sensors

MDPI

Article

Unsupervised Domain Adaptation for Inter-Session Re-Calibration of Ultrasound-Based HMIs

Antonios Lykourinas [1,2], Xavier Rottenberg [2], Francky Catthoor [2] and Athanassios Skodras [1,*]

[1] Department of Electrical and Computer Engineering, University of Patras, 26504 Patras, Greece; alykourinas@ac.upatras.gr

[2] Imec, 3001 Leuven, Belgium; francky.catthoor@imec.be (F.C.); xavier.rottenberg@imec.be (X.R.)

[*] Correspondence: skodras@upatras.gr; Tel.: +30-2610-996-167

Abstract: Human–Machine Interfaces (HMIs) have gained popularity as they allow for an effortless and natural interaction between the user and the machine by processing information gathered from a single or multiple sensing modalities and transcribing user intentions to the desired actions. Their operability depends on frequent periodic re-calibration using newly acquired data due to their adaptation needs in dynamic environments, where test–time data continuously change in unforeseen ways, a cause that significantly contributes to their abandonment and remains unexplored by the Ultrasound-based (US-based) HMI community. In this work, we conduct a thorough investigation of Unsupervised Domain Adaptation (UDA) algorithms for the re-calibration of US-based HMIs during within-day sessions, which utilize unlabeled data for re-calibration. Our experimentation led us to the proposal of a CNN-based architecture for simultaneous wrist rotation angle and finger gesture prediction that achieves comparable performance with the state-of-the-art while featuring 87.92% less trainable parameters. According to our findings, DANN (a Domain-Adversarial training algorithm), with proper initialization, offers an average 24.99% classification accuracy performance enhancement when compared to no re-calibration setting. However, our results suggest that in cases where the experimental setup and the UDA configuration may differ, observed enhancements would be rather small or even unnoticeable.

Keywords: Human–Machine Interfaces; ultrasound; hand-gesture recognition

check for updates

Citation: Lykourinas, A.; Rottenberg, X.; Catthoor, F.; Skodras, A. Unsupervised Domain Adaptation for Inter-Session Re-Calibration of Ultrasound-Based HMIs. *Sensors* **2024**, *24*, 5043. https://doi.org/10.3390/s24155043

Received: 30 May 2024
Revised: 12 July 2024
Accepted: 30 July 2024
Published: 4 August 2024

1. Introduction

In recent years, Human–Machine Interfaces (HMI) have been introduced into numerous applications, including prosthesis control, robotic arm control, exoskeletons, smart wheelchair control, smart environment control, and exergaming [1]. The goal of an HMI is to transcribe user movements or movements intention into the desired action, thus allowing for an effortless and natural interaction between the user and the machine. Therefore, several types of sensing modalities, both of invasive and non-invasive natures, have been developed for monitoring user's specific activities, such as hand, eye, limb, and joint movements. These types of sensing modalities measure different kinds of physiological signals that can be classified into three main categories: bio-potentials, muscle mechanical motion, and body-motion signals [2]. In order to accomplish their goal, HMIs leverage a wide variety of algorithms, ranging from simple thresholding to complex machine learning algorithms, for processing these different sensing modalities.

In the medical field, HMIs are introduced in rehabilitation and assistive technologies. Rehabilitative technologies aim to restore impaired motor function in individuals with motor disabilities (within the limits of each individual's disability) in order to gradually enable re-participation in activities of daily living, whereas assistive technologies attempt to allow an individual with motor disabilities to perform motor functions that are beyond their motor capabilities. In medical HMIs, bio-potentials such as electroencephalogram (EEG) [3], electromyogram (EMG) [4,5], and electroocullogram (EOG) [6]—generated by electrical sources within the human body and thus reflecting the function of organs by means of electrical activity—have been extensively utilized. Furthermore, there is a ongoing interest in signals that monitor gross muscle motion, such as force myography (FMG) [7] and mechanomyography (MMG) [8], and muscle-tendon movement such as electrical impedance tomography (EIT) [9] and medical ultrasound [10]. Hybrid HMIs have also been developed [11–13], which exploit complementary information of different sensing modalities and thus allow for improved control but come at the expense of increasing the complexity of the HMI.

Regarding all biological signals, the most widely employed sensing modality is surface-electromyography (sEMG) signals, a bio-potential that directly measures the electrical activity generated during voluntary contraction, since they can be easily acquired non-invasively and provide an intuitive control strategy to reproduce the function of a biological limb [2]. On the other hand, Ultrasound-Based (US-based) HMIs remain vastly unexplored compared to their sEMG counterparts, despite ultrasound sensing techniques providing a non-invasive framework for monitoring deep tissues inside the human body with high temporal resolution and sub-millimeter precision. In the context of US-based HMIs, two sensing modalities are commonly employed, namely, B-mode ultrasound and A-mode ultrasound [10]. Both modalities utilize a device, a transducer made from piezoelectric crystals that is capable of both transmitting and receiving US waves [14]. In B-mode ultrasound, phased-array transducers are utilized in order to synthesize a 2D image of the human tissues through either the combination of simultaneous emission of acoustic beams and software beam-forming or by sweep time control of the acoustic beam; whereas, in A-mode ultrasound, the simplest US sensing technique—a single transducer element—scans a line through the human body and the received echoes are plotted as a function of depth. Currently, research interest has shifted towards A-mode ultrasound sensing as it has been demonstrated that using a set of sparsely selected scanlines instead of the full imaging array does not hinder the HMI's performance [15,16]. The benefits of utilizing a reduced number of scanlines, such as reduced computational complexity and power consumption as well as miniaturization of the instrumentation, have motivated the proposal of novel US acquisition systems [17] and recent advancements in the development of flexible fully printed transducers targeting medical applications [18].

The aforementioned characteristics of ultrasound as a sensing modality can intuitively explain the superiority of US-based HMIs for simultaneous proportional control compared to their sEMG counterparts, as it has been shown in numerous works [19,20]. More recent works include a semi-supervised framework, featuring a Sparse Gaussian Process model and principal component analysis for operating a prosthetic device with two degrees of freedom (hand grasp and wrist rotation) [21] and a novel portable US armband system with 24 channels with a multiple receiver approach, which allowed for simultaneous and proportional control of 3-DoF and 4-DoF in an online and offline setting, respectively [22]. Except from simultaneous and proportional control, a promising application of US-based HMIs is hand gesture recognition. In [23], the authors used both a novel multi-task deep learning framework and a multi-output Gaussian process for the simultaneous estimation of wrist rotation angle and recognition of finger gestures. In a more recent work, Liu et al. [24] proposed an algorithm based on a Long Short-Term Memory framework for the recognition of handwritten digits (dynamic gesture recognition) based on A-mode ultrasound sensing.

However, there are several drawbacks that hinder the application of the aforementioned HMIs in practical applications:

1. Collecting large datasets may by feasible in laboratory conditions but remains impractical for real-life scenarios, a strict requirement for conventional ML/DL algorithms known for their data-hungry nature [25].
2. It is both cost- and time-inefficient to collect representative datasets and, thus, most datasets are well-suited only within a pre-defined context. For example, the performance of a classifier significantly drops when tested on new arm positions [26,27].

3. It is assumed that dataset samples are i.i.d. sampled from the same distribution, but in real-life scenarios, the test–time distribution quickly diverges from the distribution with which the model was initially trained. Indicative causes are muscle fatigue [28] and sensor donning and doffing [29].
4. The musculoskeletal differences between individuals hinder the interoperability of the HMI leading to only subject-specific solutions [30].

In an effort to enhance the robustness of a US-Based HMI, the authors of [31] examined the potential of adaptive learning using A-mode ultrasound sensing on mitigating the performance deterioration induced by muscle fatigue. Thus, they compared the performance between conventional adaptive ML algorithms and an adaptive convolutional neural network, in which, in contrast to the former, both the feature extractor and the classifier part of the network had adaptability capabilities. In order to compare their performance, they instructed subjects to perform 15 gestures (their selection was inspired by the prominent NinaPro Database [32], extensively used as a benchmark by the sEMG-based HMI research community) once, which corresponds to one repetition, for a total of 16 repetitions. All 16 repetitions were performed without any rest to enforce muscle fatigue. The first three repetitions were used for training, the fourth repetition was used for testing, and each one of the remaining repetitions was used as a separate testing phase. During each testing phase, the predictions as well as the embeddings of the test samples were retained for obtaining both pseudo-labels and mean class embeddings used for updating the feature extractor and classifier of the network separately. By updating their network, they were able to achieve a significant 26.79% improvement in accuracy during the late stage of muscle fatigue. In their work, the classifier is updated in order to adapt to the test–time distribution of the data, which differs from the data distribution that the classifier was initially trained on.

This gap between training and test–time distributions is encountered in various domains and is often referred as concept drift in the literature [33,34]. Regarding US-Based HMIs, one can easily construct cases where the concept drift is expected to be substantially larger than the others, i.e., multiple days of the HMI being unused, subjects with large anatomical differences, and multiple donning and doffing of the sensors. All the aforementioned factors constitute a challenging environment for the development of robust US-based HMIs. Compared to the US-based HMI research community, the sEMG-based HMI research community has made significant efforts for mitigating the aforementioned optimality gap. For example, Cote Allard et al. proposed a transfer learning (TL) framework—inspired by progressive neural networks and multi-stream adaptive batch normalization—that could take advantage of multiple small datasets, allowing models to generalize well to new subjects by utilizing a single repetition from each gesture [35]. Unfortunately, TL techniques require the user to manually annotate the data acquired for re-calibrating the HMI. A more appealing scenario is to be able to re-calibrate an HMI by using newly acquired unlabeled data. This possibility is offered by Unsupervised Domain Adaptation (UDA) algorithms that take advantage of an initial labeled dataset in order to adapt a classifier to newly unlabeled data that are sampled from a similar but different distribution.

