Inter-Subject Domain Adaptation for CNN-Based Wrist Kinematics Estimation Using sEMG

Tianzhe Bao, Syed Ali Raza Zaidi, *Member, IEEE*, Shengquan Xie, *Senior Member, IEEE*, Pengfei Yang, *Member, IEEE*, and Zhi-Qiang Zhang, *Member, IEEE*

***Abstract*— Recently, convolutional neural network (CNN) has been widely investigated to decode human intentions using surface Electromyography (sEMG) signals. However, a pre-trained CNN model usually suffers from severe degradation when testing on a new individual, and this is mainly due to domain shift where characteristics of training and testing sEMG data differ substantially. To enhance inter-subject performances of CNN in the wrist kinematics estimation, we propose a novel regression scheme for supervised domain adaptation (SDA), based on which domain shift effects can be effectively reduced. Specifically, a two-stream CNN with shared weights is established to exploit source and target sEMG data simultaneously, such that domain-invariant features can be extracted. To tune CNN weights, both regression losses and a domain discrepancy loss are employed, where the former enable supervised learning and the latter minimizes distribution divergences between two domains. In this study, eight healthy subjects were recruited to perform wrist flexion-extension movements. Experiment results illustrated that the proposed regression SDA outperformed fine-tuning, a state-of-the-art transfer learning method, in both single-single and multiple-single scenarios of kinematics estimation. Unlike fine-tuning which suffers from catastrophic forgetting, regression SDA can maintain much better performances in original domains, which boosts the model reusability among multiple subjects.**

***Index Terms*— sEMG, wrist kinematics estimation, CNN, domain adaptation, transfer learning.**

I. Introduction

THE surface electromyography (sEMG) reflects the electrical activity of muscle fibres during contraction, and it has been widely used for intelligent prostheses or exoskeleton robotics control [1], [2]. To decode human intentions from sEMG more intuitively, artificial intelligence (AI) can be leveraged in either the classification-based hand gesture recognition [3], [4] or regression-based kinematic estimation [5], [6]. Different from the classification scheme which is only able to estimate discrete movements sequentially [7], regression approaches estimate continuous joint motions and can enable simultaneous and proportional control in multiple degrees of freedoms [8].

Recently, deep learning (DL) techniques, particularly convolutional neural networks (CNN), have gained considerable attentions to shift the paradigm of AI from conventional feature engineering to feature learning. For example, Ameri *et al.* proposed a CNN-based regression model which estimated wrist angles more accurately than support vector regression (SVR) and achieved better performances in the Fitts' law test [9]. Yang *et al.* investigated data-augmentation methods for CNN, and observed that CNN outperformed SVR significantly in the decoding of wrist kinetics [10], [11]. Moreover, CNN can also work as the deep feature extractor in the hybrid CNN-RNN (RNN denotes recurrent neural networks) scheme to further increase the estimation accuracy [12], [13]. However, these results are mainly obtained in laboratory settings which are simplistic and static. In fact, characteristics of sEMG can be easily influenced by external factors including muscle fatigue, electrode shift, impedance changes in electrode-skin interface, variations of contraction forces, and arm position effects, etc. [14]–[17]. In particular, sEMG signals have a user-specific nature, causing the amplitude and frequencies to be highly variable among individuals even when signals are measured from the same location with the same motion [18]. Although it has been reported that features learned by deep neural networks may be able to share similar distributions across different subjects [11], [19], the inter-subject problem can still lead to a sharp decline in the estimation performance of the previously trained model [11], [20].

Traditional DL approaches assume that training and testing data stem from the same underlying distribution. However, this assumption barely holds in practice [21], where the source domain D_S and target domain D_T have different feature spaces or marginal probability distributions [22], i.e. $D_T \neq D_S$. This issue is also known as domain shift. To this end, transfer learning (TL) has been investigated by exploiting knowledge learned in D_S and to effectively train DL models in D_T with insufficient labelled data. A simple but prevalent deep TL approach is fine-tuning (FT), where weights of a DL model developed in D_S are used as the starting points for

Manuscript received December 12, 2020; revised March 26, 2021; accepted May 27, 2021. Date of publication June 4, 2021; date of current version June 11, 2021. This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S019219/1, in part by the School of Electronic and Electrical Engineering, University of Leeds, and in part by the National Natural Science Foundation of China (NSFC) under Grant 61972302 and Grant 61962019. *(Corresponding authors: Pengfei Yang; Zhi-Qiang Zhang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the MaPS and Engineering Joint Faculty Research Ethics Committee of University of Leeds, U.K., under Approval No. MEEC 18-002.

Tianzhe Bao, Syed Ali Raza Zaidi, Shengquan Xie, and Zhi-Qiang Zhang are with the School of Electronic and Electrical Engineering, Institute of Robotics, Autonomous Systems and Sensing, University of Leeds, Leeds LS2 9JT, U.K. (e-mail: eltb@leeds.ac.uk; s.a.zaidi@leeds.ac.uk; s.q.xie@leeds.ac.uk; z.zhang3@leeds.ac.uk).

Pengfei Yang is with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: pfyang@xidian.edu.cn).

Digital Object Identifier 10.1109/TNSRE.2021.3086401

the model to be trained in D_T. FT has also been reported to enhance model training or adaptability in sEMG-based hand motion estimation. For instance, Wang *et al.* utilized FT in the training of a multimodal recurrent CNN. In this study, D_S data came from the NinaPro project [23], and D_T was composed of multimodal data collected from experiments. Ameri *et al.* employed FT to enhance CNN performances under the condition of electrode shift [24]. Experiments in both hand gesture recognition and wrist kinematics estimation verified the outperformance of FT when compared with a simple aggregation of pre-shift and post-shift sets. In addition, Kim *et al.* also fine-tuned the supportive CNN classifiers in the proposed subject-transfer framework, such that the estimation model can be more robust in terms of intra-user variability [25]. However, FT is prone to be overfitting when too few labelled data are available in the target domain [26]. Besides, a fine-tuned network usually suffers from catastrophic forgetting which destroys the model reusability [27].

Apart from FT, another popular TL scheme is domain adaptation (DA) which improves the target predictive function f(•) by exploiting the knowledge in D_S and D_T simultaneously. Compared with FT, DA not only reduces the demand for labelled target data but also enables consistent performances on different domains [21]. The main idea of DA is to align feature distributions of D_S and D_T in an embedding space. In practice, many efforts have followed the Siamese architecture [28], i.e. a two-stream CNN with shared weights. In this structure, one stream represents the source model and the other represents the target model. By adding additional discrepancy losses such as maximum mean discrepancy (MMD) [29]–[32], correlation alignment (CORAL) [33], or higher-order moments [34] etc. in model training, distribution divergences can be effectively minimized. Representative works include deep domain confusion (DDC) [29] and deep adaptation networks (DAN) [30]. In specific, DDC exploited a two-stream CNN and minimized MMD between outputs of the last fully-connected (FC) layer in each stream. DAN expanded DDC by employing multiple MMD terms to process outputs of several FC layers. Further advancements can be found in residual transfer network (RTN) [31] and joint adaptation networks (JAN) [32], etc. However, these approaches were mainly proposed to enhance CNN classification in computer vision (CV) tasks. To our best knowledge, few investigations have been conducted to address domain shift in sEMG-based kinematics estimation.

Inspired by the recent success of DA in deep learning, we propose a novel regression scheme for supervised domain adaptation (SDA) to reduce domain shift effects on CNN-based wrist kinematics estimation in the inter-subject circumstance. In our study, the source domain D_S denotes the source subjects which provide sufficient labelled data for CNN training, and target domain D_T represents the target subject to be tested using the pre-trained model. Specifically, a Siamese architecture is established to exploit both source and target data simultaneously, such that the domain-invariant features can be extracted. To tune CNN weights effectively, three types of loss functions are employed, including the regression losses for supervised learning in D_S and D_T, a MMD loss to reduce distribution mismatches between two domains in the latent space, and a regression contrastive loss to learn more discriminative deep features for domain alignment. In this study, eight healthy subjects were recruited to perform wrist flexion-extension. In order to stimulate the fast recalibration in myoelectric control, only a very short session in D_T was leveraged for the transfer learning. This setting is also consistent with many studies on supervised domain adaptation [35], [36]. Compared with conventional deep learning methods which require the labelled dataset to be sufficiently large in the target domain, SDA effectively reduces the burden of regular model re-training/recalibration under domain shift effects.

