$.

The proof for Theorem 1 can be found in Appendix A. In the following section, we give the cross-domain generalization bound for MUDA based on Theorem 1.

B. Generalization Bound for MUDA

In MUDA, some source domains, which are more relevant to target domain than the others, might be more important to classify. Therefore, we use domain weight $\alpha_j \geq 0$ to describe this and give following theorem.

Theorem 2: Fix $\mu > 0$ and the N source domain datasets $\{\boldsymbol{X}_j, \boldsymbol{Y}_j\}_{j=1}^N$. For any scoring function $\boldsymbol{f} \in \mathcal{F}$ and $\boldsymbol{\alpha} \in \Delta = \{\boldsymbol{\alpha} : \alpha_j \geq 0, \sum_j \alpha_j = 1\}$, the following holds:

$$\mathcal{L}_Q(h_{\boldsymbol{f}}) \leq \sum_{j=1}^{N} \alpha_j \Big(\mathcal{L}_{P_j}^{\mu}(\boldsymbol{f}) + d_{\boldsymbol{f}}^{\mu}(P_j, Q)\Big) + \beta, \tag{7}$$

where $\boldsymbol{\alpha}$ is the weight of N source domain, β is the weighted margin loss combination, $\beta = \min_{\boldsymbol{f}\in\mathcal{F}}\{\sum_{j=1}^N \alpha_j \mathcal{L}_{P_j}^{\mu}(\boldsymbol{f}) + \mathcal{L}_Q^{\mu}(\boldsymbol{f})\}$, which can be reduced to a rather small value if the hypothesis space is enough.

Remark 1: The proof of Theorem 2 can be found in Appendix B. Given the fixed β, the target error (generalization error) $\mathcal{L}_Q(h_{\boldsymbol{f}})$ is determined by the distribution discrepancy $d_{\boldsymbol{f}}^{\mu}(P_j, Q)$ between P_j and Q and the expected loss $\mathcal{L}_{P_j}^{\mu}(\boldsymbol{f})$ over the j-th source domain. In Eq. (7), the term $d_{\boldsymbol{f}}^{\mu}(P_j, Q)$ bounds the performance gap caused by domain shift. The smaller $\mathcal{L}_{P_j}^{\mu}(\boldsymbol{f})$ indicates better performance of scoring function $\boldsymbol{f}$ on the j-th source domain. Note that, the above theorem only considers the ideal and complete distribution. However, the collected data fails to cover the entire distribution space in actual learning.

Therefore, we further consider the sampling error and give the empirical form of Theorem 2. First, Rademacher complexity that is widely used in the generalization theory is introduced to measure the richness of a certain hypothesis space [23], [48]. The definition is given below:

Definition 1 (Empirical Rademacher complexity): Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $\mathbb{R}$ and $\widehat{\mathcal{S}} = \{z_1, \ldots, z_T\}$ be a fixed sample of size T drawn from the distribution $\mathcal{S}$ over $\mathcal{Z}$. The empirical Rademacher complexity of $\mathcal{G}$ for sample $\widehat{\mathcal{S}}$ is defined as

$$\widehat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{G}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{T}\sum_{t=1}^{T} \sigma_t g(z_t)\right], \tag{8}$$

where $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_T)$ are independent uniform random variables taking values in $\{-1, +1\}$.

Definition 2 (Induced Scoring Function Families): Given for a space $\mathcal{F}$ of scoring function $\boldsymbol{f}$, two induced scoring function families $\Omega_1(\mathcal{F})$ and $\Omega_2(\mathcal{F})$ are defined as

$$\Omega_1(\mathcal{F}) = \Big\{(\boldsymbol{x}, y) \to \boldsymbol{f}_y(\boldsymbol{x}) \mid \boldsymbol{f} \in \mathcal{F}\Big\}, \tag{9}$$

$$\Omega_2(\mathcal{F}) = \Big\{\boldsymbol{x} \to \boldsymbol{f}'_{h_{\boldsymbol{f}}(\boldsymbol{x})}(\boldsymbol{x}) \mid \boldsymbol{f} \in \mathcal{F}, \boldsymbol{f}' \in \mathcal{F}\Big\} \tag{10}$$

where $\Omega_1(\mathcal{F})$ is the union of projections of $\mathcal{F}$ onto each dimension, and $\Omega_2(\mathcal{F})$ can be seen as the space of inner products from $\mathcal{F}$ and $\mathcal{F}$. Based on the above definition, the generalization bound of MUDA is given below.

Theorem 3: Let $\widehat{P}_j$ and $\widehat{Q}$ be the corresponding empirical distributions of P_j and Q. $\widehat{\mathcal{D}}_{sj}$ and $\widehat{\mathcal{D}}_t$ are empirical datasets sampled from the i.i.d. samples $\mathcal{D}_{sj} = (\boldsymbol{X}_j, \boldsymbol{Y}_j)$ and $\mathcal{D}_t = \boldsymbol{X}_t$ of size m, respectively. Then, for any $\delta > 0$ and $\boldsymbol{\alpha} \in \Delta = \{\boldsymbol{\alpha} : \alpha_j \geq 0, \sum_j \alpha_j = 1\}$, with probability at least $1 - 3\delta$, the following holds for any scoring function $\boldsymbol{f} \in \mathcal{F}$,

$$\mathcal{L}_Q(h_{\boldsymbol{f}}) \leq \sum_{j=1}^{N} \alpha_j \Big\{ \mathcal{L}^{\mu}_{\widehat{P}_j}(\boldsymbol{f}) + d^{\mu}_{\boldsymbol{f}}\Big(\widehat{P}_j, \widehat{Q}\Big) \Big\} + 9\sqrt{\frac{\log(2/\delta)}{2m}} + \frac{\mathcal{C}}{\mu}\Big\{ \widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_s}(\Omega_1(\mathcal{F})) + \widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\Omega_2(\mathcal{F})) \Big\} + \beta,$$

where $\mathcal{C}$ denotes the number of classes, β is constant independent of $\boldsymbol{f}$, $\widehat{\mathcal{D}}_s$ and $\widehat{\mathcal{D}}$ are datasets sampled from the mixture distribution $\sum_{j=1}^{N} \alpha_j P_j$ and $Q + \sum_{j=1}^{N} \alpha_j P_j$, respectively.

Remark 2: The proof of Theorem 3 can be found in Appendix C. Theorem 3 indicates that the expected error on target domain $\mathcal{L}_Q(h_{\boldsymbol{f}})$ is bounded by empirical margin error on source domain $\mathcal{L}^{\mu}_{\widehat{P}_j}$, empirical distribution gap $d^{\mu}_{\boldsymbol{f}}(\widehat{P}_j, \widehat{S})$, the ideal error β and rademacher complexity terms. The better generalization ability can be achieved by selecting an appropriate margin μ for margin loss. For relatively larger μ and rich hypothesis space $\mathcal{F}$, the upper bound of target error becomes tighter.

IV. Connecting Theory and Algorithm

Motivated by the upper bound of the expected error given in Theorem 3, we bridge theory and algorithm for MUDA and develop a new theoretical-ground MUDA algorithm, named WMDD, and then the detailed training steps of WMDD is givenare in this section.

A. Theoretical Optimization Goal

Given hypothesis space $\mathcal{F}$ and samples $\mathcal{D}_{sj}, \mathcal{D}_t$, margin μ, the ideal error β and complexity terms are assumed to be fixed. Therefore, minimizing the target error $\mathcal{L}_Q(h_{\boldsymbol{f}})$ in Theorem 3 is equivalent to the below optimization problem.

$$\min_{\boldsymbol{f}} \quad \sum_{j=1}^{N} \alpha_j \Big\{ \mathcal{L}^{\mu}_{\widehat{P}_j}(\boldsymbol{f}) + d^{\mu}_{\boldsymbol{f}}\Big(\widehat{P}_j, \widehat{Q}\Big) \Big\}, \tag{11}$$

then by substituting Eqs. (3) and (5) into Eq. (11), we have

$$\begin{aligned} \min_{\boldsymbol{f}} \quad & \sum_{j=1}^{N} \alpha_j \Big\{ \mathbb{E}_{(\boldsymbol{x},y)\sim\widehat{P}_j}\left(\mathbb{W}_{\mu}\left[\omega_{\boldsymbol{f}}(\boldsymbol{x}, y)\right]\right) \\ & + \max_{\boldsymbol{f}'_j \in \mathcal{F}} \left(\mathbb{E}_{\widehat{Q}}\left[\mathbb{W}_{\mu}(\omega_{\boldsymbol{f}'_j})\right] - \mathbb{E}_{\widehat{P}_j}\left[\mathbb{W}_{\mu}(\omega_{\boldsymbol{f}'_j})\right]\right) \Big\}. \end{aligned} \tag{12}$$

The above optimization is a minmax game problem. This problem can be regarded as stackelberg game model, where auxiliary scoring function $\boldsymbol{f}'_j$ is set as a leader and $\boldsymbol{f}$ is a follower. The form of optimization is given below.