UDA algorithms have achieved remarkable results in computer vision tasks, and notable performance enhancements have also been demonstrated in inter-session re-calibration of sEMG-based HMIs by using Adaptive Batch Normalization (AdaBN) [36] and by the incorporation of domain-adversarial training in a self re-calibrating neural network [37]. Their aforementioned success and unique advantages make them a perfect candidate for exploration in different sensing modalities. In this work, we investigate the effectiveness of a wide variety of UDA algorithms in the re-calibration process of a US-based HMI. The application of UDA in a particular domain is not a straightforward process and requires experimentation for the optimization of UDA algorithms hyperparameters, design of discriminator networks used in domain-adversarial training algorithms, and even the network architectures themselves in order to fully leverage their capabilities. For our purpose, we used the Ultra-Pro dataset [23], which, to the best of our knowledge, is the only publicly available dataset. Compared to the induced muscle fatigue study [31], the Ultra-Pro dataset provides a challenging benchmark since all gestures are performed with concurrent wrist rotation and there are no intermediate data to bridge the gap between different sessions of the same subject. For assessing the performance of the UDA algorithms, we introduce an adaptation scheme, resembling the adaptation schemes introduced in the re-calibration of sEMG-based HMIs [37], where the newly acquired data arrive at two different time frames, in a shorter and a larger one from when the original labeled data were acquired, with the latter representing a more challenging scenario. Finally, we discovered that with the appropriate initialization, several UDA algorithms are capable of enhancing the performance of a US-based HMI compared to its non re-calibrated counterpart. However, in a realistic scenario, where the input modality, the data acquisition protocols, the re-calibration period, or the pattern recognition algorithm may differ, it is more likely that the performance enhancements would be rather small or not achievable at all.

The main contributions of the paper are the following:

1. A thorough examination of unsupervised domain adaptation algorithms for the re-calibration of US-based HMIs, where extensive guidelines for optimizing their performance in the field of US-based hand gesture recognition are provided. We examine domain-adversarial algorithms such as Domain Adversarial Training of Neural Networks (DANN) and its variant, Virtual Adversarial Domain Adaptation (VADA), where cluster assumption violation is incorporated in its optimization objective, as well as the non-conservative UDA algorithm, in which source domain performance is ignored, such as decision boundary refinement training with a teacher (DIRT-T), which uses VADA as initialization, and SHOT, which is also a source-data agnostic algorithm. To the best of our knowledge, this is the first time UDA algorithms have been applied to this field.
2. A new CNN-based architecture, featuring 87.92% less parameters from the state-of-the art model for US-based simultaneous estimation of wrist rotation angle and prediction of finger gestures, suitable for a UDA setting without any performance degradation on the primary task.
3. A benchmark for performance comparison of different architectures on the Ultra-Pro dataset, simulating a realistic scenario where the HMI will need to be operable shortly after user data are collected.
4. Insights about the performance of each UDA algorithm, with DANN (a domain-adversarial training algorithm) offering a 24.99% average performance enhancement and systematically improving the HMIs performance after re-calibration for all subjects and sessions. Unfortunately, we discovered that all UDA algorithms examined are unable to fully restore the HMI's operability, even though newly acquired data for re-calibration were obtained from different within-day sessions, rendering them inappropriate for re-calibration purposes.

The rest of the paper is organized as follows. In Section 2, we introduce our methods, which include the different domain adaptation algorithms examined as well as our performance comparison benchmark on the Ultra-Pro dataset and the proposed adaptation scheme for the re-calibration of US-based HMIs tailored for UDA algorithms. In Section 3, we introduce our proposed CNN-based architecture and explain why modifications were deemed necessary for its suitability in a UDA task. In Section 4, we describe our experimental setup, which includes a brief description of the Ultra-Pro dataset and training details about both single-task and multi-task settings. In Section 5, we provide our results regarding the re-calibration performance of the UDA algorithms as well as our performance comparison results of the proposed architecture with the state-of-the-art in both single- and multi-task settings. Finally, in Section 6, we conclude our work.

2. Methods

2.1. Essential Background

2.1.1. Domain Adaptation

Domain Adaptation (DA) refers to the problem of leveraging knowledge from a domain with abundance of data, referred to as source domain $\mathcal{D}_s$, to improve the performance of a model in a sample-scarce domain, referred to as target domain $\mathcal{D}_t$. Samples drawn from the target domain could be either labeled or unlabeled, which yields to a Supervised DA (SDA) problem or an Unsupervised DA (UDA) problem, respectively, with the latter being the hardest scenario and the subject of this paper. In a re-calibration setting, UDA algorithms can be utilized through treating the first session of each subject as their initial labeled dataset (source dataset $\mathcal{D}_S$) and the rest of their sessions as newly acquired unlabeled data (target dataset $\mathcal{D}_T$). The ability of UDA algorithms to perform re-calibration in an unsupervised manner makes them more appealing than TL techniques as they completely remove the burden of manual data annotation from the user. DA algorithms can be mainly categorized as divergence-based or domain adversarial training based. Regardless of their nature, all DA techniques rely on the theoretical background provided in the work of Ben-David et al. [38], which suggests that for effective domain adaptation to be achieved, predictions must be made based on features that are domain-indistinguishable.

2.1.2. Commonly Used Notation

Due to the fact that domain adaptation is strongly related to source and target domain feature distribution alignment, it is convenient to think of the classifier as a composite of an embedding function and an embedding classifier $h = g \circ f$. The embedding function $f_\theta : X \to \mathcal{Z}$ maps the input data distribution X to a feature space $\mathcal{Z}$, whereas the embedding classifier $g_\theta : \mathcal{Z} \to \mathcal{C}$ maps the embeddings to $\mathcal{C}$, where $\mathcal{C}$ denotes the $(K-1)$ simplex. Furthermore, it is common to denote with $\mathcal{D}_s$ the joint distribution over input x and one-hot label y, and with X_s the marginal input distribution. Both $\mathcal{D}_t$ and X_t can be analogously defined for the target domain.

2.2. Domain Adaptation Algorithms

2.2.1. Conservative DA

In Conservative DA, we assume that the optimal classifier is the one that achieves a low generalization error in both the source and target domains. Domain adversarial training algorithms are based exactly on this assumption. Initially proposed in [39], domain adversarial training of Neural Networks (NN) allows to train a neural network to jointly learn domain-indistinguishable features that also maximize NN's discriminability on the main task. This behavior is achieved by adding a domain classifier head to the feature extractor part of the network f_θ, for domain discrimination based on its embeddings, and a gradient reversal layer between the feature extractor and the domain head that reverses the gradients flowing through f_θ, which enforces feature distribution alignment in an adversarial manner. DANN was later extended to Virtual Adversarial Domain Adaptation (VADA) [40] through the incorporation of cluster violation assumption in a UDA setting. In DANN, the goal is to minimize the following objective:

$$\min_\theta \mathcal{L}_y(\theta; \mathcal{D}_s) + \lambda_d \mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t) \tag{1}$$

where $\mathcal{L}_y$ and $\mathcal{L}_d$ in Equation (1) represent the classification and the domain loss, respectively. DANN's inability to achieve robust and reliable adaptation under several conditions led to the incorporation of constraints that rely on the cluster violation assumption in its optimization objective [40]. Cluster violation assumption states that the input data distribution X contains clusters and that samples belonging to the same cluster also belong to the same class. Thus, the decision boundaries of the classifier should not cross data-dense regions. This behavior can be achieved via the incorporation of conditional entropy:

$$\mathcal{L}_c(\theta, \mathcal{D}_t) = -\mathbb{E}_{x \sim \mathcal{D}_t}[h_\theta(x) \log(h_\theta(x))] \tag{2}$$

which forces the classifier to be confident about its predictions and, thus, pushes the decision boundaries of the classifier further away from the target data. However, the approximation of Equation (2) breaks if the classifier is not locally-Lipschitz, which means that the classifier is allowed to abruptly change its predictions in the vicinity of the training samples. Thus, the authors decided to explicitly incorporate the locally-Lipschitz constraint via Virtual Adversarial Training (VAT) [41] by including the following additional term to the objective function:

$$\mathcal{L}_v(\theta, \mathcal{D}) = \mathbb{E}_{x \sim \mathcal{D}}\left[\max_{||r||<\epsilon} \mathbf{D_{KL}}(h_\theta(x)||h_\theta(r+\epsilon))\right] \tag{3}$$

which enforces the classifier to be consistent about its predictions inside the norm-ball neighborhood of ϵ. VAT loss can be seen as a form of regularization; thus, it can be applied in both the source and the target domains. The combination of domain adversarial training with the semi-supervised objectives introduced as additional constraints in order to incorporate the cluster violation assumption in a domain-adversarial training setting yields the following optimization objective:

$$\begin{aligned} \min_\theta \mathcal{L}_y(\theta; \mathcal{D}_s) + \lambda_d \mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t) + \lambda_s \mathcal{L}_v(\theta; \mathcal{D}_s) \\ + \lambda_t[\mathcal{L}_c(\theta; \mathcal{D}_t) + \mathcal{L}_v(\theta; \mathcal{D}_t)] \end{aligned} \tag{4}$$

commonly referred to as VADA. The terms λ_d, λ_s, and λ_t are hyperparameters and state-of-the-art performance has been achieved across a wide variety of tasks by setting their values to 10^{-2}, 1, and 10^{-2}, respectively.