The remainder of this paper is structured as follows. Section II describes the regression SDA, where the model structure, the CNN layers and the loss functions are presented elaborately. Section III introduces experimental setups for wrist kinematics estimation. Section IV presents experimental results of regression SDA and baseline methods. The paper is discussed in Section V and then concluded in Section VI.

II. Methodology

In general, DA can be divided into supervised (SDA) and unsupervised (UDA) approaches [35]. In UDA, there are no labelled data in D_T, and existed works mainly focus on the alignment of feature distributions between domains. As for SDA, a small number of labelled samples in D_T can be utilized to build a bridge from sources to targets. Herein we prefer SDA since it can be more accurate in terms of the adaptation to large changes in sEMG signals [24]. The superiority in estimation performances have also been reported in CV tasks [36]. Specifically, in SDA we are given $D_S = \left\{\left(X_m^S, y_m^S\right)\right\}_{m=1}^{M}$, $D_T^{\text{train}} = \left\{\left(\boldsymbol{X}_n^T, y_n^T\right)\right\}_{n=1}^{N}$ and $D_T^{\text{test}} = \left\{\left(\boldsymbol{X}_l^T, y_l^T\right)\right\}_{l=1}^{L}$, where $\boldsymbol{X}$ denotes sEMG matrix extracted from raw signals and y is the related wrist angle (ground-truth). D_T^{train} participates in the model training together with D_S, whilst D_T^{test} is utilized to test SDA performances. It is noted that data in D_T^{train} are normally insufficient to train a conventional CNN, i.e. $N \ll L \approx M$.

A. Framework of Regression SDA

Following most efforts in DA [21], [29]–[32], [36], our regression SDA is also designed based on the two-stream CNN structure with shared weights. As depicted in Fig. 1, a pairwise sample $\left\{\left(\boldsymbol{X}_m^S, y_m^S\right), \left(\boldsymbol{X}_n^T, y_n^T\right)\right\}$ is imported into regression SDA, in which the first stream operates $\boldsymbol{x}_m^{\text{S}}$ and the second operates $\boldsymbol{x}_n^{\text{T}}$ separately. The construction of pairwise samples allows each target sample $\left(\boldsymbol{X}_n^T, y_n^T\right)$ to be paired with all source samples $\left(\boldsymbol{X}_m^S, y_m^S\right)$, which is able to effectively align the entire source data with the few target data [36]. This process can also be regarded as the Cartesian product of two datasets [37]. It is noted that the total number of pairwise samples, i.e. $M \times N$, will not be overlarge due to the size of D_T^{train}. In practice, the computational load can be further reduced by downsampling these pairwise samples for model training [36], [37]. Apart from the model structure, loss functions are also of vital importance to regression SDA. In this framework, regression losses and a domain discrepancy loss are combined to tune CNN, where the former is leveraged for supervised

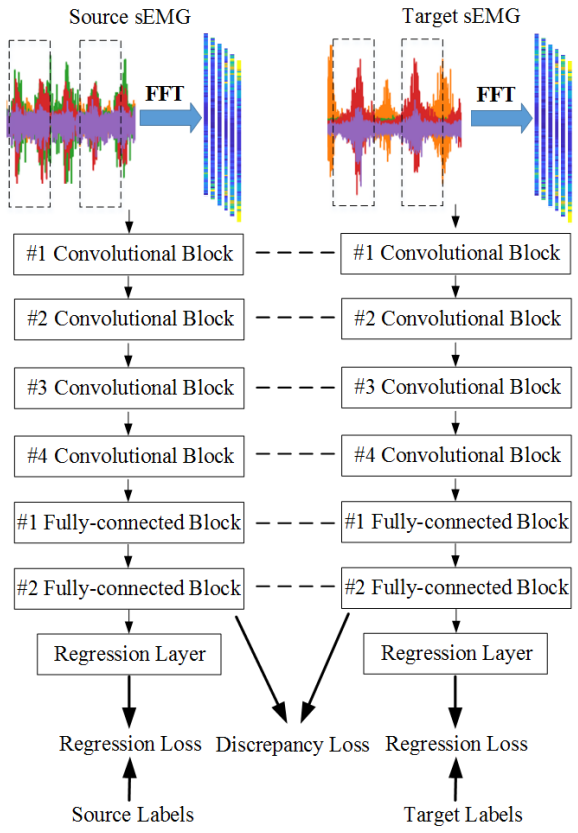

Fig. 1. Framework of regression SDA for kinematics estimation.

learning and the latter works to align feature distributions of source and target streams.

B. Design of CNN Stream

In CNN-based kinematics estimation, the sEMG matrix $\boldsymbol{X}$ is normally constructed from the pre-processed sEMG signals as the model input. Specifically, a sliding window method is utilized to obtain a segment in the size of $1 \times W \times C$, where W denotes the window length and C is the number of channels. As suggested in [13], the fast Fourier transform (FFT) is applied on the segment of each channel, and the spectrum matrix $\boldsymbol{X}$ can be finally obtained. As depicted in Fig. 1, the CNN stream consists of four convolutional blocks, two FC blocks and a regression layer. Each convolutional block is composed of a convolutional layer, a batch normalization layer, a leaky ReLU layer, a max-pooling layer and a dropout layer. The convolutional layer uses a kernel size of 3, a boundary padding of 1 and the stride of 1. There are 16 kernels in the 1st and 2nd convolutional block, whilst 32 in the 3rd and 4th block. In each FC block, a batch normalization layer, a leaky ReLU layer and a dropout layer are added subsequently to a FC layer. There are 100 hidden units in the 1st FC Block and 20 in the 2nd. Outputs of the 2nd FC Block will be utilized as deep features for to calculate domain discrepancy.

C. Design of Loss Functions

To tune CNN weights $\boldsymbol{\theta}$, the source label y_m^S and target label y_n^T are utilized to calculate the regression loss, i.e. mean square error (MSE), for each stream. Meanwhile, a domain discrepancy loss is also added to minimize the distribution divergence between two domains. Therefore, the optimal weights $\boldsymbol{\theta}^*$ can be learned by reducing the total loss which is formulated as

$$L\left(\boldsymbol{\theta} \mid \boldsymbol{X}^S, y^S, \boldsymbol{X}^T, y^T\right) = L_S + L_T + L_d \tag{1}$$

$$L_S = MSE\left(\boldsymbol{\theta} \mid \boldsymbol{X}^S, y^S\right) \tag{2}$$

$$L_T = MSE\left(\boldsymbol{\theta} \mid \boldsymbol{X}^T, y^T\right) \tag{3}$$

$$L_d = \gamma_1 MMD^2\left(\boldsymbol{\theta} \mid \boldsymbol{X}^S, \boldsymbol{X}^T\right) + \gamma_2 RContrastive\left(\boldsymbol{\theta} \mid \boldsymbol{X}^S, y^S, \boldsymbol{X}^T, y^T\right) \tag{4}$$

where L_S denotes the regression loss calculated in D_S whilst L_T represents the loss in D_T^{train}. L_d is the domain discrepancy loss combined of a MMD loss and a regression contrastive loss ($RContrastive$). In particular, $RContrastive$ is an expansion of the classification contrastive loss ($CContrastive$) originally designed to guarantee deep features with better intra-class compactness and inter-class separability in the latent space [36], [38]. Coefficients γ_1 and γ_2 are used to balance MMD loss and $RContrastive$ in model training.

1) MSE Loss: MSE loss is one of the most commonly used regression loss functions for supervised learning. It is the sum of squared distances between ground-truth and predictions:

$$MSE\left(\boldsymbol{\theta} \mid \boldsymbol{X}^S, y^S\right) = \frac{\sum_1^M \left(y_m^S - \widehat{y_m^S}\right)^2}{M} \tag{5}$$

$$MSE\left(\boldsymbol{\theta} \mid \boldsymbol{X}^T, y^T\right) = \frac{\sum_1^N \left(y_n^T - \widehat{y_n^T}\right)^2}{N} \tag{6}$$

where $\widehat{y_m^S}$ and $\widehat{y_n^T}$ denote the predicted wrist angles in the source and target stream, respectively.