$$\begin{aligned} \min_{\boldsymbol{f} \in \mathcal{F}} \quad & \sum_{j=1}^{N} \alpha_j \Big\{ \mathcal{L}^{\mu}_{\widehat{P}_j}(\boldsymbol{f}) + \mathcal{P}^{\mu}_{\widehat{P}_j, \widehat{Q}}(\boldsymbol{f}, \boldsymbol{f}'_j) \Big\}, \\ \text{s.t.} \quad & \boldsymbol{f}'_j \in \arg\max_{\boldsymbol{f}' \in \mathcal{F}} \quad \mathcal{P}^{\mu}_{\widehat{P}_j, \widehat{Q}}(\boldsymbol{f}, \boldsymbol{f}'), \end{aligned} \tag{13}$$

where the maximization of $\mathcal{P}^{\mu}_{\widehat{P}_j, \widehat{Q}} = \mathbb{E}_{\widehat{Q}}(\mathbb{W}_{\mu}) - \mathbb{E}_{\widehat{P}_j}(\mathbb{W}_{\mu})$ is the estimated MDD. In the Stackelberg game, the auxiliary function $\boldsymbol{f}'_j$ is first updated, and then $\boldsymbol{f}$ is updated based on the result of $\boldsymbol{f}'_j$. The key of the next step is to determine the value of $\boldsymbol{\alpha}$. Intuitively, the source domain with a small gap to the target domain should have a large weight. Therefore, $\boldsymbol{\alpha}$ can be defined as

$$\alpha_j = \frac{\exp\big\{ - \big|\mathcal{P}^{\mu}_{\widehat{P}_j, \widehat{Q}}(\boldsymbol{f}, \boldsymbol{f}'_j)\big| \big\}}{\sum_{j=1}^{N} \exp\big\{ - \big|\mathcal{P}^{\mu}_{\widehat{P}_j, \widehat{Q}}(\boldsymbol{f}, \boldsymbol{f}'_j)\big| \big\}}. \tag{14}$$

The source domain weight $\boldsymbol{\alpha}$ can be adjusted dynamically by the estimated MDD, that is weight-aware-based. Then combining Eqs. (13) and (14), we can achieve final theoretical optimization goal of MUDA.

B. Bridging Theory and Algorithm

There is a gap between theory and algorithm since it is impractical to solve the above optimization problem directly. Firstly, considering the complexity of input space $\mathcal{X}$, we utilize the feature extractor $\phi(\cdot)$ to map the input space $\mathcal{X}$ into feature space $\Omega = \{\phi(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}\}$. Then label function can be again defined as $h_{\boldsymbol{f}} : \phi(\boldsymbol{x}) \to \arg\max_{y \in \mathcal{Y}} f_y(\phi(\boldsymbol{x}))$.

Secondly, since the ramp loss used in margin loss easily causes gradient vanishing, it is difficult to optimize Eq. (13) through stochastic gradient descent (SGD) [24]. Therefore, we want to find a surrogate function, denoted as $\mathcal{T}(\boldsymbol{f}(\phi(\boldsymbol{x})), y)$ here, for margin loss, which can be easily trained by SGD and keep the main property of the margin. Then, the two terms in Eq. (13) can be written as

$$\begin{aligned} \mathcal{L}^{\mu}_{\widehat{P}_j}(\boldsymbol{f}) =& \mathbb{E}_{(\boldsymbol{x},y)\sim\widehat{P}_j}\Big[\mathcal{T}\big(\boldsymbol{f}(\phi(\boldsymbol{x})), y\big)\Big] \\ \mathcal{P}^{\mu}_{\widehat{P}_j, \widehat{Q}}(\boldsymbol{f}, \boldsymbol{f}'_j) =& \mathbb{E}_{\boldsymbol{x}\sim\widehat{Q}}\Big[\mathcal{T}\big(\boldsymbol{f}(\phi(\boldsymbol{x})), h_{\boldsymbol{f}'_j}(\phi(\boldsymbol{x}))\big)\Big] - \\ & \gamma\mathbb{E}_{\boldsymbol{x}\sim\widehat{P}_j}\Big[\mathcal{T}\big(\boldsymbol{f}(\phi(\boldsymbol{x})), h_{\boldsymbol{f}'_j}(\phi(\boldsymbol{x}))\big)\Big], \end{aligned} \tag{15}$$

where γ is used to attain margin μ of margin loss similar to [23]. The cross-entropy loss is used to represent surrogate function $\mathcal{T}(\cdot, \cdot)$ following [23], [24]. On source subjects, the standard cross-entropy loss is employed, that is

$$\begin{aligned} \mathcal{T}\big(\boldsymbol{f}(\phi(\boldsymbol{x})), y\big) &= -\log\Big[\Theta_y\big(\boldsymbol{f}(\phi(\boldsymbol{x}))\big)\Big] \\ \mathcal{T}\big(\boldsymbol{f}(\phi(\boldsymbol{x})), h_{\boldsymbol{f}'_j}(\phi(\boldsymbol{x}))\big) &= -\log\Big[\Theta_{h_{\boldsymbol{f}'_j}}\big(\boldsymbol{f}(\phi(\boldsymbol{x}))\big)\Big], \end{aligned} \tag{16}$$

where $\Theta_y(\boldsymbol{f})$ is the softmax function, denoted as $\Theta_y(\boldsymbol{f}) = \exp(\boldsymbol{f}_y)/\sum_{j=1}^{N}\exp(\boldsymbol{f}_j)$. On the target subject, the modified cross-entropy loss is used to prevent gradient vanishing and exploding for adversarial learning [49]. Then, we have

$$\mathcal{T}\big(\boldsymbol{f}(\phi(\boldsymbol{x})), h_{\boldsymbol{f}'_j}(\phi(\boldsymbol{x}))\big) = \log\Big[1 - \Theta_{h_{\boldsymbol{f}'_j}}\big(\boldsymbol{f}(\phi(\boldsymbol{x}))\big)\Big]. \tag{17}$$

Therefore, the optimization problem (13) can be stated as two-stage optimization problem:

$$\begin{aligned} \max_{\boldsymbol{f}'_j} \quad \mathcal{P}^{\mu}_{\widehat{P}_j, \widehat{Q}} =& \mathbb{E}_{\boldsymbol{x}\sim\widehat{Q}}\Big\{ \log\Big[1 - \Theta_{h_{\boldsymbol{f}'_j}}\big(\boldsymbol{f}(\phi(\boldsymbol{x}))\big)\Big] \Big\} \\ & + \gamma\mathbb{E}_{\boldsymbol{x}\sim\widehat{P}_j}\Big\{ \log\Big[\Theta_{h_{\boldsymbol{f}'_j}}\big(\boldsymbol{f}(\phi(\boldsymbol{x}))\big)\Big] \Big\}, \end{aligned} \tag{18}$$

$$\min_{\phi, \boldsymbol{f}} \quad \mathcal{L}_{\boldsymbol{f}, \phi} = \sum_{j=1}^{N} \alpha_j \Big\{ \mathcal{L}^{\mu}_{\widehat{P}_j}(\boldsymbol{f}) + \mathcal{P}^{\mu}_{\widehat{P}_j, \widehat{Q}}(\boldsymbol{f}, \boldsymbol{f}'_j) \Big\}, \tag{19}$$

where $\mathcal{L}^{\mu}_{\widehat{P}_j}(\boldsymbol{f}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\widehat{P}_j}\left\{-\log\left[\Theta_y\big(\boldsymbol{f}(\phi(\boldsymbol{x}))\big)\right]\right\}$. During the iterative optimization process, the auxiliary function $\boldsymbol{f}'_j$ is firstly updated and then scoring function $\boldsymbol{f}$ and feature extractor ϕ are updated.

Theorem 4: For optimization problem defined in Eq. (18), fixing the classifier $\boldsymbol{f}$, then the estimated margin disparity discrepancy between source distribution P and target distribution Q is equivalent to

$$\gamma \log\gamma - (1+\gamma)\log(1+\gamma) + \gamma KL(P||Z) + KL(Q||Z),$$

where KL denotes the Kullback–Leibler divergence and $Z = (\gamma P + Q)/(\gamma+1)$ is the mixed distribution of P and Q.

Remark 3: The proof of Theorem 4 can be found in Appendix D. Since the Kullback–Leibler divergence between two distributions is always non-negative and zero only when they are equal, $\gamma\log\gamma - (1+\gamma)\log(1+\gamma)$ is the global minimum of MDD and that the only solution is $P = Q$. This indicates the different choices of γ do not result in the mismatch between source distribution P and target distribution Q. When the γ is fixed, the estimated MDD can effectively reflect the gap between P and Q.

C. Adversarial Learning Between Generator and Classifier

In practical implementation, feature extractor $\phi(\cdot)$ is approximated by a convolution neural network (CNN). The scoring function $\boldsymbol{f}$ and auxiliary function $\boldsymbol{f}'_j$ $(j = 1, ..., N)$ are approximated by a fully connected neural network (NN). The $\boldsymbol{f}$ and $\boldsymbol{f}'_j$ have the same network frame. In the following, we denote $\phi(\cdot)$ as the feature generator, $\boldsymbol{f}$ as the classifier, and $\boldsymbol{f}'_j$ as the auxiliary classifier.

Due to the high real-time requirements of human intent recognition in practical applications, a lightweight NN is used here. Zhang *et al.* [17] pointed out that lightweight NN may have a better generalization ability, but may not fit the training dataset as accurately as a large NN. The ensemble method is commonly used to solve the above problem since it can enhance weak learners to make precise predictions [50].