2.2.2. Non-Conservative DA

In non-conservative DA, we assume that the optimal classifier in the target domain does not coincide with the classifier that achieves a low generalization error in both the source and the target domain. The authors of VADA [40] assume that the optimal classifier still violates the cluster assumption in the target domain and that this is what causes the aforementioned optimality gap. Their suggestion is to use VADA as initialization and further minimize the cluster assumption in the target domain. In order to make the optimization process parameterization invariant, the optimization is performed subject to the constraint of the Kullback–Leibler (KL) divergence between h_θ and $h_{\theta+\Delta\theta}$ being small for $x \sim \mathcal{D}_t$, which yields the corresponding Lagrangian:

$$\min_\theta \lambda_t[\mathcal{L}_t(\theta_n)] + \beta_t \mathbb{E}_{x \sim \mathcal{D}_t}\left[\mathbf{D_{KL}}(h_{\theta_{n-1}}(x)||h_{\theta_n}(x))\right] \tag{5}$$

and can be approximated by a finite number of stochastic gradient steps. The number of refinement steps is referred to as refinement interval $\mathcal{B}$. The term $\mathcal{L}_t$ is referred to as target-side cluster assumption violation loss, and it is simply the sum of conditional entropy loss and VAT loss in the target domain. The model $h_{\theta_{n-1}}$ is interpreted as the sub-optimal (teacher) model for the student model h_{θ_n}. This optimization problem is referred to as decision-boundary iterative refinement training with a teacher (DIRT-T) and, following the authors suggestions, we use ADAM with Polyak averaging for its optimization.

2.2.3. Source HypOthesis Transfer (SHOT)

A more recent work [42] suggests that robust adaptation can be achieved by using solely a model trained on the source data. They rely on the assumption that the original hypothesis learned from the source domain data is closely related to the optimal hypothesis in the target domain and, thus, should remain the same $g_s = g_t$. As a result, they propose Source HypOthesis Transfer, which attempts to learn the target-specific feature extractor module via minimizing the following objective:

$$\begin{aligned}\mathcal{L}(f_t; \mathrm{X}_t, \hat{y}_t) = \mathcal{L}_{ent}(f_t; \mathrm{X}_t) + \mathcal{L}_{div}(f_t; \mathrm{X}_t) \\ -\beta \mathbb{E}_{(x_t, \hat{y}_t) \sim X_t \times \hat{Y}_t} \sum_{k=1}^{K} \mathbb{I}_{[k=\hat{y}_t]} \log(h_\theta(x)[k])\end{aligned} \tag{6}$$

The first two terms combined form the Information Maximization (IM) Loss; the first term is the conditional entropy, whereas the second term is referred to as the fair-diversity objective $\mathcal{L}_{div}$, which enforces the classifier to predict all classes in the target domain with equal probability:

$$\mathcal{L}_{div}(h_\theta; \mathrm{X}_t) = \sum_{k=1}^{K} \hat{p}_k \log \hat{p}_k \tag{7}$$

$$= \mathbf{D_{KL}}(\hat{p}, \frac{1}{K}\mathbf{1}_K) - \log K \tag{8}$$

where $\hat{p} = \mathbb{E}_{x_t \in X_t}[h_\theta(x)]$ is the mean output embedding vector. The final term (Equation (6)) introduces a self-supervised objective, and stems from the observation that the IM Loss has the tendency to assign samples that result in low-confidence classifier predictions to the wrong clusters. At the beginning of each epoch, class prototype embeddings are computed and each sample is assigned with a pseudo-label based on the minimal distance of its embedding from the class prototypes. Class prototypes are used for generating pseudo-labels instead of the classifier predictions as they can be noisy due to domain shift. Prototype embeddings are computed from the classifier's prediction, but in the following epochs, they are updated using the pseudo-labels obtained in the previous epoch in order to obtain new pseudo-labels. In this work, pseudo-labels are updated for five epochs but, according to the authors of SHOT, sufficiently good results can be obtained by updating the pseudo-labels once.

2.3. Performance Comparison Benchmark

In order to allow for a fair performance comparison between our proposed architecture and the current state-of-the-art (as implemented by us), we propose a benchmark. The motivation behind providing results of our own implementation of the current state-of-the-art is due to the limited training details provided in the respective work, which hinders the application of a standardized benchmark. To begin with, in order to allow for a fair comparison, we performed the same train–validation–test split and also treated each session separately, following the original paper [23].

We also considered it crucial to train multiple models for each experiment, as conclusions drawn from a single trial can rather be pessimistic or optimistic. Thus, for each architecture and task in hand, 10 different models were trained for each session of the Ultra-Pro dataset using different random seeds. The seeds were sampled from a uniform distribution with values ranging from 0 to $2^{32} - 1$ and were used for initializing python libraries in order to make the data shuffling and weight initialization processes deterministic. The selected range captures all possible joint initialization values of the random number generations of the utilized libraries, and a uniform distribution was used for their sampling in order to avoid any bias. Since the seed number does not hold any physical meaning, the selection of a uniform distribution is the most suitable choice. Finally, by using optimal learning rates discovered for each architecture and task (see Appendix A), determined using the validation datasets of the Ultra-Pro dataset, we followed the procedure described below for comparing their task performance in each subject and session of the Ultra-Pro dataset:

1. Tasks: repeat for each task (three in total) in hand.
2. Architecture: repeat for each architecture.

3. Training details: Train 10 different models for each session of the Ultra-Pro dataset by drawing seeds from the uniform distribution described previously in this section. Each model is trained for 20 epochs using the following:
    (a) A batch size of 16;
    (b) ADAM as the optimizer;
    (c) The predetermined learning rates (see Appendix A for details).
4. Evaluation:
    - Each trained model is tested on the corresponding test set.
    - Results are reported using the mean of the performance metric (depending on the task) with standard deviation.

In this work, we are also interested in a performance comparison of the two architectures in a multi-task setting, where a single model simultaneously infers the finger gesture label as well as the wrist rotation angle. However, the proposed benchmark is not directly applicable to the multi-task setting. Thus, in order to apply our benchmark to the multi-task setting, we had to perform the following modifications to the overall procedure:

1. Architecture: Repeat for both architectures.
2. Multi-Task: Utilize both the class labels y_c as well as the ground truth regression values y_r to train the architecture using the improved automatic weighting function. For this part, a single neuron is connected in parallel to the classification head of the network (see Figure 1) for the prediction of the wrist rotation angles.
3. Training details: Train 10 different models by sampling random seeds from the uniform distribution described previously in this section. Each model is trained for twenty epochs using the following:
    (a) A batch size of 16;
    (b) ADAM as the optimizer;
    (c) The predetermined learning rates for the multi-task models (see Appendix A).
    (d) Evaluation:
        i. Each model is tested in the corresponding test set. It is important to note that in the multi-task setting, a model outputs both the gesture label and the wrist rotation angle during inference.
        ii. Results are reported using the mean of the performance metric with standard deviation.

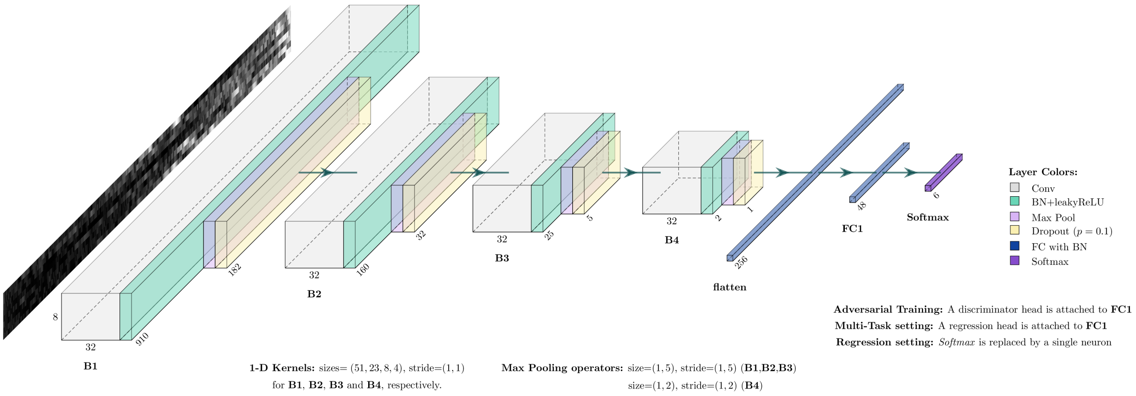
8
32
910
B1
182
32
160
B2
32
32
25
5
B3
32
2
1
B4
256
flatten
48
FC1
6
Softmax
Layer Colors:
Conv
BN+leakyReLU
Max Pool
Dropout ($p = 0.1$)
FC with BN
Softmax
Adversarial Training: A discriminator head is attached to FC1
Multi-Task setting: A regression head is attached to FC1
Regression setting: *Softmax* is replaced by a single neuron
1-D Kernels: sizes= (51, 23, 8, 4), stride=(1, 1)
for B1, B2, B3 and B4, respectively.
Max Pooling operators: size=(1, 5), stride=(1, 5) (B1,B2,B3)
size=(1, 2), stride=(1, 2) (B4)

Figure 1. Our proposed CNN architecture, consisting of 4 distinct blocks, a fully connected layer, and a Softmax layer. Each block consists of a convolutional layer, followed by a pre-activation batch normalization layer; a max pooling layer; and, finally, a regular dropout layer with a forget rate of $p = 0.1$. For each setting, modifications were made only after the last fully connected layer of the network. Finally, leaky ReLU with a negative slope of 0.1 was used as the activation function.