2) MMD Loss: Given two sets of data drawn from two distributions, MMD measures the distance between the mean of these two sets after mapping each sample to a Reproducing Kernel Hilbert Space (RKHS) [39]. The empirical estimate of squared MMD is as follows

$$MMD^2\left(\boldsymbol{X}^S, \boldsymbol{X}^T\right) = \left\| \sum_{m=1}^{M} \frac{\varphi\left(\boldsymbol{x}_m^S\right)}{M} - \sum_{n=1}^{N} \frac{\varphi\left(\boldsymbol{x}_n^T\right)}{N} \right\|_{\mathcal{H}}^2 \tag{7}$$

where $\boldsymbol{x}_m^S$ and $\boldsymbol{x}_n^T$ represent the feature vectors extracted in the 2nd FC Block of the source and target stream, respectively. $\varphi(\bullet)$ indicates the mapping of the feature vectors to RKHS, and $\| \bullet \|_{\mathcal{H}}$ denotes the RKHS norm. In practice, Eq. (7) is usually calculated using kernel tricks, and the MMD loss can be further expressed as

$$MMD^2\left(\boldsymbol{\theta} \mid \boldsymbol{X}^S, \boldsymbol{X}^T\right) = \frac{\sum_{m,m*}^{M} k\left(\boldsymbol{x}_m^S, \boldsymbol{x}_{m*}^S\right)}{M^2} - \frac{\sum_{m,n}^{M,N} k\left(\boldsymbol{x}_m^S, \boldsymbol{x}_n^T\right)}{M \times N} + \frac{\sum_{n,n*}^{N} k\left(\boldsymbol{x}_n^T, \boldsymbol{x}_{n*}^T\right)}{N^2} \tag{8}$$

where $k(\bullet,\bullet)$ is a kernel function. Following most studies in DA, the standard RBF kernel is adopted such that MMD can compare all the orders of statistic moments [35]. As suggested in [26], the variance in RBF kernel is empirically set as 1.

3) Regression Contrastive Loss: $RContrastive$ is to learn more discriminative deep features in regression tasks. The basic idea is that samples from different domains but with similar kinematics should be mapped nearby in the latent space. On the contrary, dissimilar samples should be distant from each other. Therefore, $RContrastive$ is formulated as

$$\begin{aligned} &RContrastive\left(\boldsymbol{\theta} \mid \boldsymbol{X}^S, y^S, \boldsymbol{X}^T, y^T\right) \\ &= Y\left\|\boldsymbol{x}_m^S - \boldsymbol{x}_n^T\right\|^2 + (1-Y)\left\{\max\left(0, \sigma - \left\|\boldsymbol{x}_m^S - \boldsymbol{x}_n^T\right\|\right)\right\}^2 \end{aligned} \tag{9}$$

where $\|\bullet\|$ denotes the Frobenius norm, and σ is a margin to specify the separation of feature vectors in the embedding space. Y is the label defined for the similarity of a pairwise sample. As mentioned before, $RContrastive$ is the expansion of $CContrastive$ which is designed for the classification tasks [36], [38]. In $CContrastive$, Y can be denoted as a binary value determined by the rule: $Y = 1$ if the source and target data are from the same category; otherwise $Y = 0$. However, in regression tasks y_m^S and y_n^T are continuous values that cannot be assigned to specific categories. To address this issue, the computation of Y in $RContrastive$ is modified as

$$\mathrm{Y} = 1 - \frac{\left|y_m^S - y_n^T\right|}{\alpha} \tag{10}$$

where α denotes a constant which normalizes $\frac{\left|y_m^S - y_n^T\right|}{\alpha}$ into [0,1]. From Eq. (8) we can see that Y=1 if $y_m^S = y_n^T$. By contrast, Y will become smaller or even close to zero when y_m^S and y_n^T are dissimilar substantially.

To summarize, in the proposed method, both regression losses and discrepancy losses are employed to tune CNN weights $\boldsymbol{\theta}$: L_S and L_T utilize MSE losses to enable supervised learning in each domain, whilst L_d works as a regularization term to ensure that CNN can perform well in both domains. Specifically, L_d is combined of MMD loss and $RContrastive$, where MMD loss minimizes the distribution mismatch of two different domains in the latent space, and $RContrastive$ provides more discriminative deep features to further boost domain alignment. With these losses, $\boldsymbol{\theta}$ can be updated effectively using the backpropagation algorithm. The final weights $\boldsymbol{\theta}^*$ can be leveraged to estimate wrist kinematics in D_T^{test}. The overall process is summarized in Algorithm 1.

Algorithm 1 The Proposed Regression SDA

Input: Source domain dataset $D_S = \left\{\left(\boldsymbol{X}_m^S, y_m^S\right)\right\}_{m=1}^M$, target domain dataset $D_T^{\text{train}} = \left\{\left(\boldsymbol{X}_n^T, y_n^T\right)\right\}_{n=1}^N$, learning rate β, max training epochs T, loss coefficients γ_1 and γ_2, parameters σ and α.
Output: Optimal weights $\boldsymbol{\theta}^*$
1: Construct pairwise samples $\left\{\left(\boldsymbol{X}_m^S, y_m^S\right), \left(\boldsymbol{X}_n^T, y_n^T\right)\right\}$ based on D_S and D_T^{train}.
2: Initialize $\boldsymbol{\theta}$
3: while epoch $t < T$ do
4: $\quad \widehat{y_m^S}, \boldsymbol{x}_m^S \leftarrow CNN\left(\boldsymbol{X}_m^S, \boldsymbol{\theta}(t)\right)$
5: $\quad y_n^T, \boldsymbol{x}_n^T \leftarrow CNN\left(\boldsymbol{X}_n^S, \boldsymbol{\theta}(t)\right)$
6: $\quad$ Calculate L_S, L_T and L_d based on Eq. (5)-(10)
7: $\quad \boldsymbol{\theta}(t+1) \leftarrow \boldsymbol{\theta}(t) - \beta\nabla_{\boldsymbol{\theta}}\left(L_S + L_T + L_d\right)$
8: end while
9: return $\boldsymbol{\theta}^*$

D. Baseline Methods

To demonstrate the effectiveness of regression SDA, we further compare it with several baseline methods. The descriptions of these methods are as follows.

1) Source Only (SO): SO simulates the implementation of a pre-trained CNN in the TL process $D_S \rightarrow D_T$, where only data in D_S are utilized for supervised learning.

2) Target Only (TO): TO represents the conventional training of CNN using D_T^{train}, in which the network weights are randomly initialized.

Similar to regression SDA, SO and TO are also trained using pairwise samples reconstructed from D_S and D_T^{train}. However, only L_S is adopted in SO, whereas TO utilizes L_T instead. This strategy can also be regarded as a data augmentation approach for deep learning [41].

3) Joint Training (JT): JT shares the same architecture with SDA but the discrepancy loss is excluded. It can be considered as a TL/DA approach which attempts to exploit information in both D_S and D_T.

4) Fine-Tuning (FT): As aforementioned, FT is the simplest but most prevalent TL approach in deep learning applications. Following previous research [24], convolutional layers are transferred from a CNN that is pre-trained in D_S as the initial values for a new model to be trained in D_T^{train}.

5) Ordinary Least Square (OLS): Since the least square based approaches do not heavily rely on the size of training data and computation resources, the OLS model is also included for comparison. Similar to TO, OLS is also trained using D_T^{train}. As suggested by previous studies [42]–[44], several temporal-spatial features are extracted from sEMG, including mean absolute value (MAV), root mean square (RMS), variance (VAR), and fourth-order autoregressive coefficients (4th AR). In case of overfitting, the principal component analysis (PCA) is applied to reduce redundant hand-crafted features.

III. Materials and Experimental Methods

A. Experiment Setup

Approved by the MaPS and Engineering joint Faculty Research Ethics Committee of University of Leeds, UK (MEEC 18-002), six males and two females (aged 25 to 31) participated in this experiment. A written informed consent was obtained from each subject.

1) Experiment Protocol: As shown in Fig. 2, subjects seat on the armchair, with torso fully straight and forearm relaxed. The current position of hand was set as the neutral position.

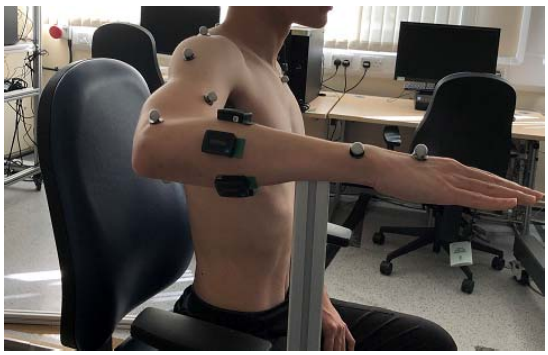

Fig. 2. The placement of electrodes and markers in data acquisition [40]. This figure also illustrates the neutral position in the wrist rotation.

In data collection, participants were asked to perform wrist flexion/extension following a continuous cycle trial: the wrist was rotated from neutral position to the flexion direction, it was then moved back to the extension direction and finally returned to neutral position. Each trial lasted around 20s and 5 trials were recorded for each participant.