To trade off generalization ability and fitting ability, one tiny feature generator $\phi(\cdot)$ and K tiny classifiers $\boldsymbol{F} = \{\boldsymbol{f}^{(i)}\}_{i=1}^{K}$ are employed here. Then, the classification results can be represented as

$$y = \boldsymbol{M}\left\{h_{\boldsymbol{f}^{(i)}}\big(\phi(\boldsymbol{x})\big)\right\}_{i=1}^{K}, \tag{20}$$

where $\boldsymbol{M}$ denotes finding the mode from set. To further enhance the generalization of classifiers in ensemble learning, we firstly introduce classifier discrepancy $\mathcal{D}^{K}_{Q}(\boldsymbol{f})$, which measures the classification difference between K classifiers on target dataset Q, denoted as

$$\mathcal{D}^{K}_{Q}(\boldsymbol{f}) = \mathbb{E}_{\boldsymbol{x}\sim Q}\left\|\boldsymbol{F}\big(\phi(\boldsymbol{x})\big) - \mathrm{mean}(\boldsymbol{F})\right\|_1, \tag{21}$$

where $\boldsymbol{F}(\cdot) = \{\boldsymbol{f}^{(1)}(\cdot), ..., \boldsymbol{f}^{(K)}(\cdot)\}$ and $\|\cdot\|_1$ is L_1 norm. Then, motivated by generative adversarial network (GAN) [51], adversarial learning (two-player zero-sum game) is introduced to further update feature generator ϕ and classifiers $\{\boldsymbol{f}^{(i)}\}_{i=1}^{K}$, described as

$$\max_{\boldsymbol{f}}\min_{\phi}\ \mathcal{D}^{K}_{Q}(\boldsymbol{f}). \tag{22}$$

Algorithm 1: WMDD

Input: N source datasets $\{\mathcal{D}_{sj}\}_{j=1}^{N}$, one target dataset $\mathcal{D}_t$, feature generator ϕ, K classifiers $\{\boldsymbol{f}^{(i)}\}_{i=1}^{K}$ and N auxiliary classifiers $\{\boldsymbol{f}'_j\}_{j=1}^{N}$.
Initialization: Randomly initialize all networks.
for $t = 1, 2, \cdots, n_{\text{iter}}$ **do**
  Draw batch samples from target dataset $\mathcal{D}_t$.
  for $j = 1, 2, \cdots, N$ **do**
    Draw batch samples from j-th source dataset $\mathcal{D}_{sj}$.
    Update auxiliary classifier $\boldsymbol{f}'_j$ according to Eq. (18).
  end for
  Compute source weight $\boldsymbol{\alpha}$ according to Eq. (14).
  Update feature generator ϕ and classifiers $\{\boldsymbol{f}^{(i)}\}_{i=1}^{K}$ according to Eq. (19).
  Update classifiers $\{\boldsymbol{f}^{(i)}\}_{i=1}^{K}$ according to Eq. (23).
  Update feature generator ϕ according to Eq. (24).
end for

Since employing a classifier that is trained on the source domains with labels and the target domain without labels to predict target labels has high uncertainty, we maximize classifier discrepancy to improve "exploration ability" of ensemble classifiers on the target domain while ensuring the fitting ability on source domains. Therefore, the learning goal of ensemble classifiers in this stage can be denoted as

$$\min_{\boldsymbol{f}}\quad \mathcal{L}_{\boldsymbol{f}} = \mathcal{L}^{\mu}_{\widehat{P}}(\boldsymbol{f}) - \eta\mathcal{D}^{K}_{\widehat{Q}}(\boldsymbol{f}), \tag{23}$$

where η trades off fitting ability and generalization ability. The goal of feature generator ϕ is to maximize "universal ability" of extracted features through ϕ. That is minimizing classifier discrepancy when inputting extracted features into ensemble classifiers, that is

$$\min_{\phi}\quad \mathcal{L}_{\phi} = \mathcal{D}^{K}_{\widehat{Q}}(\boldsymbol{f}). \tag{24}$$

In adversarial learning (two-player game), the ensemble classifiers are updated firstly fixing the feature generator, and then the feature generator is trained. The fitting ability and generalization ability can achieve a balance through constant learning.

D. The Details of Algorithm

According to the above analysis, we propose a new **W**eighted multi-source UDA algorithm based on **M**argin **D**isparity **D**iscrepancy (WMDD). To further improve the generalization of lightweight networks, the idea of adversarial learning is introduced to training steps. The detailed training steps are given below.

- **Step 1:** The j-th auxiliary classifier $\boldsymbol{f}'_j$ is optimized to maximize discrepancy between j-th source distribution P_j and target distribution Q defined in Eq. (18).
- **Step 2:** The source domain weight $\boldsymbol{\alpha}$ is obtained by estimated MDD according to Eq. (14).
- **Step 3:** The feature generator ϕ and classifiers $\{\boldsymbol{f}^{(i)}\}_{i=1}^{K}$ are optimized to minimize generalization error (target source) $\mathcal{L}_{\boldsymbol{f},\phi}$ in Eq. (19).

TABLE I
BASE HYPERPARAMETERS OF WMDD.

Parameter	Value
Optimizer	Adam
Batch size	256
Learning rate	2×10^{-4}
Number of iterations n_{iter}	400
Number of ensemble classifiers K	5
Trade-off coefficient η	5
Margin coefficient γ	0.1

- **Step 4:** Fixing feature generator $\mathcal{L}_\phi$, then classifiers $\{\boldsymbol{f}^{(i)}\}_{i=1}^K$ are updated by maximizing classifier discrepancy through Eq. (23).
- **Step 5:** Fixing classifiers $\{\boldsymbol{f}^{(i)}\}_{i=1}^K$, then feature generator ϕ is updated by minimizing classifier discrepancy $\mathcal{L}_\phi$ according to Eq. (24).

Algorithm 1 gives the pseudo-code of the proposed method WMDD. The diagram of network updating can be found in Fig. 1. The final trained WMDD method can use a feature generator and ensemble classifiers to classify the target subject intention accurately.

V. EXPERIMENTS

This section gives the details of dataset and algorithm implementation, and focuses on answering the following questions through experiments:

- $\mathbf{Q_1}$: How does WMDD compare with current unsupervised domain adaptation methods for human motion recognition in standard benchmarks?
- $\mathbf{Q_2}$: How much does WMDD performance improve with source domain weight $\boldsymbol{\alpha}$ and adversarial learning between feature generator and classifier?
- $\mathbf{Q_3}$: How do hyperparameters margin coefficient γ, the number of classifiers K and trade-off coefficient η affect the performance of the WMDD algorithm?

A. Experimental Setup

Datasets: In this paper, two public datasets are utilized to answer the above questions. One is the encyclopedia of able-bodied bilateral lower limb loco-motor signals (ENABL3S) collected by Northwestern University [52]. The other is the daily and sports activities data set (DSADS) collected by Bilken University [53].

For the ENABL3S dataset [52], ten subjects are invited to walk on several terrains and switch locomotion modes between sitting, standing, level ground walking, stair ascent, stair descent, ramp ascent, and ramp descent. It contains 7 class motion intentions in total, that is the above 7 activities. Each subject is asked to repeat walking on a circuit ten times. ENABL3S contains filtered EMG, IMU and joint angle signals, which are segmented by a 300 ms wide sliding window, and the segmented signals are used to recognize human motion intention [17].

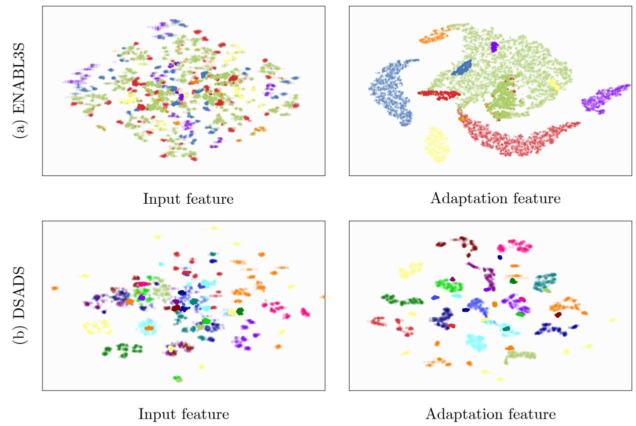

Fig. 3. The visualization of t-SNE projection of non-adapted input features and the hidden features adapted by the feature generator. All features are extracted from the training set for ENABL3S (a) and DSADS (b). The different color points represent different classes. The dark-color and light-color points denote the target and source features, respectively

For the DSADS dataset [53], eight subjects are asked to perform 19 activities, including sitting, standing, running, riding a bike, jumping, and playing basketball. The DSADS contains five 9-axis IMU signals and the captured signals are segmented by 5 wide sliding window segments. Since there is no transition between different activities, the DASDS dataset is only utilized to classify the human motion modes [18].

Experimental Details: In MUDA experiments, a target subject (domain) is selected first and then the remaining subjects are used as source subjects. ENABL3S and DSADS datasets contain 22,000 and 9000 signal segments, respectively. The data from each subject are randomly shuffled and divided into a training set (70%) and a test set (30%). Table I gives the base hyper-parameters of the WMDD method. The learning rate is set as 2×10^{-4}, the optimizer is Adam, the number of iterations is 400 and the batch size is set as 256. The margin coefficient γ is set to 0.1. Note that WMDD sets the same parameters for ENABL3S and DSADS datasets, which is different from previous methods that need setting different parameters for different datasets. This is also the advantage of the proposed method WMDD.

The network framework information is as follows. The network is composed of one feature generator, K classifiers and N auxiliary classifiers, where the feature generator contains 3 convolution layers and the classifier includes 3 fully connected layers. The performance of WMDD has a close connection with domain weight $\boldsymbol{\alpha}$, number of ensemble classifiers K and margin coefficient γ. The related experiments can be found in part C in this section. The code for WMDD is available at github.com/xiaoyinliu0714/WMDD. The code is run under an Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz and an NVIDIA GeForce RTX 4090 GPU.