This standardized benchmark allows us to effectively compare the performance of different architectures on the Ultra-Pro dataset without making use of the validation sets and, thus, simulate a more realistic scenario while also avoiding any bias. However, it is important to note that the validation sets were utilized for determining the optimal learning rates for each architecture and task in hand. Furthermore, we also provide the optimal learning rates for the current state-of-the-art model, which are not provided in the respective paper [23]. Regarding the performance metrics of each task, we used *classification accuracy* for evaluating the model's performance on the task of finger gesture recognition and the *coefficient of determination* R^2 for evaluating the model's performance on the task of wrist rotation angle estimation. The selection of the aforementioned metrics was based on their use in relevant existing works [21–23].

2.4. Adaptation Scheme

As mentioned previously in Section 4.1, the Ultra-Pro dataset features three separate sessions for each subject. Between consecutive sessions, the subjects were allowed to rest for a reasonable amount of time, although not pre-determined as examining the robustness of the HMI across multiple within-day sessions was not on the dataset's creators' initial intentions, in order to prevent muscle fatigue. In this work, in order to assess the performance of UDA algorithms for the re-calibration of US-based HMIs, we introduce the following adaptation scheme. First and foremost, we treat the first session of each subject (only training data with labels) as its source dataset $\mathcal{D}_{S1} = \mathcal{D}_s$ and then treat the rest of the subject's sessions (only training data) as our newly acquired unlabeled data used to re-calibrate the US-based HMI. This scheme allows us to create two adaptation scenarios for the re-calibration of the HMI. In our first scenario, we attempt to re-calibrate the HMI using data solely from the second session, whereas in our second scenario we attempt to re-calibrate the HMI by using data solely from the third session. The latter represents a more challenging scenario, since the time interval between the acquisition of the first and the third session is larger and will consequently result in a larger domain shift.

This particular adaptation scheme will allow us to understand to what extent the UDA algorithms could improve the performance of the HMI based on the time interval between the moment the original labeled dataset was collected and the moment the acquisition of newly unlabeled data for re-calibration occurred. Due to the within-day data acquisition of all sessions, we will not be able to provide insights about the effect of UDA on the re-calibration of an HMI for long-term enhancements, as previously demonstrated in sEMG-based research [37]. However, the proposed adaptation scheme is sufficient for obtaining insights about their performance enhancement capabilities and understand if more research effort would be worthwhile. Finally, in order to establish a baseline performance, i.e., when no unlabeled data are available for re-calibration, we used the trained CNN-based modules (with the details provided in Section 4.3) of each subject's first session, i.e., our source dataset, and evaluated their performance on the rest of each subject's sessions. Thus, the results of the non re-calibrated HMIs on sessions 2 and 3 will be referred to as non re-calibration results.

3. Proposed CNN Architecture

In this work, a new CNN architecture is proposed for the task of simultaneous wrist rotation angle estimation and finger gesture prediction as well as for each task individually. The CNN architecture for the classification task is illustrated in Figure 1. The architecture consists of four distinct blocks, a fully connected layer (acting as a bottleneck), and a *Softmax* layer. Furthermore, each block B consists of a convolutional layer, followed by a pre-activation batch normalization layer [43], a max pooling layer, and a dropout layer with a forget rate of $p = 0.1$ [44]. It is important to mention that pre-activation batch normalization is also included in the fully connected layer of the network. Finally, we used *Leaky ReLU* with a negative slope of 0.1 as the activation function [45].

Following the work of [23], we also apply 1-D kernels, operating along the width dimension, with a stride equal to the size of the kernel in the width dimension and a stride of 1 in the height dimension. The sizes of the kernels for the convolutional layers of blocks B1, B2, B3, and B4 are 51, 23, 8, and 4, respectively. Similarly to all other DNN architectures, the trainable parameters of our network are learned in the training process by updating them using gradient-based methods and backpropagation of error algorithm. The error is associated with a loss function, which is estimated multiple times during training from a reduced subset of the initial dataset, referred to as the batch, instead of the whole dataset in order to accelerate the training process. Finally, the addition of batch normalization layers, whose transformations rely on batch statistics, allows faster convergence of our network to desirable weights for a wider range of learning rates.

The motivation behind the proposal of a new architecture is twofold; first and foremost, we believe that by adopting design trends from models utilized in a UDA setting we will encourage the formation of class-discriminative clusters and thus facilitate source- and target-distribution alignment [42]. The need of class-discriminative features is highlighted by the fact that discrepancy between the two domains can be minimized by simply mixing samples from different classes, which will inevitably lead to degradation of the classification performance [46]. The trends that we adopted in our network were the use of *Leaky ReLU* as well as dropout layers in the feature extractor part of the network, which are commonly used in network architectures regarding UDA in various fields [37,40]. The proposed architecture's ability to learn class-discriminative features is illustrated in Figure 2. Lastly, we believe that by enhancing the model's discriminative ability on the main task, i.e., the classification of finger gestures, we will also be able to achieve better adaptation results.

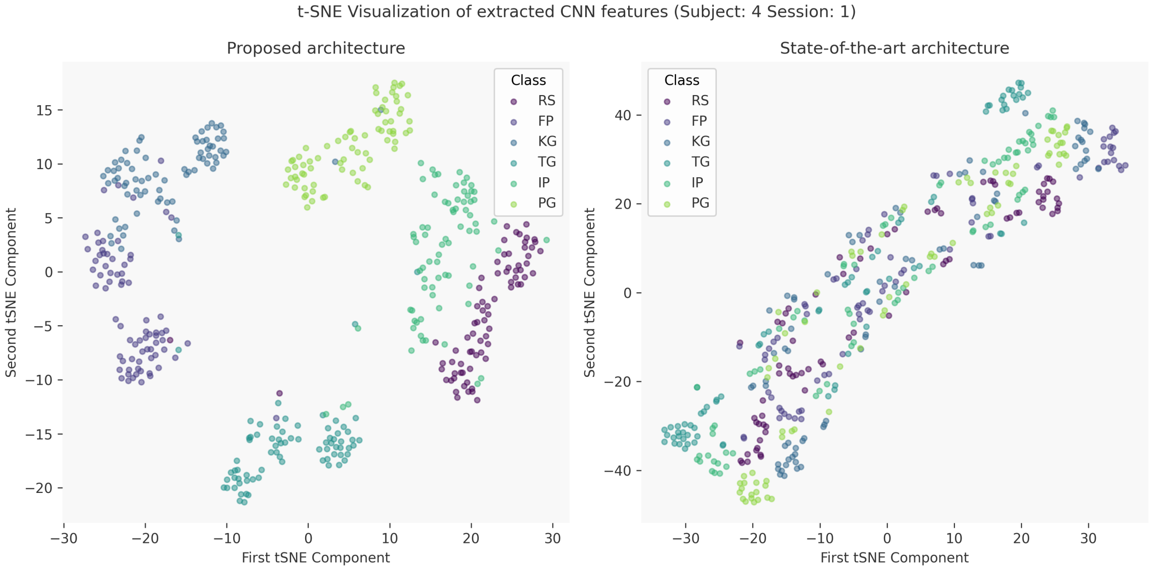

Figure 2. t-SNE visualization of the features learned from our network and the current state-of-the-art. Our network adopts class-discriminative features, which help alleviate the domain shift in a UDA setting, in contrast with the state-of-the-art model, which projects the samples to an 8192-dimensional feature space and adopts more generic features. Thus, the state-of-the-art model is more prone to mixing different class labels when learning domain-indistinguishable features in a UDA setting.

3.1. Improving the Original CNN Model

Our experimentation with the original CNN model [23] allowed us to identify two main setbacks on its design. First and foremost, the original network is characterized by a small receptive field, which influences the quality of the extracted spatial features, and secondly, the connections from the extracted spatial features to the first fully connected layer contribute most of the network's learnable parameters. The dimensionality of the original feature space is equal to 8192 (32 filters × 8 height × 32 width), and the first fully connected layer—in which neurons are trained to distinguish patterns of the extracted spatial features—features 48 neurons. Each neuron in the fully connected layer is connected to each individual feature and their importance is expressed through a corresponding weight. Additionally, each neuron in the first fully connected layer also includes a bias term. This results in a total of 393.264 learnable parameters (8192 features × 48 neurons plus the corresponding 48 bias terms).

Based on our observations, we decided to experiment with the depth of the network. This design choice was based on the fact that it is well known that deeper convolutional neural network architectures are capable of capturing rich and more complex features of the data [47]. Increasing the depth of the network was also deemed beneficial due to the high resolution of our input data in the width dimension. Furthermore, by increasing the depth of the network, we also managed to address the second setback we identified, as the total number of learnable parameters in the first fully connected layer is significantly reduced since the dimensionality of the proposed model's feature space (a 256-dimensional feature space compared to the 8192-dimensional feature space of the original model) is significantly reduced. By increasing the depth of the network and experimenting with different sizes of kernels and max pooling operators, we determined the appropriate depth and layer parameters for our network by training all the different models on the sessions of the Ultra-Pro dataset and evaluating their performance on the corresponding validation sets. Finally, the derivation of the proposed model was finalized by the incorporation of the aforementioned design techniques commonly employed in a UDA setting.