2) Acquisition of sEMG: Delsys TrignoTM system was used to record sEMG signals. Following SENIAM recommendation [45], electrodes were placed over five primary wrist muscles over right forearm: Flexor Carpi Radialis (FCR), Flexor Carpi Ulnaris (FCU), Extensor Carpi Radialis Longus (ECRL), Extensor Carpi Radialis Brevis (ECRB) and Extensor Carpi Ulnaris (ECU). The sampling rate was set as 2000 Hz.

3) Motion Capture: To capture wrist movements through the motion capture system (Vicon Motion Systems Ltd. UK), 16 reflective markers were placed on the subject's right upper limb. As illustrated in Fig. 4, markers were allocated over the spinous process of the 7th and the 10th thoracic vertabra, right scapula, xiphoid, acromio-clavicular joint, clavicle, lateral/medial humerus medial epicondyle, right radial/ulnar styloid, middle forearm and the right third metacarpus. The sampling rate of Vicon Motion System was 250 Hz and the synchronization of the kinematic data and sEMG were conducted using a trigger module. The Vicon upper limb model were applied to calculate wrist joint angles as the measured angles or ground truth.

B. Data Pre-Processing

Collected sEMG signals were pre-processed using a 3rd order Butterworth high pass filter (20 Hz) to remove movement artifacts [46] and a low pass filter (450 Hz) to remove unusable high frequency noise [47]. To extract samples for CNN, the size of sliding windows was set to be 100ms length with 50ms increment. Since subjects were asked to rotate their wrists in a comparatively low speed, the label of a sample was obtained by computing the mean value of measured angles within a sliding window. Besides, samples of each individual were normalized by dividing the peak value of each given muscle in the isometric maximum voluntary contraction [48].

C. Hyper-Parameter Setting

In this study, the two-stream network was trained in a 32 sized mini-batch for 100 epochs via adaptive moment estimation (ADAM). The dynamic learning rate was 0.001. The slope scale of leaky ReLU layers was set as 0.1. The max-pooling layer used a pool size of 3, whilst the dropout rate was set to be 30%. Following [36], we also set $\sigma = 1$ for $RContrastive$. Based on experiment protocols, α in Eq. (10) was set to be 180 since wrist rotations were normally within $[-90°, 90°]$. In addition, we empirically set $\gamma_1 = 1000$ and $\gamma_2 = 0.1$. The training of the network was implemented using Pytorch backend.

D. Model Evaluation

Two commonly applied metrics, i.e. the normalized root mean square error (NRMSE) and the coefficient of determination (R^2) were used to evaluate the performances of regression SDA. In particular, NRMSE and R^2 indicate the difference in terms of amplitude and correlation between the estimated kinematics and ground-truth, respectively. Specifically, NRMSE is defined as [49]

$$NRMSE = \frac{RMSE}{y_{\max} - y_{\min}} = \frac{\sqrt{\sum_{t=1}^{N} \frac{(y_t - \widehat{y_t})^2}{N}}}{y_{\max} - y_{\min}} \tag{11}$$

where y_t is the actual value of sample t, $\widehat{y_t}$ is the estimated value, N is the total number of samples for evaluation, $y_{\max}$ and $y_{\min}$ are the maximum and minimum of the actual values, respectively.

The mathematical expression of R^2 [50] is

$$R^2 = 1 - \frac{\sum_{t=1}^{N} (y_t - \widehat{y_t})^2}{\sum_{t=1}^{N} (y_t - \overline{y_t})^2} \tag{12}$$

where $\bar{y}_t$ is the mean of y_t. The R^2 of a perfect estimation is close to 1, and it becomes negative if the square sum of estimation errors are larger than the variance of ground-truth. In this study we compared SDA with baseline methods in both single-single and multiple-single scenarios of kinematics estimation. Since this study focused on the inter-subject transfer learning, the dataset of each subject was categorized as either the source or target domain for each TL process, which thus resulted in 56 processes in the single-single scenario. For the sake of simplicity, we use $D_{Si} \rightarrow D_{Tj}$ $(i, j = 1, 2, \ldots, 8, i \neq j)$ to define the TL process from source subject i to target subject j. To guarantee a sufficient training, data in five experiment trials of a source subject were combined to construct a comparatively large D_S. Differently, in the multiple-single scenario, the inter-user data are leveraged for model training before testing on a new participant. Assuming that more general and informative features could be learned by CNN based on data aggregated from multiple individuals, this scenario is also prevalent in the inter-subject evaluations of TL approaches. In our experiment, for each D_{Tj}, the corresponding D_S was composed of data from the rest seven subjects.

According to the settings of regression SDA [35], [36], i.e. sufficient labeled training data in the source domain and sparse ones in target, D_T^{train} was composed of only 10% data collected in one experimental trial (about $2 \sim 3$ sec to cover a wrist contraction circle from extension to flexion), whilst the rest data of this trial are stored in D_T^{test} for evaluation. Besides,

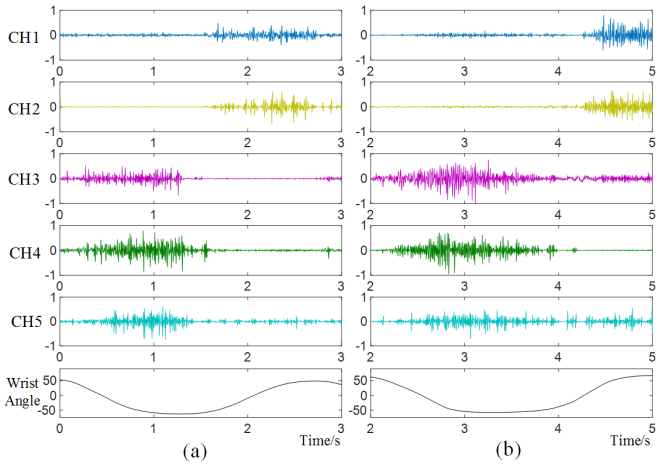

Fig. 3. Normalized sEMG signals and wrist angles of (a) subject 6 and (b) subject 1 in a rotation cycle. The wrist angles are measured in degrees. The channel numbers and measured muscles were consistent among all subjects: CH1-FCR, CH2-FCU, CH3-ECRL, CH4-ECRB, CH5-ECU. As shown in this figure, muscle activations varied dramatically among two subjects. In particular, ECU of subject 6 was mainly activated during wrist extension. By contrast, high activations can be found in ECU of subject 1 during flexion.

to reduce the computational load, only the first 20% of the shuffled pairwise samples were utilized for model training. This procedure is similar to the ratio filter applied in [36], [37]. Empirically, we observed that the estimation accuracies when using downsampled training sets were close to those when all $M \times N$ pairwise samples were involved.

IV. Results

A. Domain Shift Effects on Inter-Subject Estimation

Fig. 3 demonstrates sEMG signals and related wrist angles of two subjects in kinematics estimation. Amplitudes of sEMG in each channel indicate the activation levels of the measured muscle. As we can see, in some channels (such as CH5) sEMG patterns can differ substantially among subjects even though wrist motions are similar. We then validated the performances of a conventional CNN model (see Section II.B) in both intra-subject (D_S and D_T are from one subject) and inter-subject (D_S and D_T are from two different subjects) circumstance. As shown in Fig. 4, the validation loss decreases effectively in the former circumstance but can hardly converge in the latter.

B. Learning Process of Regression SDA

In this section, we investigated the learning process of regression SDA in the inter-subject domain adaptation. Fig. 5 illustrates convergences of L_S, L_T and L_d in the TL process $D_{S6} \rightarrow D_{T1}$. From this figure it can be observed that two regression losses and the domain discrepancy loss could decrease simultaneously via backpropagation. Different from iteration performances in Fig. 4, the convergence of L_S was substantially restricted due to the regularization of L_d, which helped CNN to avoid overfitting to the low-error regions of D_S. Besides, we can also find that L_T decreased faster than L_S even though their coefficients were set to be the

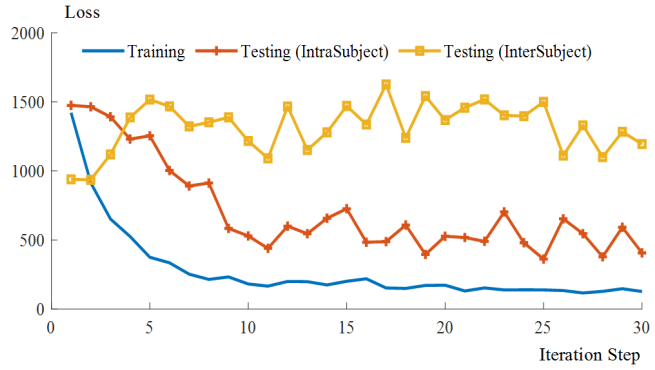

Fig. 4. Loss performances of CNN during model learning in both intra-subject and inter-subject scenarios.