B. Feature Alignment and Classification Results

Since several subjects are in one dataset, each subject is selected as a target subject sequentially to build training and

TABLE II
THE DETAILED RESULTS FOR DIFFERENT TARGET SUBJECTS ON ENABL3S AND DSADS DATASETS. STD INDICATES THE STANDARD DEVIATION. S1 REPRESENTS THE SUBJECT 1, WHICH MEANS THE FIRST SUBJECT IS THE TARGET SUBJECT AND OTHER SUBJECTS ARE SOURCE SUBJECTS.

Dataset	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	**Means**	**Std**
ENABL3S	97.8	95.3	94.9	91.5	94.6	94.2	91.3	95.4	96.7	94.5	**94.6**	2.0
DSADS	99.1	99.7	98.8	99.7	99.4	99.4	98.2	97.7	–	–	**99.0**	0.7

TABLE III
THE MEAN ACCURACY OF CLASSIFICATION FOR THE TARGET SUBJECT OF ENABL3S AND DSADS USING DIFFERENT DOMAIN ADAPTATION METHODS. STD INDICATES THE STANDARD DEVIATION.

Method	**ENABL3S**		**DSADS**	
	Mean(%)	**Std(%)**	**Mean(%)**	**Std(%)**
DANN [20]	88.5	4.8	91.1	5.2
MMD [21]	92.7	2.2	95.4	3.3
MCD [22]	93.9	1.8	95.3	4.5
DFA [55]	91.8	3.0	92.5	3.3
GFA [16]	93.7	1.7	96.9	3.3
EDHKD [17]	94.4	1.7	97.4	4.9
WMDD (Ours)	**94.6**	2.0	**99.0**	0.7

TABLE IV
THE ABLATION EXPERIMENTS FOR DOMAIN WEIGHT AND ADVERSARIAL LEARNING. NO-W AND NO-A REPRESENT THAT THE DOMAIN WEIGHT METHOD AND ADVERSARIAL LEARNING ARE NOT EMPLOYED, RESPECTIVELY. NO-W-A REPRESENTS BOTH METHODS ARE NOT USED.

Name	Weight $\boldsymbol{\alpha}$	Adversary	ENABL3S	DSADS
No-W-A			93.1 ± 2.2	92.7 ± 4.2
No-W		✓	93.6 ± 2.3	98.0 ± 2.1
No-A	✓		93.0 ± 2.6	92.3 ± 4.4
Ours	✓	✓	$\mathbf{94.6 \pm 2.0}$	$\mathbf{99.0 \pm 0.7}$

testing datasets. Therefore, the average classification accuracy of each target subject is used as the final evaluation indicator.

The Results of Feature Alignment: The dimensional reduction method t-SNE [54], which can keep the clustering of high dimensional space, is used to visualize the results of feature extraction following previous work [17]. Fig. 3 shows the t-SNE projection of the non-adapted input features and the hidden features adapted by WMDD. It can be found that the input features are not clustered, and the source distribution is not aligned with the target distribution. After feature alignment, the features of the same class almost cluster together for two datasets. For DSADS dataset, source and target features that belong to the same class are almost in the same region. However, for ENABL3S dataset, some source and target features that belong to the same class aren't in the same region, which causes the classification accuracy of ENABL3S dataset is worse than that of DSADS dataset. The reason is that about 30% of data for ENABL3S are transition activities instead of steady activities, where the activity is defined as a transition activity if the locomotion mode of the current activity is different from that of the last activity.

The Classification Results on ENABL3S and DSADS: To answer $\mathbf{Q_1}$, we compare different domain adaptation methods in ENABL3S and DSADS datasets, including DANN [20], MMD [21], MCD [22], DFA [55], GFA [16] and EDHKD [17]. The classification results on ENABL3S and DSADS of these methods come from article [17], where EDHKD is the state-of-the-art method for HMI using domain adaptation technology. Table II The detailed results for different target subjects on ENABL3S and DSADS datasets. Table III compares the mean accuracy of classification on the target subject for different domain adaptation methods. The test time of methods on this table is close to 1 ms, which guarantees the real-time of the test.

Table III shows that WMDD achieves the accuracy of $94.6\% \pm 2.0\%$ and $99.0\% \pm 0.7\%$ for target subject in ENABL3S and DSADS datasets respectively, which are 0.2% and 1.6% higher than the state-of-the-art HMI result using domain adaptation methods. This verifies that WMDD can achieve a better performance than previous methods in both ENABL3S and DSADS datasets. However, the improvement in ENABL3S is not significant despite that the samples of ENABL3S are larger than DSADS. The reason lies in the effects of transition activities. This results of classification are consistent with that of feature alignment.

C. Ablation Experiments

Effects of Weight $\boldsymbol{\alpha}$ and Adversarial Learning: To answer $\mathbf{Q_2}$, the ablation experiments for two parts are designed (see Table IV). The weight $\boldsymbol{\alpha}$ reflects the discrepancy between different source domains and target domain. The adversarial learning is used to improve the classifier's performance. From this table, the performance in ENABL3S and DSADS is improved by 1.07% and 1.02% respectively when considering the difference between source domains. The performance is improved by 1.72% and 7.25% respectively when introducing the adversarial learning between the feature generator and classifier. Therefore, the usage of domain weight and adversarial learning can both improve the performance of the classifier, and the effect of adversarial learning is more significant.

Effect of Margin Coefficient γ: The margin coefficient γ determines the margin disparity discrepancy between source and target distributions and influences the classification accuracy on the target subject, that is generalization error. Theorem 4 gives the equivalent form of MDD. It reflects the minimum of MDD is $\gamma \log \gamma - (1+\gamma)\log(1+\gamma)$ when the source distribution equals the target distribution. The margin

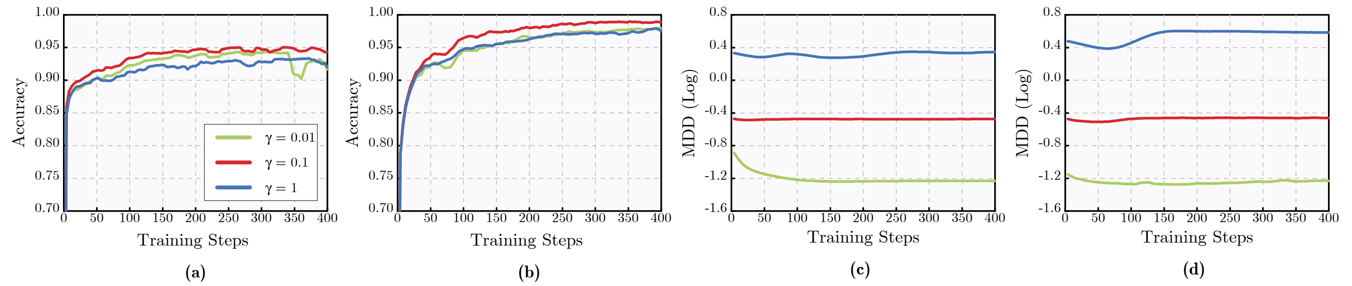

Fig. 4. The comparison results of training curve under different margin μ on ENABL3S (a) and DSADS (b) datasets. The comparison results of the average margin disparity discrepancy between source and target distribution under different μ on ENABL3S (c) and DSADS (d) datasets. For the convenience of comparison, the ordinate values of sub-figure (c) and (d) are $\log \mathrm{MDD}$.

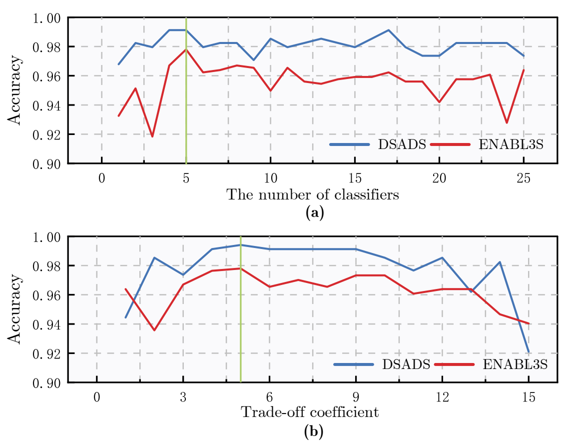

Fig. 5. (a) The relationship between classification accuracy and the number of classifiers K on ENABL3S and DSADS datasets. The red line and blue line represent the classification accuracy of ENABL3S and DSADS datasets respectively. (b) The relationship between classification accuracy and trade-off coefficient η on ENABL3S and DSADS datasets. The red dotted line represents the relatively optimal parameters on two datasets.

coefficient γ only influences the value of MDD and doesn't affect the gap between source and target distributions.

To answer $\mathbf{Q_3}$, we give the training curve and MDD under margin coefficient $\gamma = 0.01, 0.1, 1$ on ENABL3S and DSADS datasets. Fig. 4 shows that the method WMDD achieves better classification accuracy under margin coefficient $\gamma = 0.1$ for ENABL3S and DSADS datasets. The recognition accuracy are 91.6%, 94.6%, 92.0% under $\gamma = 0.01$, 0.1, 1 respectively for ENABL3S dataset, and are 97.9%, 99.0%, 97.5% for DSADS dataset. The final average MDD values are 0.06, 0.34, 2.20 under $\gamma = 0.01$, 0.1, 1 respectively for ENABL3S dataset, and are 0.06, 0.35, 3.84 for DSADS dataset. The results indicate that the average MDD between each source subject and target subject increases as margin coefficient γ increases. However, for too large γ, the recognition accuracy cannot be improved. This shows that the margin μ in Theorem 3 cannot be set too large.