3.2. Computational and Time Complexity Analysis

The growing interest of DNN-based solutions is mainly due to two reasons: (a) advancements made in hardware, which allow us to train and deploy, and (b) the availability of data. However, recent research efforts have focused on deriving optimal architectures under user-defined constraints and, furthermore, deploying them in resource-constrained devices such as mobile phones, drones, and robots [48–50]. Thus, we provide results for commonly used metrics such as the number of floating point operations, multiply–accumulate operations, and direct memory accesses to evaluate its computational efficiency as well as the corresponding memory requirements for deployment (see Table 1). Furthermore, we also provide the same results for the original model.

Table 1. Computational efficiency and memory resources requirements for the proposed architecture.

Model	Parameters	MFLOPs	MMACs	MDMACs	Inference Time (ms)	Inference Time CPU (ms)	Memory (MB)
Proposed Model	401,382	87.77	43.97	44.51	0.58	1.21	0.193
Original model	50,582	85.34	42.64	43.62	0.25	0.191	1.598

According to [48], our proposed CNN architecture is suitable for deployment in resource-constrained devices since the number of floating point operations required for a forward pass lies within the range of 10-150 MFLOPs. Furthermore, the inference time of 0.58 ms on our GPU and 1.21 ms on our CPU is suitable for the targeted applications since latency for prosthetic control is suggested to be within the range of 100 to 250 ms [51]. With our modifications, we are able to significantly reduce the model's size, though at the cost of increasing the number of floating-point operations, multiply–accumulate operations, and direct memory accesses.

3.3. Real-World Applications of the Proposed CNN Architecture

Based on our computational complexity and memory resources analysis, our proposed architecture is rendered appropriate for applications targeting recourse-constrained devices. Thus, the possibility of potential integration to a prosthetic socket or a wearable armband is offered. The proposed CNN architecture could be integrated in an assistive technology, by assisting amputees through transcribing their indented motions to commands and control signals for operating a prosthetic arm, and in a rehabilitative technology, by enabling people with motor disabilities (e.g., neuromuscular disorders) to gradually restore motor functions within the limits of their disabilities through interactive means such as exergaming.

4. Experimental Setup

4.1. Ultra-Pro Dataset

The ultrasound-based adaptive prosthetic control (Ultra-Pro) dataset [23], is a publicly-available dataset that features US RF data as well as sEMG and Inertial Measurement Unit sensor data from the upper limb of four subjects (all transradial amputees), targeting the task of simultaneous estimation of the wrist rotation angle and recognition of finger gestures. The US RF data were captured using a customized armband, featuring eight evenly spaced transducers operating at 5 MHz, driven by a customized ultrasound system [17]. Each subject participated in three separate sessions. In each session, the subjects were instructed to perform six finger gestures—namely, Rest (RS), Fine Pinch (FP), Key Grip (KG), Tripod Grid (TG), Index Point (IP), and Power Grip (PG)—and each gesture was performed for 50 s with concurrent wrist rotation at a frequency of approximately 0.5 Hz. During the execution of the instructed movements, the data streams generated from all sensors were synchronously captured using custom specialized software. For a detailed description of the dataset, we encourage readers to refer to the original paper [23].

4.2. Data Prepossessing

For preprocessing the RF US signals, we followed the standard procedure of time–gain compensation, bandpass filtering, envelope detection, and log compression, adopting the procedure and the hyperparameters provided in [52] with the exception of replacing the commonly used Gaussian filter with a 101-tap bandpass FIR filter designed using the window method (hamming window). The filter is characterized by a low and a high cutoff frequency of 2 and 8 MHz, respectively. Following the work of [23], we used the processed ultrasound signals as the input modality to all networks.

4.3. Training Details

For hand-motion recognition, i.e., a classification task, we compared the performance of categorical cross-entropy loss and its variant with label smoothing [53]. For wrist rotation angle estimation—i.e., a regression task—mean squared error loss was used. Furthermore, we acknowledge the tedious process of determining optimal weights for each task in-hand in a multi-task learning setting and, thus, we automatically update their weights by using the loss function proposed in [54] for simultaneous hand motion recognition and wrist rotation angle prediction. The selected loss function also considers the homoscedastic uncertainty of each task [55], but the regularization term of each task τ is modified to $R_\tau = \ln(1 + \sigma_\tau^2)$. The updated term enforces positive regularization values and, as a consequence, prevents loss from becoming negative during training.

5. Results

We considered it crucial to determine a fixed set of hyperparameters that can ensure near-optimal performance across subjects and sessions in order to simulate a realistic setting, where the HMI will need to be operable shortly after user data are collected. The optimal hyperparameters were determined using Optuna [56] for each combination of architectures (both ours and current state-of-the-art architectures [23]) and tasks in hand (see Appendix A).

5.1. Results on Single-Task Models

In this set of experiments, we compared the performance between our proposed architecture and the current state-of-the-art (our implementation) across all subjects and sessions of the Ultra-Pro dataset on the tasks of wrist rotation angle estimation and finger gesture recognition individually. For the performance comparison, we used the benchmark we proposed in Section 2.3. The results for finger gesture recognition (using cross entropy and its variant with label smoothing as the loss function) and wrist rotation angle estimation can be seen in Figures 3, 4, and 5, respectively.

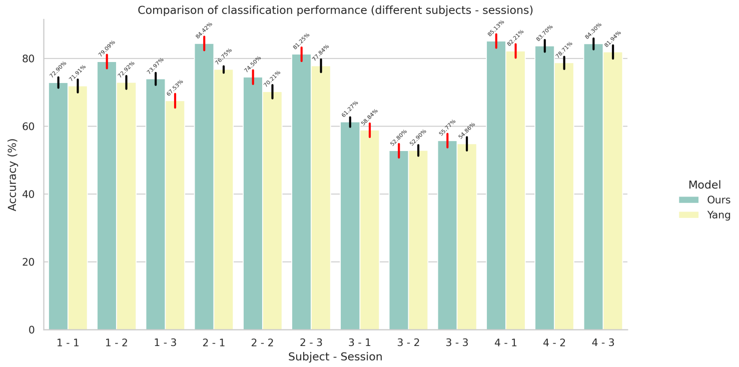

Figure 3. Comparison of finger gesture recognition performance in terms of accuracy between the proposed CNN-based model and the model proposed by Yang et al. [23] in a single-task setting. The red colored error bars represent standard deviations that exceed 2%.

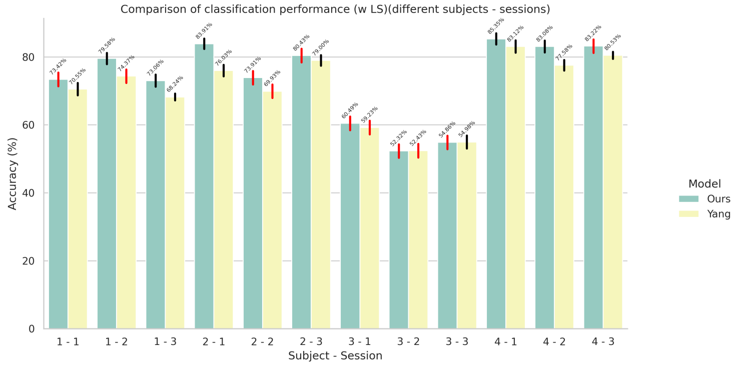

Figure 4. Comparison of finger gesture recognition performance (with label smoothing) in terms of accuracy between the proposed CNN-based model and the model proposed by Yang et al. [23] in a single-task setting. The red colored error bars represent standard deviations that exceed 2%.

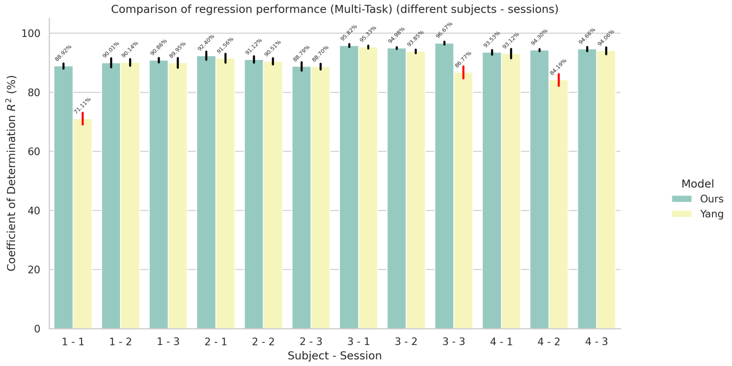

Figure 5. Comparison of wrist rotation angle estimation performance in terms of coefficient of determination R^2 between the proposed CNN-based model the model proposed by Yang et al. [23] in a single-task setting. The red colored error bars represent standard deviations that exceed 2%.

According to our results, it is evident that our modifications to the original network, which were deemed necessary in order to improve its performance in the UDA setting, do not hinder its performance in either of the two tasks. In some cases—for example, the first session of the second subject—we notice a 7.88% overall improvement in the task of finger gesture recognition. However, we have noticed cases where the performance of our model is inferior, for example, the second session of the third subject.