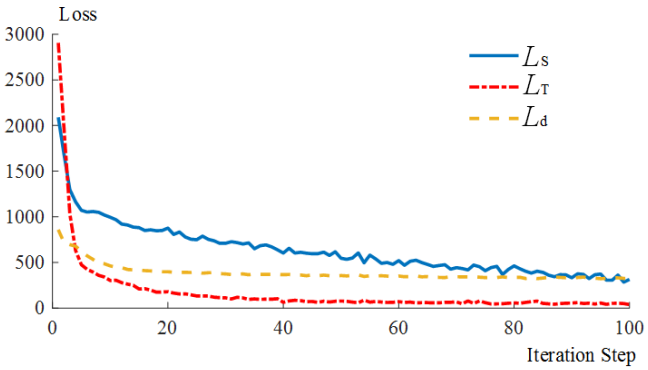

Fig. 5. Loss performances of regression SDA during model learning in the TL process $D_{S6} \rightarrow D_{T1}$. Specifically, L_S denotes the regression loss calculated in D_S, L_T represents the loss in D_T^{train}, L_d is the domain discrepancy loss combined of a MMD loss and the *RContrastive* loss.

same. Similar observations can also be found in many other TL processes in our experiment. A possible reason is that, although samples in D_T^{train} were extremely augmented in the pairwise combinations, the information provided by D_T^{train} is much less than D_S due to its limited size.

C. Estimation Performances in Single-Single TL Process

To illustrate the regression performances intuitively, Fig. 6 plots the estimated trajectories of all listed methods in the transfer process $D_{S6} \rightarrow D_{T1}$. The absolute error of each method with regard to the ground-truth are also summarized in the histograms accordingly. From Fig. 6 it can be observed that, due to domain shift effects in the inter-subject circumstance, the predicted trajectory of SO is quite distant from the ground-truth. The absolute errors of testing samples are substantially larger than other those of other methods. By contrast, trajectories of JT, FT and regression SDA are much closer to the ground-truth, which can be also verified by the their better distributions of absolute errors. In particular, the trajectory of SDA can mostly fit the ground-truth, with absolute errors mainly smaller than 50 degrees.

For an explicit comparison, Table I summarizes the NRMSE and R^2 of SO, JT, FT and SDA in the single-single TL processes targeted at Subject 1, i.e. $D_{Si} \rightarrow D_{T1} (i = 2, 3, \ldots, 8)$. Besides, according to the definition of TO and

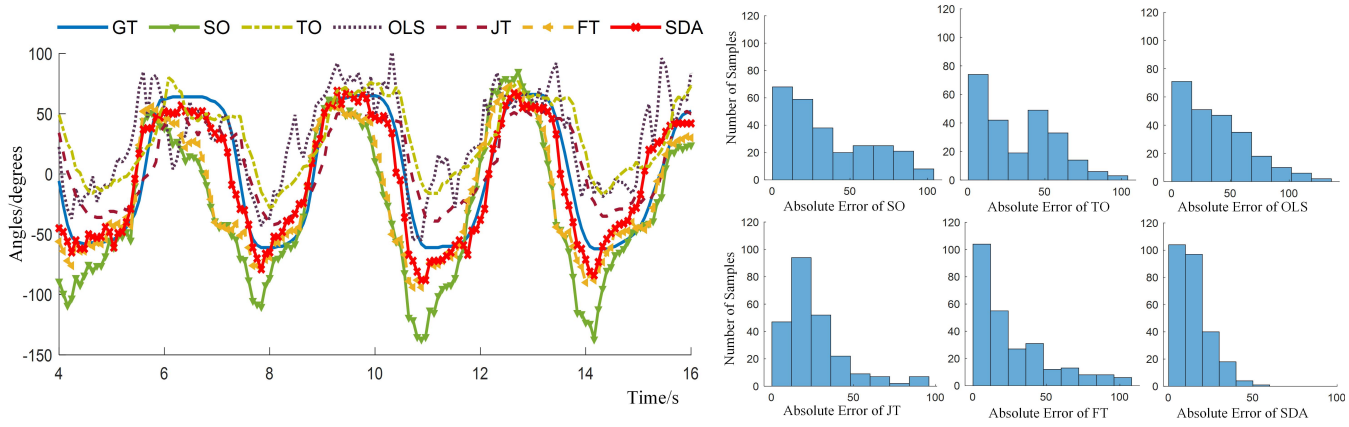

Fig. 6. Estimation performances (predicted wrist angles and absolute errors with respect to the ground-truth) of all methods in the TL process $D_{S6} \rightarrow D_{T1}$. GT denotes the ground-truth. The NRMSE of SO, JT, FT and SDA are 0.35, 0.26, 0.25, 0.22 in $D_{S6} \rightarrow D_{T1}$, and the R^2 of these four methods are 0.38, 0.59, 0.59, 0.72, respectively. The NRMSE/R^2 of TO and OLS in D_{T1} (TO and OLS are calculated once in each target subject) are 0.32/0.43 and 0.30/0.51.

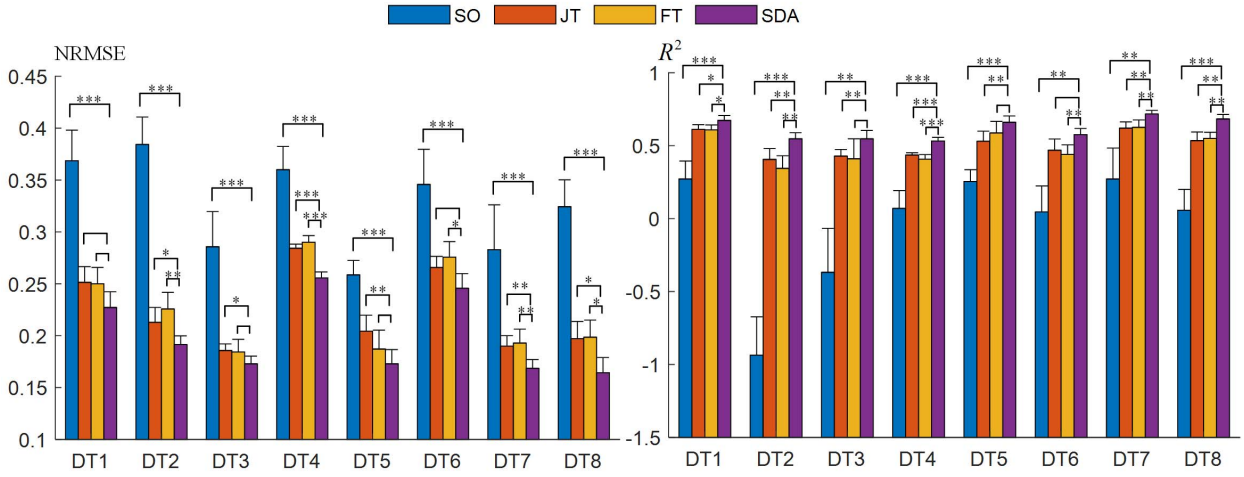

Fig. 7. Statistical analysis of SO, JT, FT and regression SDA for each target subject in the single-single transfer learning scenario (***p-value $<$ 0.001, **p-value $<$ 0.05, and *p-value $<$ 0.1). TO and OLS are excluded in this figure since they are computed once in each target subject).