Effects of the Number of Classifiers K ***and Trade-off Coefficient*** η: To answer $\mathbf{Q_3}$, the more experiments for the number of classifiers K and trade-off coefficient η are conducted below. The more classifiers often consume more computation cost. It is a necessity to choose an appropriate number of classifiers to achieve a balance between the classification performance and computation cost. The coefficient η is used to trade off the fitting ability and generalization ability of the network.

Fig. 5 (a) gives the relationship between classification accuracy and the number of classifiers K on ENABL3S and DSADS datasets. This shows that it's not that the more classifiers there are, the better the classification performance. Compared with the accuracy of 5 classifiers, the accuracy of 25 classifiers decreases by 1.56% and 5.61% for ENABL3S and DSADS datasets respectively. For lightweight networks, only by selecting an appropriate number of classifiers can classification performance be effectively improved. Fig. 5 (b) shows the relationship between classification accuracy and trade-off coefficient η on ENABL3S and DSADS datasets. The results indicate that too large or too small trade-off coefficient η will decrease the performance of the algorithm. The relatively optimal parameters for K and η are both 5 on ENABL3S and DSADS datasets.

VI. Discussion

This paper aims to extend MDD theory to multi-source UDA, and propose a novel multi-source UDA algorithm for HMI recognition based on the derived theory. In the below part, we give further discussions for multi-source UDA theory and algorithm.

A. The Discussion for Theory

The developed theory in this paper is based on [23], [40]. It can effectively answer the challenges listed in section I. *Challenge 1:* The margin disparity discrepancy is used to measure the gap between source and target domains. The MDD has below features. 1) The MDD contains more generalization bound information. Theorem 3 shows the generalization bound is influenced by the margin μ. The better performance on the the target subject can be achieved through choosing appropriate margin μ. 2) The MDD can accurately measure the gap between different distributions. Theorem 4 shows the estimated MDD reflects the gap through KL divergence, which avoids the error brought by estimating KL divergence.

Challenge 2: The dataset is collected from multiple source domains that might be different not only from the target domain but also from each other. The source domain weight determined by MDD is incorporated into Theorem 1 to measure the difference between source domains, where the source domain with the small gap to the target domain should have a large weight. The weight is adjusted adaptively by MDD according to Eq. (14). Since the training data is randomly selected, the drawn data hardly fully describe the entire distribution characteristic. Adjusting weight dynamically according to the drawn data is beneficial to improving performance on the target domain. Table IV also verifies this conclusion.

B. The Discussion for Algorithm

Since the margin loss easily leads to gradient vanishing, the cross-entropy loss is employed to replace the margin loss in this paper. The γ in Eq. (18) reflects the margin μ in MDD. Theorem 4 shows that the result of using a surrogate function can better measure the discrepancy between different distributions and maintain the main characteristics of MDD through γ. The algorithm is essentially a two-stage game problem. The first stage is stackelberg game, and the second stage is zero-sum game. The details are given below.

1) Stackelberg game: Eq. (12) can be regarded as stackelberg game model, where auxiliary classifiers $\boldsymbol{f}_j'$ is set as leader and classifiers $\boldsymbol{f}$ is follower. In this stage, the auxiliary classifiers are first updated and the MDD between source and target domain is estimated. Then classifiers and feature generator are updated together through minimizing the generalization error.
2) Zero-sum game: Eq. (22) can be regarded as two-player zero-sum game model, where feature generator ϕ and classifiers $\{\boldsymbol{f}^{(i)}\}_{i=1}^K$ are the adversarial parties. The generalization ability (performance on target domain) is improved through constant adversarial learning between feature generator and classifiers.

VII. Conclusion

This paper developed a new theory for multi-source UDA based on margin disparity discrepancy and derived a novel generalization bound for multi-source UDA. Motivated by generalization bound, a novel weight-aware-based multi-source UDA algorithm (WMDD) was proposed for HMI recognition. The proposed method WMDD can improve classification accuracy of HMI recognition tasks through considering the difference between each source subject, where the source domain weight can be adjusted adaptively by the estimated discrepancy. WMDD can also guarantee the real-time of HMI recognition by utilizing a lightweight network. Extensive experiments confirmed that WMDD can achieve state-of-the-art accuracy on HMI recognition tasks. We expect the proposed theory and algorithm can provide reference and inspiration for other multi-source UDA theories and cross-subject application fields.

Appendix

A. Proof of Theorem 1

Before proving the Theorem 1, we firstly give the below Proposition similar to the previous work [23].

Proposition 1: Fix μ. For any scoring function $\boldsymbol{f} \in \mathcal{F}$,

$$\mathbb{W}_\mu[\omega_{\boldsymbol{f}'}(x, h_{\boldsymbol{f}}(x))] \le \mathbb{W}_\mu[\omega_{\boldsymbol{f}}(x,y)] + \mathbb{W}_\mu[\omega_{\boldsymbol{f}'}(x,y)],$$

where $\omega_{\boldsymbol{f}}(\boldsymbol{x}, y) = [f_y(\boldsymbol{x}) - \max_{y' \neq y} f_{y'}(\boldsymbol{x})]/2$ is the margin of $\boldsymbol{f}$ in sample (x,y), and $\mathbb{W}_\mu(v)$ is the ramp loss that is defined in Eq. (4).

Proof. For any sample (x,y), if $h_{\boldsymbol{f}}(x) \neq y$ or $h_{\boldsymbol{f}'}(x) \neq y$, $\omega_{\boldsymbol{f}}(x,y)$ or $\omega_{\boldsymbol{f}}(x,y)$ is small than zero, implying the right side of above equation reach 1, and further deducing that the inequality always holds. Otherwise $h_{\boldsymbol{f}}(x) = y$ and $h_{\boldsymbol{f}'}(x) = y$, then we can derive

$$\begin{aligned} &\mathbb{W}_\mu[\omega_{\boldsymbol{f}'}(x, h_{\boldsymbol{f}}(x))] \\ \le &\mathbb{W}_\mu[\omega_{\boldsymbol{f}'}(x, h_{\boldsymbol{f}}(x))] + \mathbb{W}_\mu[\omega_{\boldsymbol{f}}(x,y)] \\ \le &\mathbb{W}_\mu[\omega_{\boldsymbol{f}'}(x,y)] + \mathbb{W}_\mu[\omega_{\boldsymbol{f}}(x,y)]. \end{aligned} \tag{25}$$

This completes the proof of Proposition 1. Next we give the proof for Theorem 1. The proof follows the previous works [23], [24].

The Proof for Theorem 1: Let be the ideal scoring function $\boldsymbol{f}^*$ which minimizes the combined margin loss,

$$\boldsymbol{f}^* = \arg\min_{\boldsymbol{f} \in \mathcal{F}} \left\{ \mathcal{L}_Q^\mu(\boldsymbol{f}) + \mathcal{L}_P^\mu(\boldsymbol{f}) \right\}. \tag{26}$$

Then, for any $\boldsymbol{f} \in \mathcal{F}$,

$$\begin{aligned} &\mathcal{L}_Q(h_{\boldsymbol{f}}) = \mathbb{E}_Q\{\mathbb{I}[h_{\boldsymbol{f}}(\boldsymbol{x}) \neq y]\} \\ \le &\mathbb{E}_Q\{\mathbb{I}[h_{\boldsymbol{f}}(\boldsymbol{x}) \neq h_{\boldsymbol{f}^*}(\boldsymbol{x})]\} + \mathbb{E}_Q\{\mathbb{I}[h_{\boldsymbol{f}^*}(\boldsymbol{x}) \neq y]\} \\ \le &\mathbb{E}_Q\{\mathbb{W}_\mu[\omega_{\boldsymbol{f}^*}(x, h_{\boldsymbol{f}}(x))]\} + \underbrace{\mathbb{E}_Q\{\mathbb{W}_\mu[\omega_{\boldsymbol{f}^*}(x,y)]\}}_{:=\mathcal{L}_Q^\mu(\boldsymbol{f}^*)} \\ &+ \underbrace{\mathbb{E}_P\{\mathbb{W}_\mu[\omega_{\boldsymbol{f}}(x,y)]\}}_{:=\mathcal{L}_P^\mu(\boldsymbol{f})} - \underbrace{\mathbb{E}_P\{\mathbb{W}_\mu[\omega_{\boldsymbol{f}}(x,y)]\}}_{\text{Proposition 1}} \\ \le &\mathcal{L}_P^\mu(\boldsymbol{f}) + \mathcal{L}_Q^\mu(\boldsymbol{f}^*) + \underbrace{\mathbb{E}_P\{\mathbb{W}_\mu[\omega_{\boldsymbol{f}^*}(x,y)]\}}_{:=\mathcal{L}_P^\mu(\boldsymbol{f}^*)} \\ &+ \underbrace{\mathbb{E}_Q\{\mathbb{W}_\mu[\omega_{\boldsymbol{f}^*}(x, h_{\boldsymbol{f}}(x))]\} - \mathbb{E}_P\{\mathbb{W}_\mu[\omega_{\boldsymbol{f}^*}(x, h_{\boldsymbol{f}}(x))]\}}_{\text{Eq. (5)}} \\ \le &\mathcal{L}_P^\mu(\boldsymbol{f}) + d_{\boldsymbol{f}}^\mu(P,Q) + \lambda, \end{aligned}$$

where the second inequality is the important property of margin loss for any μ and $\boldsymbol{f}$, and $\lambda = \mathcal{L}_Q^\mu(\boldsymbol{f}^*) + \mathcal{L}_P^\mu(\boldsymbol{f}^*) = \min_{\boldsymbol{f} \in \mathcal{F}}\{\mathcal{L}_Q^\mu(\boldsymbol{f}) + \mathcal{L}_P^\mu(\boldsymbol{f})\}$. Therefore, the proof of Theorem 1 is completed.