5.2. Results on Multi-Task Models

In this set of experiments, we evaluated the performance of our proposed architecture for the task of simultaneous wrist rotation angle prediction and finger gesture classification. The results for simultaneous finger gesture recognition and wrist rotation angle estimation for the multi-task variants can be seen in Figures 6 and 7, respectively.

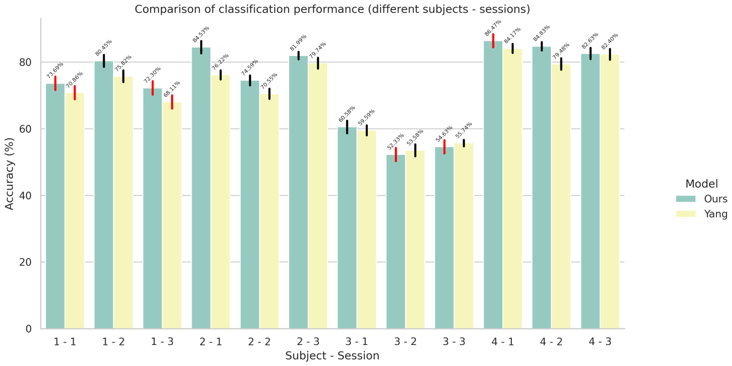

Figure 6. Comparison of finger gesture recognition performance in terms of accuracy between the proposed network and the network proposed by Yang et al. [23] in a multi-task setting using the improved automatic weighting loss function. The red colored error bars represent standard deviations that exceed 2%.

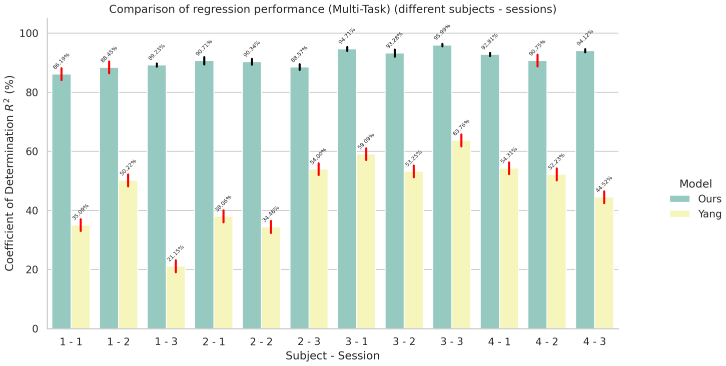

Figure 7. Comparison of wrist rotation angle performance in terms of coefficient of determination R^2 between the proposed network and the network proposed by Yang et al. [23] in a multi-task setting using the improved automatic weighting loss function. The red colored error bars represent standard deviations that exceed 2%.

As we can see in Figure 7, there is a substantial difference between the performance of our model compared to the performance of the original model on the wrist rotation angle estimation when using the improved loss function for automatic weighting of each task. These results indicate the need for an additional hyperparameter γ, introduced in the original paper [23], included in the automatic weighting loss function for favoring the regression task (wrist rotation angle estimation) over the classification task. The need for a hyperparameter violates the whole purpose of using an automatically weighed loss function. Instead, our model is able to achieve a better balance between the two tasks without the need of any additional hyperparameter.

5.3. Results on UDA

In the last set of our experiments, we evaluated the performance of all different domain adaptation algorithms on the two adaptation scenarios, introduced in Section 2.4. We have established our baselines for the non re-calibration cases, i.e., where no newly acquired data are available for re-calibration of the HMI, by inferring the best first-session models from our single-task experiments to the test dataset of the rest of each subject's sessions. For the experimentation phase, we used the validation sets of the corresponding adaptation scenarios and monitored each individual loss term as well as the validation accuracy in order to evaluate different configurations and hyperparameters for the UDA algorithms. This was deemed crucial due to the UDA algorithm's sensitivity to initialization. During our experimentation with domain-adversarial-based algorithms, i.e., DANN and VADA, we derived three important conclusions:

1. We found that the default hyperparameters, suggested by the corresponding authors, were the optimal hyperparameters for our case (See Appendix A). Furthermore, after several experiments with mini-batch gradient descent and its variants, we discovered that the recommended optimization algorithms with our pre-determined learning rates (for the single-task models) yielded the best results.
2. The process of jointly learning domain-indistinguishable features from both source and target domains was benefited by the use of complex discriminator networks. Through experimentation with different domain discriminator networks, we discovered that the optimal discriminator network for re-calibration of a US-based HMI featured three hidden layers, with 32, 24, and 16 neurons each.
3. We found that considering domain adversarial training with what is traditionally known as the feature extractor part of the network in CNN-based architectures, i.e., the network's layers before the fully connected layers, leads to sub-optimal results and it is beneficial to perform domain adversarial training in order to enforce feature distribution alignment in the first fully connected layer of our network (see Figure 1). The selection of the first fully connected layer as the feature space yielded a 48-dimensional space. The consideration of a lower-dimensional feature space can also be observed in the experimental setup of SHOT [42].

For our SHOT experiments, we selected the first-session best models trained using label smoothing from our single-task experiments for initialization, as recommended by the authors of SHOT [42]. Furthermore, we discovered that the optimal learning rates used for learning the target-specific feature extractor module coincide with the learning rates used to train the initialization modules. Moreover, we followed the authors' suggestions and adopted mini-batch gradient descent with a Nesterov momentum of 0.9 and a weight decay of 10^{-3} for updating the weights using backpropagation. At this point, it is important to note that due to our concerns about the optimization objective of SHOT, we considered two different cases. In the first case, we treat the SHOT optimization objective as the sum of information maximization loss and a self-supervised objective, whereas in the second case, we treat each term of the SHOT's objective function independently, since they can represent self-sufficient objectives (for hyperparameters and details, see Appendix A).

In our last set of experiments, we decided to include the normalized first principal components of the source and target's domains samples as supervisory signals in the optimization objective of DANN. The selection of the normalized principal component as a supervisory signal was based on the fact that it is linearly related to the wrist rotation angle and can also be obtained easily in an unsupervised manner [57]. Our motivation was to investigate if the performance of DANN could be augmented by the incorporation of two auxiliary tasks, i.e., estimating the wrist rotation angle in both the source and the target domains. The incorporation of the two auxiliary tasks in DANN yields the following optimization objective:

$$\min_{\theta} \mathcal{L}_y(\theta; \mathcal{D}_s) + \lambda_r \left[\mathcal{L}_{r-source}(\theta; \mathcal{D}_s) + \mathcal{L}_{r-target}(\theta; \mathcal{D}_t) \right] + \lambda_d \mathcal{L}_d(\theta; \mathcal{D}_s, \mathcal{D}_t) \tag{9}$$

where λ_r is a hyperparameter for controlling the influence of the two regression objectives during domain-adversarial training, which was set to 0.35. We also performed experiments using the ground truth regression labels; in this case, the hyperparameter was set to 0.45. We will refer to the results of DANN with self-supervision and ground truth wrist rotation angles as DANN SS and DANN GT, respectively. Finally, since the normalized principal component is an approximation of the true wrist rotation angle, we used mean absolute error instead of mean squared error in order to mitigate the influence of possible outliers.

Finally, model selection for each UDA algorithm was performed by selecting the weights that minimized its corresponding training loss. The accuracy results of the different UDA algorithms are presented in Table 2.

Table 2. Re-calibration accuracy results of Unsupervised Domain Adaptation algorithms.

	1-1	**1-2**	**1-3**	**2-1**	**2-2**	**2-3**	**3-1**	**3-2**	**3-3**	**4-1**	**4-2**	**4-3**
No Re-calibration	0.740	0.238	0.264	0.879	0.291	0.202	0.684	0.291	0.349	0.482	0.482	0.336
DANN	-	0.287	0.402	-	0.395	**0.247**	-	0.316	0.396	-	0.580	0.428
DANN (w LS)	-	0.293	0.322	-	0.258	0.231	-	0.342	0.415	-	0.588	0.397
VADA	-	**0.313**	0.380	-	0.224	0.182	-	0.316	0.371	-	0.542	0.398
VADA (w LS)	-	0.266	0.376	-	0.356	0.235	-	0.337	0.415	-	0.611	0.408
DIRT-T	-	0.282	**0.428**	-	0.340	0.235	-	0.317	0.396	-	**0.644**	0.435
SHOT (original)	-	0.200	0.295	-	0.242	$0,178$	-	0.302	0.297	-	0.529	0.378
SHOT (ours)	-	0.213	0.311	-	0.240	0.180	-	0.307	0.303	-	0.531	0.382
DANN-GT	-	0.262	0.367	-	0.322	0.200	-	**0.362**	0.438	-	0.571	0.371
DANN-SS	-	0.256	0.287	-	**0.407**	0.238	-	0.341	**0.442**	-	0.351	**0.442**

Bold entries indicate the best re-calibration results for each adaptation scenario.

From the obtained results, we drew the following conclusions:

1. The algorithms that rely on domain adversarial training, i.e., DANN and VADA, perform better than SHOT, which is a source-data agnostic UDA algorithm. This means that in the case where the domain shift is large, it is beneficial to jointly learn domain-indistinguishable features and maximize classifier's discriminability on the primary task.
2. It is beneficial to use DIRT-T to further minimize target-side cluster assumption in cases where the domain shift is small—for example, both adaptation scenarios of subject 4—but does not always guarantee better results.
3. Compared to all the UDA algorithms, DANN offers the greatest performance enhancements with an average improvement of 24.99%. Unfortunately, re-calibration using SHOT can even have a negative impact on the performance of the HMI.
4. Incorporation of the wrist rotation angle as a supervisory signal in the DANN objective, either by using the ground truth values or by obtaining them in an unsupervised manner by utilizing the normalized first principal component, does not lead to better results than re-calibrating the HMI by solely using DANN.