TABLE I

NRMSE AND R^2 OF SO, JT, FT AND SDA IN TL PROCESSES $D_{Si} \rightarrow D_{T1}$ $(i = 2, 3, \ldots, 8)$

Target	Metric	Method	D_{S2}	D_{S3}	D_{S4}	D_{S5}	D_{S6}	D_{S7}	D_{S8}	Ave	Std
D_{T1}	NRMSE	SO	0.38	0.34	0.49	0.32	0.35	0.32	0.38	0.37	0.06
		JT	0.30	0.27	0.23	0.21	0.26	0.26	0.23	0.25	0.03
		FT	0.30	0.27	0.24	0.20	0.26	0.25	0.23	0.25	0.03
		SDA	0.27	0.25	0.20	0.19	0.20	0.22	0.22	0.22	0.03
	R^2	SO	0.21	0.39	-0.24	0.46	0.38	0.46	0.24	0.27	0.25
		JT	0.52	0.55	0.66	0.7	0.59	0.6	0.66	0.61	0.06
		FT	0.51	0.55	0.62	0.71	0.59	0.62	0.66	0.61	0.07
		SDA	0.55	0.61	0.72	0.73	0.72	0.69	0.69	0.67	0.07

OLS, these two methods are calculated once in each target subject. The NRMSE/R^2 of TO and OLS in D_{T1} (TO and OLS) are 0.32/0.43 and 0.30/0.51, respectively. Since trajectories of SO usually differ a lot from the ground-truth (see Fig. 6) due to the domain shift impact, the R^2 of SO in some TL processes can be negative, such as -0.24 in $D_{S4} \rightarrow D_{T1}$. From Table I it can be observed that SDA surpasses other methods, especially JT and FT in most cases. Another interesting observation is that in the same row TL performances also vary a lot. This is because the domain shift effects cannot be the same between every two subjects due to different similarities in their biochemical or physiological characteristics.

To better verify the effectiveness of regression SDA, the one-way analysis of variance (ANOVA) is applied for statistical analysis of SO, JT, FT, and SDA in each target subject, and the results can be found in Fig. 7. As aforementioned, the performances of each method vary substantially among TL processes in each target subject,

TABLE II
NRMSE AND R^2 OF ALL LISTED METHODS IN TL PROCESSES OF MULTIPLE-SINGLE SCENARIO

Metric	Target	SO	TO	OLS	JT	FT	SDA
NRMSE	D_{T1}	0.33	0.32	0.30	0.23	0.22	0.19
	D_{T2}	0.31	0.27	0.25	0.19	0.21	0.17
	D_{T3}	0.25	0.21	0.26	0.17	0.17	0.14
	D_{T4}	0.31	0.30	0.28	0.26	0.26	0.23
	D_{T5}	0.24	0.23	0.23	0.19	0.19	0.16
	D_{T6}	0.29	0.30	0.27	0.26	0.25	0.23
	D_{T7}	0.23	0.26	0.28	0.18	0.18	0.15
	D_{T8}	0.28	0.25	0.18	0.20	0.21	0.18
R^2	D_{T1}	0.41	0.43	0.51	0.66	0.67	0.73
	D_{T2}	-0.20	0.08	0.23	0.54	0.43	0.60
	D_{T3}	-0.06	0.25	0.22	0.55	0.50	0.68
	D_{T4}	0.33	0.35	0.46	0.39	0.56	0.60
	D_{T5}	0.34	0.37	0.41	0.58	0.61	0.70
	D_{T6}	0.34	0.31	0.44	0.46	0.50	0.62
	D_{T7}	0.42	0.29	0.24	0.70	0.67	0.76
	D_{T8}	0.17	0.41	0.58	0.52	0.52	0.66

which can result in large standard deviations (Std). To this end, we choose larger p-values to indicate the significance in statistics (***p-value $<$ 0.001, **p-value $<$ 0.05, and *p-value $<$ 0.1). Since TO and OLS was computed once in each target subject, statistical analysis of TO/OLS were only included in the multiple-single scenario (see Section IV.D).

D. Estimation Performances in Multiple-Single TL Process

In this subsection, performances of SO, TO, OLS, JT, FT and SDA are compared in multiple-single processes. Table II lists NRMSE and R^2 of all listed methods in the multiple-single scenario, where for each target subject $D_{Ti}(i = 1 \cdots 8)$, the corresponding D_S is composed of data from the rest seven subjects. To fully exploit the capability of neural network methods, the leave-one-out cross-validation (LOOCV) is applied to tune hyper-parameters, where data of each source subject work alternatively as the validation subject in each TL process. Herein, four hyper-parameters are selected to be optimized in SDA, including learning rate β, max training epochs T, loss coefficients γ_1 and γ_2. Each hyper-parameter is given five optional values, i.e. $\beta \in [0.1, 0.05, 0.001, 0.0005, 0.0001]$, $T \in [25, 50, 100, 150, 200]$, $\gamma_1 \in [100, 500, 1000, 2000, 5000]$, and $\gamma_2 \in [0.01, 0.05, 0.1, 0.5, 1.0]$. After LOOCV, the configured hyper-parameters are applied to SDA for the target subject. Statistical analysis is shown in Fig. 8, from which it can be indicated that regression SDA still obtains the best estimation performance in each process.

E. Estimation Performances in Source Domains

Apart from better results in D_T, another main advantage of regression SDA over FT is that the former can maintain estimation performances in D_S. Theoretically, without extra guidance to tune CNN weights $\boldsymbol{\theta}$ for original tasks, FT only learns a final point $\boldsymbol{\theta^*}$ that yields a low error for D_T but not D_S. This issue is also known as the catastrophic forgetting

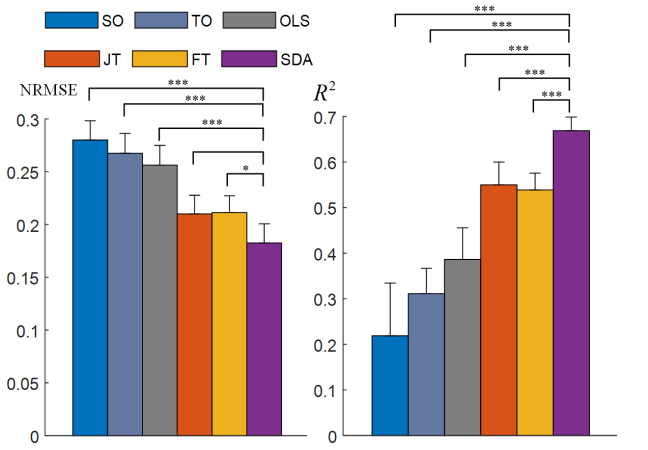

Fig. 8. Statistical analysis of SO, TO, OLS, JT, FT and SDA on eight target subjects in the multiple-single scenario (***p-value $<$ 0.001, **p-value $<$ 0.05, *p-value $<$ 0.1).

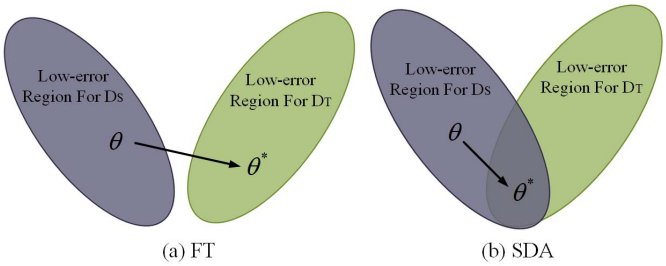

Fig. 9. Optimization of CNN weights via FT and regression SDA. The low-error region of two domains will become closer or more overlapped in regression SDA due to the reduction of domain discrepancy.

[26], [27] which is prevalent in conventional deep neural networks when learning new tasks. On the contrary, due to the special design of loss functions in regression SDA, $\boldsymbol{\theta^*}$ can be kept in the low-error regions of both D_T but not D_S. The differences in the optimization of CNN weights $\boldsymbol{\theta}$ via FT and regression SDA are illustrated in Fig. 9.

In Fig. 10 we further compare the overall performances of FT and regression SDA on each source domain D_{Si} after conducting TL processes to the rest seven target domains of single-single scenario. As we can see, performances of FT on D_{Si} degraded substantially due to the catastrophic forgetting [27]. By contrast, performances of regression SDA can be maintained in a much better level for every D_{Si}. Compared with FT, this advantage of regression SDA can effectively boost the model reusability among subjects. Although it is claimed that after FT a specific network could be stored for each subject separately, this strategy might be impractical in real-time applications since an extra step is then required to distinguish which subject the testing data should belong to.