B. Proof of Theorem 2

The Proof for Theorem 2: Let $\widetilde{P}$ be the mixture distribution of the N source domains, denoted as $\widetilde{P} = \sum_{j=1}^N \alpha_j P_j$, and $\mathcal{D}_{\tilde{s}}$ be the combined samples from N source domains. We denote $\widetilde{P}$ and Q as the source distribution and the target distribution in Theorem 1, respectively. Then, we have

$$\mathcal{L}_Q(h_{\boldsymbol{f}}) \le \mathcal{L}_{\widetilde{P}}^\mu(\boldsymbol{f}) + d_{\boldsymbol{f}}^\mu\left(\widetilde{P}, Q\right) + \widetilde{\lambda}. \tag{27}$$

On the one hand, for any $\boldsymbol{f} \in \mathcal{F}$, the following holds

$$\mathcal{L}_{\widetilde{P}}^\mu(\boldsymbol{f}) = \sum_{j=1}^N \alpha_j \mathcal{L}_{P_j}^\mu(\boldsymbol{f}), \tag{28}$$

then $\widetilde{\lambda}=\min_{\boldsymbol{f}\in\mathcal{F}}\{\sum_{j=1}^{N}\alpha_j\mathcal{L}_{P_j}^{\mu}(\boldsymbol{f})+\mathcal{L}_{Q}^{\mu}(\boldsymbol{f})\}$, we denote it as β here. On the other hand, according to Eq. (5) the term $d_{\boldsymbol{f}}^{\mu}(\widetilde{P},Q)$ can be upper bounded by

$$\begin{aligned}
d_{\boldsymbol{f}}^{\mu}\left(\widetilde{P},Q\right) &= \sup_{\boldsymbol{f}'\in\mathcal{F}}\left\{\mathbb{E}_Q\left[\mathbb{W}_\mu\right]-\mathbb{E}_{\widetilde{P}}\left[\mathbb{W}_\mu\right]\right\}\\
&= \sup_{\boldsymbol{f}'\in\mathcal{F}}\left\{\mathbb{E}_Q\left[\mathbb{W}_\mu\right]-\sum_{j=1}^{N}\alpha_j\mathbb{E}_{P_j}\left[\mathbb{W}_\mu\right]\right\}\\
&= \sup_{\boldsymbol{f}'\in\mathcal{F}}\left\{\sum_{j=1}^{N}\alpha_j\Big(\mathbb{E}_Q\left[\mathbb{W}_\mu\right]-\mathbb{E}_{P_j}\left[\mathbb{W}_\mu\right]\Big)\right\}\\
&\leq \sum_{j=1}^{N}\alpha_j\sup_{\boldsymbol{f}'\in\mathcal{F}}\left\{\mathbb{E}_Q\left[\mathbb{W}_\mu\right]-\mathbb{E}_{P_j}\left[\mathbb{W}_\mu\right]\right\}\\
&= \sum_{j=1}^{N}\alpha_j d_{\boldsymbol{f}}^{\mu}\left(P_j,Q\right),
\end{aligned} \tag{29}$$

where the first inequality is by the sub-additivity of the sup function. Then bringing the Eqs. (28) and (29) into Eq. (27), we can get

$$\mathcal{L}_Q\left(h_{\boldsymbol{f}}\right)\leq\sum_{j=1}^{N}\alpha_j\left(\mathcal{L}_{P_j}^{\mu}\left(\boldsymbol{f}\right)+d_{\boldsymbol{f}}^{\mu}\left(P_j,Q\right)\right)+\beta. \tag{30}$$

This completes the proof of Theorem 2.

C. Proof of Theorem 3

Lemma 1: Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{X}$ to $[0,1]$ and $\widehat{\mathcal{D}}$ be empirical datasets sampled from an i.i.d. sample $\mathcal{D}$ of size m. Then, for any $\delta>0$, with probability at least $1-\delta$, the following holds for all $g\in\mathcal{G}$,

$$\left|\mathbb{E}_{\mathcal{D}}\left[g\right]-\mathbb{E}_{\widehat{\mathcal{D}}}\left[g\right]\right|\leq 2\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\mathcal{G})+3\sqrt{\frac{\log(2/\delta)}{2m}}, \tag{31}$$

where $\mathbb{E}_{\widehat{\mathcal{D}}}\left[g\right]=\frac{1}{m}\sum_{i=1}^{m}\left[g(x_i)\right]$ is the empirical form of $\mathbb{E}_{\mathcal{D}}\left[g\right]$. This proof can be found in Theorem 3.3 of work [56].

Lemma 2: Let $\mathcal{G}$ be a family of functions, denoted as $\mathcal{G}=\{\max\{\boldsymbol{f}_1,\boldsymbol{f}_2,...,\boldsymbol{f}_k\}\mid\boldsymbol{f}_i\in\mathcal{F},i\in\{1,2,...,k\}\}$. Then, for any sample $\mathcal{D}$ of size m, the following holds for all $g\in\mathcal{G}$,

$$\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\mathcal{G})\leq k\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\mathcal{F}). \tag{32}$$

This Lemma is the modified version of Lemma C.6 in previous work [23].

Proposition 2: Let $\mathcal{G}$ be a family of margin loss functions defined in Eq. (3) and $\Omega_1(\mathcal{F})$ be a family of functions defined in Definition 2. Then, for any sample $\widehat{\mathcal{D}}=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^{m}$, where $y_i\in\{1,...,\mathcal{C}\}$, the following relation holds between the empirical Rademacher complexities of $\mathcal{G}$ and $\Omega_1(\mathcal{F})$:

$$\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\mathcal{G})\leq\frac{\mathcal{C}}{2\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\Omega_1(\mathcal{F})). \tag{33}$$

Proof. According to Definition 1, the empirical Rademacher complexity of $\mathcal{G}$ can be written as:

$$\begin{aligned}
\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\mathcal{G}) =&\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\mathbb{W}_\mu\left[\omega_{\boldsymbol{f}}(\boldsymbol{x}_i,y_i)\right]\right]\\
=&\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\frac{\boldsymbol{f}_y(\boldsymbol{x}_i)-\max_{y'\neq y_i}\boldsymbol{f}_{y'}(\boldsymbol{x}_i)}{2\mu}\right]\\
\leq&\underbrace{\frac{1}{2\mu}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\boldsymbol{f}_{y_i}(\boldsymbol{x}_i)\right]}_{:=\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\Omega_1(\mathcal{F}))}+\\
&\underbrace{\frac{1}{2\mu}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\max_{y'\neq y_i}\boldsymbol{f}_{y'}(\boldsymbol{x}_i)\right]}_{:=\Delta}.
\end{aligned} \tag{34}$$

Because $\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\mathbb{W}_\mu(v)\right]=0$ holds for $\mu\leq v$ or $v\leq 0$, so here we only consider the case of $0\leq v\leq\mu$. Let $\Omega_1(\mathcal{F}^{\mathcal{C}-1})=\{\max\{\boldsymbol{f}_1,...,\boldsymbol{f}_{\mathcal{C}-1}\}\mid\boldsymbol{f}_j\in\Omega_1(\mathcal{F}),j\in\{1,...,\mathcal{C}-1\}\}$. Then according to Lemma 2, the term Δ can be rewritten as:

$$\begin{aligned}
\Delta &= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\max_{y'\neq y_i}\boldsymbol{f}_{y'}(\boldsymbol{x}_i)\right]\\
&= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\max_{j\in\{1,...,\mathcal{C}-1\}}\boldsymbol{f}_j(\boldsymbol{x}_i)\right]\\
&= \widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\Omega_1(\mathcal{F}^{\mathcal{C}-1}))\\
&\leq (\mathcal{C}-1)\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\Omega_1(\mathcal{F})).
\end{aligned} \tag{35}$$

Therefore, $\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\mathcal{G})\leq\frac{\mathcal{C}}{2\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\Omega_1(\mathcal{F}))$. This completes the proof of Proposition 2.

Proposition 3: Let $\mathcal{G}$ be a family of margin loss functions defined as $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_x}\left\{\mathbb{W}_\mu\left[\omega_{\boldsymbol{f}'}(x,h_{\boldsymbol{f}}(x))\right]\right\}$ and $\Omega_2(\mathcal{F})$ be a family of functions defined in Definition 2. Then, for any sample $\widehat{\mathcal{D}}_x=\{(\boldsymbol{x}_i\}_{i=1}^{m}$, the following relation holds between the empirical Rademacher complexities of $\mathcal{G}$ and $\Omega_2(\mathcal{F})$:

$$\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_x}(\mathcal{G})\leq\frac{\mathcal{C}}{2\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_x}(\Omega_2(\mathcal{F})). \tag{36}$$

Proof. The proof is similar to Proposition 2, the empirical Rademacher complexity of $\mathcal{G}$ can be written as:

$$\begin{aligned}
&\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_x}(\mathcal{G})\\
=&\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f},\boldsymbol{f}'\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\mathbb{W}_\mu\left[\omega_{\boldsymbol{f}'}(\boldsymbol{x}_i,\boldsymbol{f}(x))\right]\right]\\
=&\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\boldsymbol{f},\boldsymbol{f}'\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\frac{\boldsymbol{f}'_{h_{\boldsymbol{f}}(\boldsymbol{x})}(\boldsymbol{x}_i)-\max_{y'\neq h_{\boldsymbol{f}}(\boldsymbol{x})}\boldsymbol{f}_{y'}(\boldsymbol{x}_i)}{2\mu}\right]\\
\leq&\frac{\mathcal{C}}{2\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_x}(\Omega_2(\mathcal{F})).
\end{aligned} \tag{37}$$

This completes the proof of Proposition 3.