6. Conclusions and Future Work

Inspired by sEMG-based HMI research, we investigated the possibility of re-calibrating US-based HMIs using unlabeled data from different within-day sessions by employing state-of-the-art UDA algorithms. This is a more challenging task than making a classifier adapt to muscle fatigue [31], where no continuous stream of data is available for continual adaptation, in a more demanding dataset featuring concurrent wrist rotation. For our experiments, we introduced an adaptation scheme on the publicly available Ultra-Pro dataset, which allows us to investigate the performance enhancements of UDA algorithms through two adaptation scenarios based on the time interval between the initial label dataset that was collected and the acquisition of unlabeled data used for re-calibration. Our experimentation also led to the proposal of a new architecture, featuring 87.92% less parameters than the current state-of-the-art. For effective performance comparison of the two architectures, we introduced a benchmark that simulates a realistic scenario while also avoiding any bias, where no validation set will be available and the HMI will need to be operable shortly after user data are collected. Furthermore, we provide extensive guidelines for the re-calibration of US-based HMIs using UDA algorithms and draw important conclusions on their drawbacks and performance. According to our findings, our proposed CNN-based architecture achieves similar performance with the current state-of-the-art [23] while featuring only 50,582 trainable parameters instead of 401,382. Also, by using DANN, a domain-adversarial based algorithm, we achieved a 24.99% average performance enhancement and systematically improved the classification performance of a US-based HMI compared to its non re-calibrated counterpart for all subjects and sessions. However, our results indicate that in cases where the data acquisition process, the re-calibration period, and the network architecture may differ and proper initialization of the UDA algorithms may not be feasible, the observed enhancements would be rather small or even not noticeable.

We believe that the findings of our work raise several research questions that need to be addressed. To begin with, UDA algorithms require careful initialization in order to fully leverage their capabilities and cannot fully restore the operability of a US-based HMI. These results indicate the need for a online learning learning algorithm, capable of adapting in dynamic environments where test–time data continuously change in unforeseen ways. Furthermore, it is important to ensure that important information will not be lost during the dynamic adaptation of the model, a common issue that is often referred to as catastrophic forgetting [58]. To continue with, most of the aforementioned techniques rely on backpropagation of error algorithm to update the network's parameters, a non-local and computationally intensive learning rule. Based on our observations, our main research focus would be the development of an online continual learning algorithm, capable of adapting to dynamic environments using local learning rules. Finally, we plan to collect data in order to construct a representative dataset, featuring hand movements performed in activities of daily living and multiple sessions for a better evaluation of the re-calibration performance of different algorithms.

Author Contributions: Conceptualization, A.L., F.C., X.R. and A.S.; methodology, F.C., A.L. and A.S.; software, A.L.; validation, A.L.; formal analysis F.C., A.S. and A.L.; investigation, A.L., F.C. and A.S.; resources, F.C. and A.S.; writing—original draft preparation, A.L.; writing—review and editing, F.C., A.S. and X.R.; visualization, A.L.; supervision, F.C., X.R. and A.S.; project administration, X.R., F.C. and A.S.; funding acquisition, F.C., X.R. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by IMEC (Belgium) under the contract ORD-372666-C3B9V.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Ultra-Pro dataset used in this article is available online: https://doi.org/10.6084/m9.figshare.20448489.v1.

Conflicts of Interest: Authors X. R. and F. C. are employed by IMEC. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HMI	Human–Machine Interface
UDA	Unsupervised Domain Adaptation
US	Ultrasound
CNN	Convolutional Neural Network
DANN	Domain-adversarial Training of Neural Networks
EEG	Electroencephalogram
EMG	Electromyogram
EOG	Electrooculogram
FMG	Force Myography
MMG	Mechanomyography
EIT	Electrical Impedance Tomography
sEMG	Surface-electromyography
ML	Machine Learning
DL	Deep Learning
NinaPro	Non-Invasive Adaptive Hand Prosthetics
TL	Transfer Learning
AdaBN	Adaptive Batch Normalization
Ultra-Pro	Ultrasound-based Adaptive Prosthetic Control
DA	Domain Adaptation
SDA	Supervised Domain Adaptation
VADA	Virtual Adversarial Domain Adaptation
ADAM	Adaptive Moment Estimation
VAT	Virtual Adversarial Training
DIRT-T	Decision Boundary Iterative Refinement Training with a Teacher
SHOT	Source Hypothesis Transfer
SGD	Stochastic Gradient Descent

Appendix A. Hyperparameters

Table A1. Optimal learning rates for single-task models.

Task	Finger Gesture Recognition w/o LS	Figure Gesture Recognition w LS	Wrist Rotation Angle Estimation
Proposed model	0.0026	0.0035	0.0008
Original model	0.00011	0.00018	0.000079

Table A2. Optimal learning rates for multi-task models.

	Learning Rate
Proposed model	0.0014
Original model	0.00012

Table A3. Hyperparameters for DANN–VADA configurations.

UDA Algorithm	Domain Weight λ_d	Regression Weight λ_r	VAT Source λ_s	Target-Side CA Violation λ_t
DANN	0.1	-	-	-
DANN SS	0.1	0.36	-	-
DANN GT	0.1	0.45	-	-
VADA	0.1	-	1	10^{-2}

Table A4. Hyperparameters for SHOT.

UDA Algorithm	Information Maximization Loss	Self-Supervised Objective (β)
SHOT	1	0.1

Table A5. Hyperparameters for SHOT (our variant).

UDA Algorithm	Conditional Entropy	Fair Diversity	Self-Supervised Objective (β)
SHOT	1.0	0.7	0.1

References

1. Mat Rosly, M.; Mat Rosly, H.; Davis OAM, G.M.; Husain, R.; Hasnan, N. Exergaming for individuals with neurological disability: a systematic review. *Disabil. Rehabil.* **2017**, *39*, 727–735. [CrossRef] [PubMed]
2. Esposito, D.; Centracchio, J.; Andreozzi, E.; Gargiulo, G.D.; Naik, G.R.; Bifulco, P. Biosignal-Based Human–Machine Interfaces for Assistance and Rehabilitation: A Survey. *Sensors* **2021**, *21*, 6863. [CrossRef] [PubMed]
3. Gao, H.; Luo, L.; Pi, M.; Li, Z.; Li, Q.; Zhao, K.; Huang, J. EEG-Based Volitional Control of Prosthetic Legs for Walking in Different Terrains. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 530–540. [CrossRef]
4. Bao, T.; Xie, S.Q.; Yang, P.; Zhou, P.; Zhang, Z.Q. Toward Robust, Adaptiveand Reliable Upper-Limb Motion Estimation Using Machine Learning and Deep Learning–A Survey in Myoelectric Control. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 3822–3835. [CrossRef] [PubMed]
5. Atzori, M.; Cognolato, M.; Müller, H. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Front. Neurorobotics* **2016**, *10*, 9. [CrossRef] [PubMed]
6. Perez Reynoso, F.D.; Niño Suarez, P.A.; Aviles Sanchez, O.F.; Calva Yañez, M.B.; Vega Alvarado, E.; Portilla Flores, E.A. A Custom EOG-Based HMI Using Neural Network Modeling to Real-Time for the Trajectory Tracking of a Manipulator Robot. *Front. Neurorobotics* **2020**, *14*, 578834. [CrossRef] [PubMed]
7. Xiao, Z.G.; Menon, C. A Review of Force Myography Research and Development. *Sensors* **2019**, *19*, 4557. [CrossRef] [PubMed]
8. Beck, T.W.; Housh, T.J.; Cramer, J.T.; Weir, J.P.; Johnson, G.O.; Coburn, J.W.; Malek, M.H.; Mielke, M. Mechanomyographic amplitude and frequency responses during dynamic muscle actions: a comprehensive review. *BioMed. Eng. OnLine* **2005**, *4*, 67. [CrossRef] [PubMed]
9. Wu, Y.; Jiang, D.; Liu, X.; Bayford, R.; Demosthenous, A. A Human–Machine Interface Using Electrical Impedance Tomography for Hand Prosthesis Control. *IEEE Trans. Biomed. Circuits Syst.* **2018**, *12*, 1322–1333. [CrossRef]
10. Yang, X.; Castellini, C.; Farina, D.; Liu, H. Ultrasound as a Neurorobotic Interface: A Review. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 3534–3546. [CrossRef]
11. Kyranou, I.; Krasoulis, A.; Erden, M.S.; Nazarpour, K.; Vijayakumar, S. Real-time classification of multi-modal sensory data for prosthetic hand control. In Proceedings of the 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), Singapore, 26–29 June 2016; pp. 536–541. [CrossRef]
12. Li, X.; Samuel, O.W.; Zhang, X.; Wang, H.; Fang, P.; Li, G. A motion-classification strategy based on sEMG-EEG signal combination for upper-limb amputees. *J. Neuroeng. Rehabil.* **2017**, *14*, 2. [CrossRef] [PubMed]
13. Wei, S.; Zhang, Y.; Liu, H. A Multimodal Multilevel Converged Attention Network for Hand Gesture Recognition With Hybrid sEMG and A-Mode Ultrasound Sensing. *IEEE Trans. Cybern.* **2023**, *53*, 7723–7734. [CrossRef] [PubMed]
14. Szabo, T.L. Chapter 5—Transducers. In *Diagnostic Ultrasound Imaging: Inside Out*, 2nd ed.; Szabo, T.L., Ed.; Academic Press: Boston, MA, USA, 2014; pp. 121–165. [CrossRef]
15. Akhlaghi, N.; Dhawan, A.; Khan, A.A.; Mukherjee, B.; Diao, G.; Truong, C.; Sikdar, S. Sparsity Analysis of a Sonomyographic Muscle–Computer Interface. *IEEE Trans. Biomed. Eng.* **2020**, *67*, 688–696. [CrossRef] [PubMed]
16. Fernandes, A.J.; Ono, Y.; Ukwatta, E. Evaluation of Finger Flexion Classification at Reduced Lateral Spatial Resolutions of Ultrasound. *IEEE Access* **2021**, *9*, 24105–24118. [CrossRef]
17. Yang, X.; Chen, Z.; Hettiarachchi, N.; Yan, J.; Liu, H. A Wearable Ultrasound System for Sensing Muscular Morphological Deformations. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 3370–3379. [CrossRef]
18. Keller, K.; Leitner, C.; Baumgartner, C.; Benini, L.; Greco, F. Fully Printed Flexible Ultrasound Transducer for Medical Applications. *Adv. Mater. Technol.* **2023**, *8*, 2300577. [CrossRef]
19. Guo, J.Y.; Zheng, Y.P.; Huang, Q.H.; Chen, X.; He, J.F.; Chan, H.L.W. Performances of one-dimensional sonomyography and surface electromyography in tracking guided patterns of wrist extension. *Ultrasound Med. Biol.* **2009**, *35*, 894–902. [CrossRef] [PubMed]
20. Guo, J.Y.; Zheng, Y.P.; Kenney, L.P.; Bowen, A.; Howard, D.; Canderle, J.J. A Comparative Evaluation of Sonomyography, Electromyography, Force, and Wrist Angle in a Discrete Tracking Task. *Ultrasound Med. Biol.* **2011**, *37*, 884–891. [CrossRef] [PubMed]
21. Yang, X.; Yan, J.; Yin, Z.; Liu, H. Sonomyographic Prosthetic Interaction: Online Simultaneous and Proportional Control of Wrist and Hand Motions Using Semisupervised Learning. *IEEE/ASME Trans. Mechatronics* **2023**, *28*, 804–813. [CrossRef]

