Linear Domain Adaptation for Robustness to Electrode Shifts

Rui Liu[1] and Benjamin Paaßen[1]

1- Bielefeld University, Faculty of Technology
{rliu,bpaassen}@techfak.uni-bielefeld.de
Inspiration 1, 33619 Bielefeld, Germany

Abstract. Machine learning approaches have shown impressive achievements in bionic prostheses control. However, translating the machine learning models from labs to patient's everyday lives remains a challenge due to various disturbances, such as electrodes shifts. To mitigate the influence of electrode shifts, we investigate two linear domain adaptation methods and a robust training approach. In experiments, we compare all methods on both simulated electrode shifts on the Ninapro DB2 data set as well as real electrode shifts on Ninapro DB8. We find that linear domain adaptation could estimate the shift and reduce the impact of electrodes shift best, but robust training approaches similar performance without the need for new data.

1 Introduction

Countless patients lose their hands every year, one major reason being traumatic injuries e.g. due to the Russian invasion of Ukraine [1]. For amputees, bionic protheses offer the promise of regaining some hand function [2]. Such prostheses are typically controlled by machine learning methods through surface electromyography (sEMG) of patients. Machine learning-wise, the prostheses control task is typically regarded as real-time sEMG motion classification: each time window of sEMG data is classified online, and the classification result is treated as a motion command for the controller of the prosthesis.

Unfortunately, the accuracy of these machine learning methods suffers due to a lack of robustness to disturbances like electrodes shifts. Electrodes shifts create a domain gap, resulting in two different domains: the undisturbed condition is the source domain, the condition disturbed by electrodes shifts is the target domain. A model trained in the undisturbed condition often fails to recognize the sEMG signals from the disturbed condition.

To address this problem, [3] have proposed linear domain adaptation methods which model the influence of electrode shift as a linear operator on the sEMG data. However, this prior work has focused exclusively on linear classifiers, whereas state-of-the-art sEMG classification is performed using convolutional neural nets [4]. Further, the linear domain adaptation scheme requires recording new data whenever a disturbance occurs. To address these shortcomings, we investigate the potential of linear domain adaptation in combination with convolutional neural nets and with a robust training approach which aims for a model that is robust against electrode shifts without the need for recording new data.

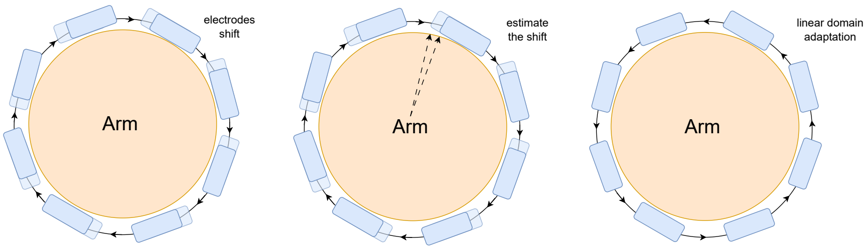

Figure 1: An illustration of linear domain adaptation [3]. Electrodes shift in a ring around the arm (left). This is assumed to be caused by a linear operator $\boldsymbol{T}(\alpha)$. The amount of shift α is estimated (center) and then corrected by applying the transpose $\boldsymbol{T}(\alpha)^T$ (right).

In summary, our contributions in this paper are 1) we investigate a variant of the linear domain adaptation approach of [3] based on the linear inverse; 2) we propose a novel robust training scheme which augments the training data with simulated electrode shifts, thus achieving more robustness against electrode shifts; 3) we evaluate the proposed approaches on state-of-the-art CNN-based sEMG classifiers on two real-world motion datasets, including data both from able-bodied and amputee participants.

2 Method

Our overall setting is as follows. We record an sEMG signal with a ring of C electrodes around the forearm ($C = 8$ in our data), apply standard filters, and then subdivide the signal into windows of M samples ($M = 400$ in our data) following the procedure described in [4]. This yields a time series of $M \times C$ matrices $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$, each of which shall be classified into one of K movements. On these data, we would train some classifier $f : \mathbb{R}^{M \times C} \to \{1, \ldots, K\}$. Our architecture for f is a three-layer 1D-convolutional neural net followed by a fully connected and a distance layer for few-shot classification, in line with [4].

Now, we consider electrode shifts. For disturbed data $\hat{\boldsymbol{x}}_t$ under electrode shifts, our classifier f will likely fail. In line with [3], we model the influence of an electrode shift as a linear operator on the channels. In particular, we assume that the value of the jth electrode in the disturbed condition can be described as a linear combination of the jth and $j-1$th electrode in the undisturbed condition (refer also to Fig. 1):

$$\hat{x}_{t,:,j} \approx \alpha \cdot x_{t,:,j-1} + (1-\alpha) \cdot x_{t,:,j}, \tag{1}$$

where $\alpha \in [0, 1]$ indicates how far the electrodes have shifted in units of inter-electrode spaces. Overall, we obtain $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t \cdot \boldsymbol{T}(\alpha)$ for the operator $\boldsymbol{T}(\alpha)$ given in Eq. (1). For shifts in the opposite direction, the transposed operator $\boldsymbol{T}(\alpha)^T$ is applied [3]. We emphasize that, in reality, the effects of electrode shifts are nonlinear; hence, our model is a simplification. We now consider three different schemes to counteract electrode shifts:

Linear domain adaptation (LDAd): Under the assumption of the linear model (1), [3] suggest to counteract electrode shifts by recording a small set of shifted data $\hat{\boldsymbol{x}}$ for a small subset of motions (only four). Then, we apply the transposed operator $\boldsymbol{T}(\alpha)^T$ for varying values of $\alpha \in [0, 1]$, and record the classification accuracy of f on the "cleaned" data $\hat{\boldsymbol{x}}_t \cdot \boldsymbol{T}(\alpha)^T$. We find the value α^* which maximizes the classification accuracy and finally apply $\boldsymbol{T}(\alpha^*)^T$ to all test data. This approach has been shown to effectively counteract electrode shifts both on simulated as well as real data, at least for linear classifiers [5, 3]. We will evaluate the effectiveness of the linear operator $\boldsymbol{T}$ also for convolutional neural nets and contrast it to the two following approaches.