Proposition 4: Let $\widehat{P}$ and $\widehat{Q}$ be the corresponding empirical distributions for sample $\mathcal{D}_s=(\boldsymbol{X},\boldsymbol{Y}_j)$ and $\mathcal{D}_t=\boldsymbol{X}_t$. For any

$\delta > 0$, with probability at least $1-2\delta$, the following holds for any scoring function $\boldsymbol{f} \in \mathcal{F}$,

$$\left|d_{\boldsymbol{f}}^{\mu}(P,Q) - d_{\boldsymbol{f}}^{\mu}\left(\widehat{P},\widehat{Q}\right)\right| \leq \frac{\mathcal{C}}{\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_s}(\Omega_2(\mathcal{F})) + 3\sqrt{\frac{\log(2/\delta)}{2|\widehat{\mathcal{D}}_s|}} + \frac{\mathcal{C}}{\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_t}(\Omega_2(\mathcal{F})) + 3\sqrt{\frac{\log(2/\delta)}{2|\widehat{\mathcal{D}}_t|}}.$$

Proof. Eq. (5) gives the margin disparity discrepancy $d_{\boldsymbol{f}}^{\mu}(P,Q) = \sup_{\boldsymbol{f}'\in\mathcal{F}}\left\{\mathbb{E}_Q\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right] - \mathbb{E}_P\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right]\right\}$, then we have,

$$\begin{aligned}&\left|d_{\boldsymbol{f}}^{\mu}(P,Q) - d_{\boldsymbol{f}}^{\mu}\left(\widehat{P},\widehat{Q}\right)\right| \\ \leq& \sup_{\boldsymbol{f}'\in\mathcal{F}}\left|\mathbb{E}_Q\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right] - \mathbb{E}_{\widehat{Q}}\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right]\right| \\ &+ \sup_{\boldsymbol{f}'\in\mathcal{F}}\left|\mathbb{E}_P\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right] - \mathbb{E}_{\widehat{P}}\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right]\right|.\end{aligned} \tag{38}$$

By applying Lemma 1 and Proposition 3, then

$$\begin{aligned}&\left|\mathbb{E}_Q\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right] - \mathbb{E}_{\widehat{Q}}\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right]\right| \\ \leq& \frac{\mathcal{C}}{\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_t}(\Omega_2(\mathcal{F})) + 3\sqrt{\frac{\log(2/\delta)}{2|\widehat{\mathcal{D}}_t|}}, \\ &\left|\mathbb{E}_P\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right] - \mathbb{E}_{\widehat{P}}\left[\mathbb{W}_\mu(\omega_{\boldsymbol{f}'})\right]\right| \\ \leq& \frac{\mathcal{C}}{\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_s}(\Omega_2(\mathcal{F})) + 3\sqrt{\frac{\log(2/\delta)}{2|\widehat{\mathcal{D}}_s|}}.\end{aligned} \tag{39}$$

Combing Eqs. (38) and (39), we can get the final result, which completes the proof of Proposition 4. Next, we give the proof for Theorem 3.

The Proof for Theorem 3: First, according to Lemma 1 and Proposition 2, for $\mathcal{L}_P^{\mu}(\boldsymbol{f}) = \mathbb{E}_{(\boldsymbol{x},y)\sim P}\left\{\mathbb{W}_\mu\left[\omega_{\boldsymbol{f}}(\boldsymbol{x},y)\right]\right\}$, we have

$$\left|\mathcal{L}_P^{\mu}(\boldsymbol{f}) - \mathcal{L}_{\widehat{P}}^{\mu}(\boldsymbol{f})\right| \leq \frac{\mathcal{C}}{\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\Omega_1(\mathcal{F})) + 3\sqrt{\frac{\log(2/\delta)}{2|\widehat{\mathcal{D}}|}}. \tag{40}$$

Then, combining Theorem 2 and Proposition 4, we have

$$\begin{aligned}\mathcal{L}_Q(h_{\boldsymbol{f}}) \leq& \sum_{j=1}^{N}\alpha_j\left\{\mathcal{L}_{\widehat{P}_j}^{\mu}(\boldsymbol{f}) + d_{\boldsymbol{f}}^{\mu}\left(\widehat{P}_j,\widehat{Q}\right)\right. \\ &\left.+\frac{\mathcal{C}}{\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_{sj}}(\Omega_2(\mathcal{F})) + \frac{\mathcal{C}}{\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_{sj}}(\Omega_1(\mathcal{F})) + 6\sqrt{\frac{\log(2/\delta)}{2|\widehat{\mathcal{D}}_{sj}|}}\right\} \\ &+\frac{\mathcal{C}}{\mu}\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_t}(\Omega_2(\mathcal{F})) + 3\sqrt{\frac{\log(2/\delta)}{2|\widehat{\mathcal{D}}_t|}} + \beta.\end{aligned} \tag{41}$$

Since $\widehat{\mathcal{D}}_{sj}$ and $\widehat{\mathcal{D}}_t$ are empirical datasets sampled from the i.i.d. sample $\mathcal{D}_{sj} = (\boldsymbol{X}_j, \boldsymbol{Y}_j)$ and $\mathcal{D}_t = \boldsymbol{X}_t$ of size m, the above equation can be written as

$$\begin{aligned}\mathcal{L}_Q(h_{\boldsymbol{f}}) \leq& \sum_{j=1}^{N}\alpha_j\left\{\mathcal{L}_{\widehat{P}_j}^{\mu}(\boldsymbol{f}) + d_{\boldsymbol{f}}^{\mu}\left(\widehat{P}_j,\widehat{Q}\right)\right\} + 9\sqrt{\frac{\log(2/\delta)}{2m}} \\ &+ \frac{\mathcal{C}}{\mu}\left\{\widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}_s}(\Omega_1(\mathcal{F})) + \widehat{\mathfrak{R}}_{\widehat{\mathcal{D}}}(\Omega_2(\mathcal{F}))\right\} + \beta,\end{aligned}$$

where $\widehat{\mathcal{D}}_s$ and $\widehat{\mathcal{D}}$ are datasets sampled from the mixture distribution $\sum_{j=1}^{N}\alpha_j P_j$ and $Q + \sum_{j=1}^{N}\alpha_j P_j$, respectively. Therefore, the proof of Theorem 3 is completed.

D. Proof of Theorem 4

The Proof for Theorem 4: This proof follows the previous works [23], [49]. Since the function $f(y) = a\log(y) + b\log(1-y)$ achieves its maximum in $[0,1]$ at $a/(a+b)$, for the below optimization problem: $\max \gamma\mathbb{E}_P[\log D(\boldsymbol{x})] + \mathbb{E}_Q[\log(1-D(\boldsymbol{x}))]$, the optimal $D(\boldsymbol{x})$ is

$$D^*(\boldsymbol{x}) = \frac{\gamma P(\boldsymbol{x})}{\gamma P(\boldsymbol{x}) + Q(\boldsymbol{x})}. \tag{42}$$

Then, under the optimal $D^*(\boldsymbol{x})$, the maximization of $\gamma\mathbb{E}_P[\log D(\boldsymbol{x})] + \mathbb{E}_Q[\log(1-D(\boldsymbol{x}))]$ can be denoted as

$$\begin{aligned}&\gamma\mathbb{E}_P\left[\log D(\boldsymbol{x})\right] + \mathbb{E}_Q\left[\log(1-D(\boldsymbol{x}))\right] \\ =&\int_{\boldsymbol{x}\in\mathcal{X}}\gamma P(\boldsymbol{x})\log\left[\frac{\gamma P(\boldsymbol{x})}{\gamma P(\boldsymbol{x}) + Q(\boldsymbol{x})}\right]\mathrm{d}x \\ &+\int_{\boldsymbol{x}\in\mathcal{X}}Q(\boldsymbol{x})\log\left[\frac{Q(\boldsymbol{x})}{\gamma P(\boldsymbol{x}) + Q(\boldsymbol{x})}\right]\mathrm{d}x \\ =&\int_{\boldsymbol{x}\in\mathcal{X}}\gamma P(\boldsymbol{x})\log\left[\frac{\gamma P(\boldsymbol{x})}{\frac{\gamma P(\boldsymbol{x})+Q(\boldsymbol{x})}{\gamma+1}}\right]\mathrm{d}x \\ &+\int_{\boldsymbol{x}\in\mathcal{X}}Q(\boldsymbol{x})\log\left[\frac{Q(\boldsymbol{x})}{\frac{\gamma P(\boldsymbol{x})+Q(\boldsymbol{x})}{\gamma+1}}\right]\mathrm{d}x \\ &+\int_{\boldsymbol{x}\in\mathcal{X}}\gamma P(\boldsymbol{x})\log\gamma - \left[\gamma P(\boldsymbol{x}) + Q(\boldsymbol{x})\right]\log(\gamma+1)\mathrm{d}x \\ =&\gamma KL\left(P\ \middle|\middle|\ \frac{\gamma P + Q}{\gamma+1}\right) + KL\left(Q\ \middle|\middle|\ \frac{\gamma P + Q}{\gamma+1}\right) \\ &+\gamma\log\gamma - (\gamma+1)\log(\gamma+1).\end{aligned} \tag{43}$$

Therefor, when the above $D(\boldsymbol{x})$ represents the function $\Theta_{h_{\boldsymbol{f}'_j}}(\boldsymbol{f}(\phi(\boldsymbol{x})))$, fixing the classifier $\boldsymbol{f}$, the theorem 4 can be proven.