22. Sgambato, B.G.; Hasbani, M.H.; Barsakcioglu, D.Y.; Ibáñez, J.; Jakob, A.; Fournelle, M.; Tang, M.X.; Farina, D. High Performance Wearable Ultrasound as a Human-Machine Interface for wrist and hand kinematic tracking. *IEEE Trans. Biomed. Eng.* **2023**, *71*, 484–493. [CrossRef]
23. Yang, X.; Liu, Y.; Yin, Z.; Wang, P.; Deng, P.; Zhao, Z.; Liu, H. Simultaneous Prediction of Wrist and Hand Motions via Wearable Ultrasound Sensing for Natural Control of Hand Prostheses. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 2517–2527. [CrossRef] [PubMed]
24. Liu, D.; Zhang, D.; Zhang, G.; Liu, H. Dynamic Hand Gesture Recognition Based on A-Mode Ultrasound Sensing: Proposing an Algorithm Based on the Long Short-Term Memory Framework. *IEEE Syst. Man Cybern. Mag.* **2023**, *9*, 45–52. [CrossRef]
25. Rahimian, E.; Zabihi, S.; Asif, A.; Farina, D.; Atashzar, S.F.; Mohammadi, A. FS-HGR: Few-Shot Learning for Hand Gesture Recognition via Electromyography. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2021**, *29*, 1004–1015. [CrossRef] [PubMed]
26. Akhlaghi, N.; Baker, C.A.; Lahlou, M.; Zafar, H.; Murthy, K.G.; Rangwala, H.S.; Kosecka, J.; Joiner, W.M.; Pancrazio, J.J.; Sikdar, S. Real-Time Classification of Hand Motions Using Ultrasound Imaging of Forearm Muscles. *IEEE Trans. Biomed. Eng.* **2016**, *63*, 1687–1698. [CrossRef] [PubMed]
27. Li, S.; Sun, W.; Li, W.; Yu, H. Enhancing Robustness of Surface Electromyography Pattern Recognition at Different Arm Positions for Transhumeral Amputees Using Deep Adversarial Inception Domain Adaptation. *Appl. Sci.* **2024**, *14*, 3417. [CrossRef]
28. Zeng, J.; Zhou, Y.; Yang, Y.; Yan, J.; Liu, H. Fatigue-Sensitivity Comparison of sEMG and A-Mode Ultrasound based Hand Gesture Recognition. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 1718–1725. [CrossRef] [PubMed]
29. Yang, X.; Sun, X.; Zhou, D.; Li, Y.; Liu, H. Towards Wearable A-Mode Ultrasound Sensing for Real-Time Finger Motion Recognition. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2018**, *26*, 1199–1208. [CrossRef] [PubMed]
30. Wang, X.; Ao, D.; Li, L. Robust myoelectric pattern recognition methods for reducing users' calibration burden: challenges and future. *Front. Bioeng. Biotechnol.* **2024**, *12*, 1329209. [CrossRef]
31. Zeng, J.; Sheng, Y.; Zhou, Z.; Yang, Y.; Wang, J.; Xie, Q.; Liu, H. Adaptive Learning Against Muscle Fatigue for A-Mode Ultrasound-Based Gesture Recognition. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–10. [CrossRef]
32. Atzori, M.; Gijsberts, A.; Castellini, C.; Caputo, B.; Hager, A.G.M.; Elsig, S.; Giatsidis, G.; Bassetto, F.; Müller, H. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Sci. Data* **2014**, *1*, 140053. [CrossRef]
33. Bayram, F.; Ahmed, B.S.; Kassler, A. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowl.-Based Syst.* **2022**, *245*, 108632. [CrossRef]
34. Karimian, M.; Beigy, H. Concept drift handling: A domain adaptation perspective. *Expert Syst. Appl.* **2023**, *224*, 119946. [CrossRef]
35. Côté-Allard, U.; Fall, C.L.; Drouin, A.; Campeau-Lecours, A.; Gosselin, C.; Glette, K.; Laviolette, F.; Gosselin, B. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 760–771. [CrossRef] [PubMed]
36. Du, Y.; Jin, W.; Wei, W.; Hu, Y.; Geng, W. Surface EMG-Based Inter-Session Gesture Recognition Enhanced by Deep Domain Adaptation. *Sensors* **2017**, *17*, 458. [CrossRef]
37. Côté-Allard, U.; Gagnon-Turcotte, G.; Phinyomark, A.; Glette, K.; Scheme, E.J.; Laviolette, F.; Gosselin, B. Unsupervised Domain Adversarial Self-Calibration for Electromyography-Based Gesture Recognition. *IEEE Access* **2020**, *8*, 177941–177955. [CrossRef]
38. Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; Vaughan, J.W. A theory of learning from different domains. *Mach. Learn.* **2010**, *79*, 151–175. [CrossRef]
39. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
40. Shu, R.; Bui, H.H.; Narui, H.; Ermon, S. A dirt-t approach to unsupervised domain adaptation. *arXiv* **2018**, arXiv:1802.08735.
41. Miyato, T.; Maeda, S.I.; Koyama, M.; Ishii, S. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1979–1993. [CrossRef]
42. Liang, J.; Hu, D.; Wang, Y.; He, R.; Feng, J. Source Data-absent Unsupervised Domain Adaptation through Hypothesis Transfer and Labeling Transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 8602–8617. [CrossRef]
43. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; Volume 37, pp. 448–456.
44. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Volume 48, pp. 1050–1059.
45. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv* **2015**, arxiv:1505.00853.
46. Li, S.; Song, S.; Huang, G.; Ding, Z.; Wu, C. Domain Invariant and Class Discriminative Feature Learning for Visual Domain Adaptation. *IEEE Trans. Image Process.* **2018**, *27*, 4260–4273. [CrossRef] [PubMed]
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
48. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
49. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.

50. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 6105–6114.
51. Farrell, T.R.; Weir, R.F. The Optimal Controller Delay for Myoelectric Prostheses. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2007**, *15*, 111–118. [CrossRef] [PubMed]
52. Yang, X.; Zhou, Y.; Liu, H. Wearable Ultrasound-Based Decoding of Simultaneous Wrist/Hand Kinematics. *IEEE Trans. Ind. Electron.* **2021**, *68*, 8667–8675. [CrossRef]
53. Müller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2019; Volume 32.
54. Liebel, L.; Körner, M. Auxiliary Tasks in Multi-task Learning. *arXiv* **2018**, arxiv:1805.06334.
55. Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7482–7491.
56. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631.
57. Yang, X.; Yan, J.; Fang, Y.; Zhou, D.; Liu, H. Simultaneous Prediction of Wrist/Hand Motion via Wearable Ultrasound Sensing. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2020**, *28*, 970–977. [CrossRef]
58. French, R.M. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* **1999**, *3*, 128–135. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.