V. DISCUSSION

Domain shift issues are prevalent in sEMG-based motion estimation, particularly when DL models are implemented in the inter-subject circumstance. As illustrated in Fig. 3, a main reason is that the physiological, anatomical and biochemical characteristics of muscles are highly variable among individuals. In addition, subjects may use different muscle

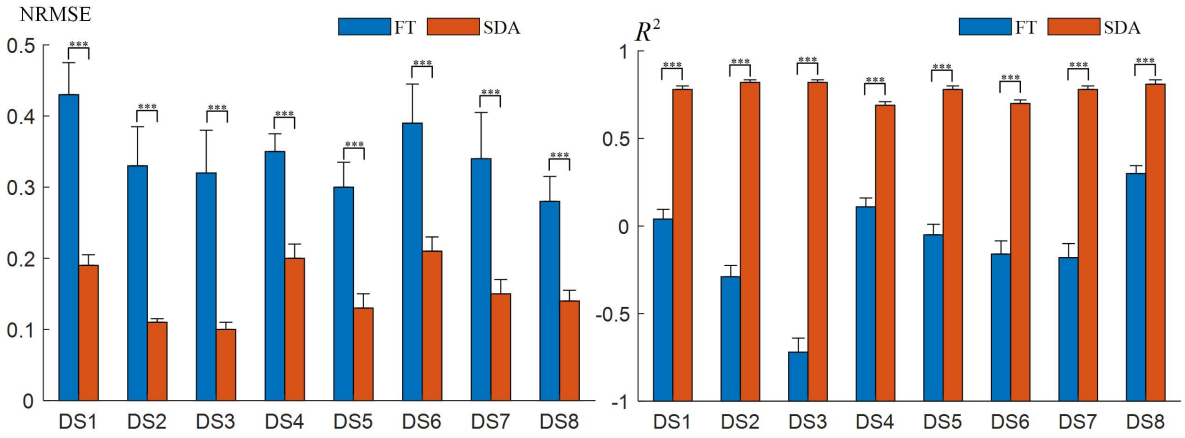

Fig. 10. Statistical analysis of FT and regression SDA for each source subject in the single-single transfer learning scenario (***p-value $<$ 0.001).

control strategies to produce the same movement [18], [48]. Consequently, results of SO in our experiments (Fig. 6-Fig.8, Table I and II) indicate that models trained with sEMG from previous subjects may fail to predict accurately on a new subject, which results in a great challenge to the practical application of myoelectric control. In previous literatures many efforts have been reported to enhance the model generalization among individuals, including both machine learning approaches [51]–[55] and deep learning ones [25], [56], [57]. However, most of these works mainly focused on the hand gestures recognition rather than kinematics estimation, where specific designs were proposed for the classier or to match classification strategies.

To this end, we propose the regression SDA to reduce domain shift effects on CNN performances in the inter-subject kinematics estimation. According to the experiment results in both single-single and multiple-single scenario it can be concluded that 1) by exploiting information of both source and target domains, the proposed SDA can outperform baseline methods significantly; 2) with help of the discrepancy losses, SDA can further surpass JT which simply combines labelled data of two domains; 3) due to the effectiveness of automatic feature extraction via CNN, deep learning methods, i.e. JT/FT/SDA, are better than OLS which depends heavily on the quality of hand-crafted features; 4) different from FT which suffers from the catastrophic forgetting in the source domains, SDA can maintain good performances in two domains and thus boost the model reusability among subjects.

Another interesting observation of SDA is that D_T^{train} and D_T^{test} might be able to come from different tasks. In particular, participants were asked to perform two contractions: 1) wrist flexion (WF) to move the wrist towards to the palm side and then return to neutral position; 2) wrist extension (WE), which starts from neutral position, move the wrist towards to the back-hand side and then return to neutral position. Apparently, both features (muscle activations denoted by sEMG) and labels (wrist angles) are different between two tasks. Herein, both D_S and D_T^{test} are composed of data from WE, whilst D_T^{train} is obtained from WF. In our experiment, we empirically found that TO, OLS and FT all performed very poorly since they only utilized D_T^{train} which is irrelevant to D_T^{test}. Therefore,

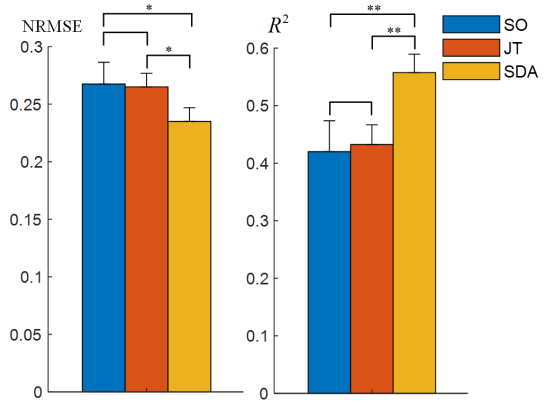

Fig. 11. Statistical analysis of SO, JT and SDA in multiple-single TL processes when D_T^{train} and D_T^{test} are from two tasks (**p-value $<$ 0.05, *p-value $<$ 0.1). Specifically, D_S and D_T^{test} are composed of data from wrist extension, whilst D_T^{train} is obtained from wrist flexion.

we mainly compared the performances of SO, JT and SDA. Figure 11 illustrates the statistical results in the multiple-single scenario, from which it can be found that SDA outperforms both JT and SO significantly (p-value $<$ 0.05 for R^2). Interestingly, there is no significance between JT and SO, indicating that a simple addition of the irrelevant D_T^{train} may not benefit the transfer learning between different tasks.

As aforementioned, domain adaptation can be divided into SDA and UDA approaches [35]. Different from SDA, UDA is also be of significance due to the exclusion of extra hardware and time for data relabelling. In fact, our method is a framework which is suitable for both SDA and UDA. Based on the proposed two-stream CNN architecture, a UDA model can be constructed when only the source regression loss L_S and MMD Loss are included. This setting is similar to those proposed in [29], [30]. However, we empirically observed that although UDA can continuously outperform SO, its performances are significantly inferior to those of SDA, mainly due to the lack of target labels to provide discriminative information. Similar statements can also be found in many relative works of myoelectric control [24], [52] and other research fields [36], [58].

Currently, the costs of hardware set-up and computations during recalibration is still a limitation of the proposed SDA, and there might be some solutions to address these disadvantages. Firstly, quantization approaches [59] have been widely investigated in recent years to reduce the computation load of CNN models, and will be further explored to enhance SDA approaches in our future work. Secondly, the hardware setups in this experiment can also be further simplified by using the armband for sEMG detection [57] and Leap Motion Controller [60] which provides a cheap and efficient way to track the joint angles as labels. With the acceleration of computation and the simplification of hardware set-up, it can further benefit users by requesting fewer trials for recalibration via SDA.

VI. Conclusion

In this study, we propose the regression SDA to reduce domain shift effects on CNN performances in the inter-subject circumstance. Based on the two-stream structure, data in both source and target subject can be exploited simultaneously. By adding the discrepancy loss in model training, distribution divergences between two domains can be effectively minimized. The main merit of regression SDA compared with fine-tuning can be summarized as 1) it further improves the estimation accuracy with very limited data in the target domain; 2) it also maintains good performance in original domain and thus boosts the model reusability.

References

[1] J. M. Hahne, M. A. Schweisfurth, M. Koppe, and D. Farina, "Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users," *Sci. Robot.*, vol. 3, no. 19, Jun. 2018, Art. no. eaat3630.

[2] S. Pasinetti, M. Lancini, I. Bodini, and F. Docchio, "A novel algorithm for EMG signal processing and muscle timing measurement," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 11, pp. 2995–3004, Nov. 2015.

[3] J. Maier, A. Naber, and M. Ortiz-Catalan, "Improved prosthetic control based on myoelectric pattern recognition via wavelet-based de-noising," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 506–514, Feb. 2018.

[4] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao, "Surface EMG hand gesture recognition system based on PCA and GRNN," *Neural Comput.*, vol. 32, pp. 6343–6351, May 2020.

[5] K. Bakshi, M. Manjunatha, and C. S. Kumar, "Estimation of continuous and constraint-free 3 DoF wrist movements from surface electromyogram signal using kernel recursive least square tracker," *Biomed. Signal Process. Control*, vol. 46, pp. 104–115, Sep. 2018.

[6] J. Liu, Y. Ren, D. Xu, S. H. Kang, and L.-Q. Zhang, "EMG-based real-time linear-nonlinear cascade regression decoding of shoulder, elbow, and wrist movements in able-bodied persons and stroke survivors," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 5, pp. 1272–1281, May 2020.

[7] S. Benatti, F. Montagna, V. Kartsch, A. Rahimi, D. Rossi, and L. Benini, "Online learning and classification of EMG-based gestures on a parallel ultra-low power platform using hyperdimensional computing," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 3, pp. 516–528, Jun. 2019.

[8] A. W. Shehata, E. J. Scheme, and J. W. Sensinger, "Evaluating internal model strength and performance of myoelectric prosthesis control strategies," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 5, pp. 1046–1055, May 2018.

[9] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, "Regression convolutional neural network for improved simultaneous EMG control," *J. Neural Eng.*, vol. 16, no. 3, Jun. 2019, Art. no. 036015.