References

[1] P. Huang, Z. Li, M. Zhou, and Z. Kan, "Divergent component of motion planning and adaptive repetitive control for wearable walking exoskeletons," *IEEE Trans. Cybern.*, vol. 54, no. 4, pp. 2244–2256, 2024.

[2] G. Li, L. Cheng, Z. Gao, X. Xia, and J. Jiang, "Development of an untethered adaptive thumb exoskeleton for delicate rehabilitation assistance," *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3514–3529, 2022.

[3] G. Li, X. Liang, H. Lu, T. Su, and Z.-G. Hou, "Development and validation of a self-aligning knee exoskeleton with hip rotation capability," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 32, pp. 472–481, 2024.

[4] G. Li, Z. Li, C.-Y. Su, and T. Xu, "Active human-following control of an exoskeleton robot with body weight support," *IEEE Trans. Cybern.*, vol. 53, no. 11, pp. 7367–7379, 2023.

[5] Z. Li, T. Zhang, P. Huang, and G. Li, "Human-in-the-loop cooperative control of a walking exoskeleton for following time-variable human intention," *IEEE Trans. Cybern.*, vol. 54, no. 4, pp. 2142–2154, 2024.

[6] D. D. Molinaro, I. Kang, and A. J. Young, "Estimating human joint moments unifies exoskeleton control, reducing user effort," *Sci. Robot.*, vol. 9, no. 88, p. eadi8852, 2024.

[7] X. Hao, Z. Li, P. Huang, P. Shi, and G. Li, "Hierarchical task-oriented whole-body locomotion of a walking exoskeleton using adaptive dynamic motion primitive for cart pushing," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–12, 2023.

[8] X. Yu, W. He, Y. Li, C. Xue, J. Li, J. Zou, and C. Yang, "Bayesian estimation of human impedance and motion intention for human–robot collaboration," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 1822–1834, 2021.

[9] L.-L. Li, G.-Z. Cao, H.-J. Liang, Y.-P. Zhang, and F. Cui, "Human lower limb motion intention recognition for exoskeletons: A review," *IEEE Sens. J*, vol. 23, no. 24, pp. 30 007–30 036, 2023.

[10] S. Qiu, W. Guo, D. Caldwell, and F. Chen, "Exoskeleton online learning and estimation of human walking intention based on dynamical movement primitives," *IEEE Trans. Cogn. Develop. Syst.*, vol. 13, no. 1, pp. 67–79, 2021.

[11] K. Gui, U.-X. Tan, H. Liu, and D. Zhang, "Electromyography-driven progressive assist-as-needed control for lower limb exoskeleton," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 1, pp. 50–58, 2020.

[12] G. Durandau, D. Farina, and M. Sartori, "Robust real-time musculoskeletal modeling driven by electromyograms," *IEEE. Trans. Biomed. Eng.*, vol. 65, no. 3, pp. 556–564, 2018.

[13] J. Liu, X. Zhou, B. He, P. Li, C. Wang, and X. Wu, "A novel method for detecting misclassifications of the locomotion mode in lower-limb exoskeleton robot control," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7779–7785, 2022.

[14] Z. Yin, J. Zheng, L. Huang, Y. Gao, H. Peng, and L. Yin, "Sa-svm-based locomotion pattern recognit. for exoskeleton robot," *Appl. Sci.*, vol. 11, no. 12, p. 5573, 2021.

[15] C. Wang, Z. Guo, S. Duan, B. He, Y. Yuan, and X. Wu, "A real-time stability control method through semg interface for lower extremity rehabilitation exoskeletons," *Front. Neurosci.*, vol. 15, p. 645374, 2021.

[16] K. Zhang, J. Chen, J. Wang, Y. Leng, C. W. de Silva, and C. Fu, "Gaussian-guided feature alignment for unsupervised cross-subject adaptation," *Pattern Recognit.*, vol. 122, p. 108332, 2022.

[17] K. Zhang, J. Chen, J. Wang, X. Chen, Y. Leng, C. W. de Silva, and C. Fu, "Ensemble diverse hypotheses and knowledge distillation for unsupervised cross-subject adaptation," *Inf. Fusion*, vol. 93, pp. 268–281, 2023.

[18] K. Zhang, J. Wang, C. W. de Silva, and C. Fu, "Unsupervised cross-subject adaptation for predicting human locomotion intent," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 28, no. 3, pp. 646–657, 2020.

[19] Z. Mo, Z. Zhang, Q. Miao, and K.-L. Tsui, "Sparsity-constrained invariant risk minimization for domain generalization with application to machinery fault diagnosis modeling," *IEEE Trans. Cybern.*, vol. 54, no. 3, pp. 1547–1559, 2024.

[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.

[21] M. Long, Z. Cao, J. Wang, and P. S. Yu, "Learning multiple tasks with multilinear relationship networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[22] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3723–3732.

[23] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2019, pp. 7404–7413.

[24] Y. Zhang, B. Deng, H. Tang, L. Zhang, and K. Jia, "Unsupervised multi-class domain adaptation: Theory, algorithms, and practice," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2775–2792, 2022.

[25] N. Xiao and L. Zhang, "Dynamic weighted learning for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 242–15 251.

[26] K. Li, J. Lu, H. Zuo, and G. Zhang, "Multidomain adaptation with sample and source distillation," *IEEE Trans. Cybern.*, vol. 54, no. 4, pp. 2193–2205, 2024.

[27] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2015, pp. 1180–1189.

[28] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2015, pp. 97–105.

[29] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[30] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007.

[31] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[32] Y. Li, D. E. Carlson *et al.*, "Extracting relationships by multi-domain matching," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[33] Z. Chen, Y. Liao, J. Li, R. Huang, L. Xu, G. Jin, and W. Li, "A multi-source weighted deep transfer network for open-set fault diagnosis of rotary machinery," *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1982–1993, 2023.

[34] L. Ren and X. Cheng, "Single/multi-source black-box domain adaption for sensor time series data," *IEEE Trans. Cybern.*, 2023.

[35] S. Wang, B. Wang, Z. Zhang, A. A. Heidari, and H. Chen, "Class-aware sample reweighting optimal transport for multi-source domain adaptation," *Neurocomputing*, vol. 523, pp. 213–223, 2023.

[36] J. Bruinsma and R. Carloni, "Imu-based deep neural networks: Prediction of locomotor and transition intentions of an osseointegrated transfemoral amputee," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 29, pp. 1079–1088, 2021.

[37] J. Yang and Y. Yin, "Novel soft smart shoes for motion intent learning of lower limbs using lstm with a convolutional autoencoder," *IEEE Sens. J*, vol. 21, no. 2, pp. 1906–1917, 2021.

[38] N. Sun, M. Cao, Y. Chen, Y. Chen, J. Wang, Q. Wang, X. Chen, and T. Liu, "Continuous estimation of human knee joint angles by fusing kinematic and myoelectric signals," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 30, pp. 2446–2455, 2022.

[39] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, 2006.

[40] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, pp. 151–175, 2010.

[41] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint arXiv:0902.3430*, 2009.

[42] S. Kuroki, N. Charoenphakdee, H. Bao, J. Honda, I. Sato, and M. Sugiyama, "Unsupervised domain adaptation based on source-guided discrepancy," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, 2019, pp. 4122–4129.

[43] J. Wen, R. Greiner, and D. Schuurmans, "Domain aggregation networks for multi-source domain adaptation," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2020, pp. 10 214–10 224.

[44] Y. Yao, X. Li, Y. Zhang, and Y. Ye, "Multisource heterogeneous domain adaptation with conditional weighting adversarial network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2079–2092, 2023.

[45] Z.-G. Liu, L.-B. Ning, and Z.-W. Zhang, "A new progressive multisource domain adaptation network with weighted decision fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 1062–1072, 2024.

[46] S. Chen, L. Zheng, and H. Wu, "Riemannian representation learning for multi-source domain adaptation," *Pattern Recognit.*, vol. 137, p. 109271, 2023.

[47] S. Chen, "Multi-source domain adaptation with mixture of joint distributions," *Pattern Recognit.*, vol. 149, p. 110295, 2024.

[48] M. Mohri and A. Muñoz Medina, "New analysis and algorithm for learning with drifting distributions," in *Proc. Int. Conf. Algorithmic Learn. Theory*. Springer, 2012, pp. 124–138.

[49] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.

[50] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[51] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[52] B. Hu, E. Rouse, and L. Hargrove, "Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals," *Front. Robot. AI*, vol. 5, p. 14, 2018.

[53] B. Barshan and M. C. Yüksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *Comput. J.*, vol. 57, no. 11, pp. 1649–1667, 2014.

[54] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.

[55] J. Wang, J. Chen, J. Lin, L. Sigal, and C. W. de Silva, "Discriminative feature alignment: Improving transferability of unsupervised domain adaptation by gaussian-guided latent alignment," *Pattern Recognit.*, vol. 116, p. 107943, 2021.

[56] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.