[10] W. Yang, D. Yang, J. Li, Y. Liu, and H. Liu, "EMG dataset augmentation approaches for improving the multi-DOF wrist movement regression accuracy and robustness," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 1268–1273.

[11] W. Yang, D. Yang, Y. Liu, and H. Liu, "Decoding simultaneous multi-DOF wrist movements from raw EMG signals using a convolutional neural network," *IEEE Trans. Human-Mach. Syst.*, vol. 49, no. 5, pp. 411–420, Oct. 2019.

[12] P. Xia, J. Hu, and Y. Peng, "EMG-based estimation of limb movement using deep learning with recurrent convolutional neural networks," *Artif. Organs*, vol. 42, no. 5, pp. E67–E77, May 2018.

[13] T. Bao, S. A. R. Zaidi, S. Xie, P. Yang, and Z.-Q. Zhang, "A CNN-LSTM hybrid model for wrist kinematics estimation using surface electromyography," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[14] E. C. Hill *et al.*, "Effect of sex on torque, recovery, EMG, and MMG responses to fatigue," *J. Musculoskelet Neuronal Interact.*, vol. 16, no. 4, pp. 310–317, 2016.

[15] L. Pan, D. Zhang, N. Jiang, X. Sheng, and X. Zhu, "Improving robustness against electrode shift of high density EMG for myoelectric control through common spatial patterns," *J. NeuroEng. Rehabil.*, vol. 12, no. 1, p. 110, Dec. 2015.

[16] R. Kusche and M. Ryschka, "Combining bioimpedance and EMG measurements for reliable muscle contraction detection," *IEEE Sensors J.*, vol. 19, no. 23, pp. 11687–11696, Dec. 2019.

[17] A. Waris, I. K. Niazi, M. Jamil, K. Englehart, W. Jensen, and E. N. Kamavuako, "Multiday evaluation of techniques for EMG-based classification of hand motions," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 4, pp. 1526–1534, Jul. 2019.

[18] C. J. De Luca, "Surface electromyography: Detection and recording," *DelSys Incorporated*, vol. 10, no. 2, pp. 1–10, 2002.

[19] L. Xu, X. Chen, S. Cao, X. Zhang, and X. Chen, "Feasibility study of advanced neural networks applied to sEMG-based force estimation," *Sensors*, vol. 18, no. 10, p. 3226, Sep. 2018.

[20] D. Xiong, D. Zhang, X. Zhao, and Y. Zhao, "Deep learning for EMG-based human-machine interaction: A review," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 3, pp. 512–533, Mar. 2021.

[21] C. Yu *et al.*, "Learning to match distributions for domain adaptation," 2020, *arXiv:2007.10791*. [Online]. Available: http://arxiv.org/abs/2007.10791

[22] Y.-P. Lin and T.-P. Jung, "Improving EEG-based emotion classification using conditional transfer learning," *Frontiers Hum. Neurosci.*, vol. 11, p. 334, Jun. 2017.

[23] W. Wang, B. Chen, P. Xia, J. Hu, and Y. Peng, "Sensor fusion for myoelectric control based on deep learning with recurrent convolutional neural networks," *Artif. Organs*, vol. 42, no. 9, pp. E272–E282, Sep. 2018.

[24] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, "A deep transfer learning approach to reducing the effect of electrode shift in EMG pattern recognition-based control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 2, pp. 370–379, Feb. 2020.

[25] K.-T. Kim, C. Guan, and S.-W. Lee, "A subject-transfer framework based on single-trial EMG analysis using convolutional neural networks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 94–103, Jan. 2020.

[26] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 801–814, Apr. 2019.

[27] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[28] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 539–546.

[29] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: http://arxiv.org/abs/1412.3474

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

[31] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv Neural Inf Process Syst*, 2016, pp. 136–144.

[32] M. Long, H. Zhu, J. Wang, and M. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[33] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 443–450.

[34] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," 2017, *arXiv:1702.08811*. [Online]. Available: http://arxiv.org/abs/1702.08811

[35] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[36] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5715–5725.

[37] L. H. Morsing, O. A. Sheikh-Omar, and A. Iosifidis, "Supervised domain adaptation using graph embedding," 2020, *arXiv:2003.04063*. [Online]. Available: http://arxiv.org/abs/2003.04063

[38] C. Chen, Z. Chen, B. Jiang, and X. Jin, "Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3296–3303.

[39] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, Jul. 2006.

[40] Y. Zhao, Z. Zhang, Z. Li, Z. Yang, A. A. Dehghani-Sanij, and S. Xie, "An EMG-driven musculoskeletal model for estimating continuous wrist motion," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 12, pp. 3113–3120, Dec. 2020.

[41] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1909–1921, Apr. 2018.

[42] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "EMG feature evaluation for improving myoelectric pattern recognition robustness," *Expert Syst. Appl.*, vol. 40, no. 12, pp. 4832–4840, Sep. 2013.

[43] S. Muceli and D. Farina, "Simultaneous and proportional estimation of hand kinematics from EMG during mirrored movements at multiple degrees-of-freedom," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 3, pp. 371–378, May 2012.

[44] B. Yu, X. Zhang, L. Wu, X. Chen, and X. Chen, "A novel postprocessing method for robust myoelectric pattern-recognition control through movement pattern transition detection," *IEEE Trans. Human-Mach. Syst.*, vol. 50, no. 1, pp. 32–41, Feb. 2020.

[45] D. Stegeman and H. Hermens, "Standards for surface electromyography: The European project surface EMG for non-invasive assessment of muscles (SENIAM)," Tech. Rep., 2007.

[46] C. J. D. Luca, L. D. Gilmore, M. Kuznetsov, and S. H. Roy, "Filtering the surface EMG signal: Movement artifact and baseline noise contamination," *J. Biomech.*, vol. 43, no. 8, pp. 1573–1579, May 2010.

[47] S. Micera, J. Carpaneto, and S. Raspopovic, "Control of hand prostheses using peripheral information," *IEEE Rev. Biomed. Eng.*, vol. 3, pp. 48–68, 2010.

[48] M. Halaki and K. Ginn, "Normalization of EMG signals: To normalize or not to normalize and what to normalize to," *Computational Intelligence in Electromyography Analysis: A Perspective on Current Applications and Future Challenges*. 2012, pp. 175–194.

[49] L. Bi and C. Guan, "A review on EMG-based motor intention prediction of continuous human upper limb motion for human-robot collaboration," *Biomed. Signal Process. Control*, vol. 51, pp. 113–127, May 2019.

[50] N. Jiang, I. Vujaklija, H. Rehbaum, B. Graimann, and D. Farina, "Is accurate mapping of EMG signals on kinematics needed for precise online myoelectric control?" *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 3, pp. 549–558, May 2014.

[51] T. Matsubara and J. Morimoto, "Bilinear modeling of EMG signals to extract user-independent features for multiuser myoelectric interface," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 8, pp. 2205–2213, Aug. 2013.

[52] B. Xue *et al.*, "Multiuser gesture recognition using sEMG signals via canonical correlation analysis and optimal transport," *Comput. Biol. Med.*, vol. 130, Mar. 2021, Art. no. 104188.

[53] R. N. Khushaba, "Correlation analysis of electromyogram signals for multiuser myoelectric interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 745–755, Jul. 2014.

[54] X. Sheng, B. Lv, W. Guo, and X. Zhu, "Common spatial-spectral analysis of EMG signals for multiday and multiuser myoelectric interface," *Biomed. Signal Process. Control*, vol. 53, Aug. 2019, Art. no. 101572.

[55] Y. Zhang, Y. Chen, H. Yu, X. Yang, and W. Lu, "Dual layer transfer learning for sEMG-based user-independent gesture recognition," *Pers. Ubiquitous Comput.*, pp. 1–12, Apr. 2020.

[56] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, "Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network," *Frontiers Neurosci.*, vol. 11, p. 379, Jul. 2017.

[57] U. Côté-Allard *et al.*, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Apr. 2019.

[58] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning with double encoding-layer autoencoder for transfer learning," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, pp. 1–17, Jan. 2018.

[59] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu, "Quantized CNN: A unified approach to accelerate and compress convolutional networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4730–4743, Oct. 2018.

[60] F. Quivira, T. Koike-Akino, Y. Wang, and D. Erdogmus, "Translating sEMG signals to continuous hand poses using recurrent neural networks," in *Proc. IEEE EMBS Int. Conf. Biomed. Health Informat. (BHI)*, Mar. 2018, pp. 166–169.