Linear inverse domain adaptation (LinDAd): One inconsistency in the approach of [3] is that it uses $\boldsymbol{T}(\alpha)^T$ to counteract a shift $\boldsymbol{T}(\alpha)$. However, $\boldsymbol{T}(\alpha)^T$ is *not* the inverse of $\boldsymbol{T}(\alpha)$ (except for the special cases $\alpha = 0$ and $\alpha = 1$). Accordingly, we consider a variant of the linear domain adaptation approach above using the operator $\boldsymbol{T}(\alpha)^{-1}$ instead of $\boldsymbol{T}(\alpha)^T$ to counteract electrode shifts. Note that $\boldsymbol{T}(\alpha)$ is always invertible, except for $\alpha = 0.5$.

Robust training (RT): Linear domain adaptation requires recording new data whenever an electrode shift occurs. However, in everyday lives, it may be difficult to set up proper conditions to record training data. Hence, we would like to avoid any need for new data. Accordingly, we propose a data augmentation approach where we extend the training data $\boldsymbol{x}_t$ with shifted versions $\hat{\boldsymbol{x}}_t \cdot \boldsymbol{T}(\alpha)$ for various $\alpha \in [0, 1]$. Hence, we hope to improve the robustness of the model towards electrode shifts. We call this approach *robust training* in line with [6].

3 Experiments

In the experiments, we compare LDAd, LInDAd and RT on two data sets, NinaPro DB2 and NinaPro DB8. The experimental code is available at `https://gitlab.com/fewshotsiamesenetwork/linear-domain-adaptation`.

NinaPro DB2 includes sEMG data of 40 able-bodied participants performing three exercises: exercise B contains 9 thumb gestures and 8 wrist motions; exercise C involves 23 different movements; as well as exercise D includes flexions of fingers. In our experiments, we choose the 9 thumb gestures and 4 wrist motions from exercise B to evaluate our three domain adaptation methods and 4 wrist motions as calibration data to estimate α in LDAd and LInDAd. Training and test data sets consist of different motions to also evaluate few-shot accuracy in line with [4]. The sEMG data are recorded at 2 kHz with 14 electrodes. Of those, we use only the data from 8 electrodes placed in a ring around the forearm in line with Fig. 1. As this data set contains no electrode shifts, we simulate electrode shifts using the linear operator (1). The train data from 40 participants has 230,310 data points and test data has 114,171.

NinaPro DB8 contains sEMG data of 10 able-bodied participants and 2 ransradial amputees. Participants performed 10 movements such as rest, thumb

Table 1: Mean accuracy ($\pm$ std.) across participants of all methods on NinaPro DB2 for simulated linear shift with increasing α.

Method $\backslash\alpha$	$\boldsymbol{\alpha = 0.2}$	$\boldsymbol{\alpha = 0.3}$	$\boldsymbol{\alpha = 0.4}$	$\boldsymbol{\alpha = 0.6}$	$\boldsymbol{\alpha = 0.7}$	$\boldsymbol{\alpha = 0.8}$
base	0.36 ± 0.12	0.29 ± 0.10	0.23 ± 0.10	0.15 ± 0.07	0.13 ± 0.05	0.11 ± 0.05
LDAd	0.31 ± 0.11	0.27 ± 0.10	0.23 ± 0.09	0.25 ± 0.09	0.29 ± 0.10	0.34 ± 0.12
LInDAd	0.45 ± 0.11	0.44 ± 0.10	0.45 ± 0.10	0.45 ± 0.10	0.45 ± 0.11	0.45 ± 0.11
RT	0.43 ± 0.10	0.45 ± 0.10	0.46 ± 0.10	0.48 ± 0.10	0.47 ± 0.10	0.46 ± 0.10

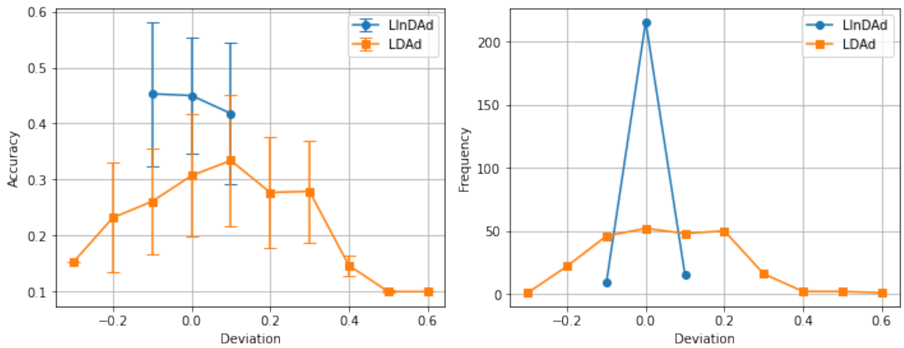

Figure 2: The accuracy (left) and frequency (right) versus the deviation between estimated and true α for Both LDAd and LInDAd.

extension, and grip. The sEMG data are recorded at 1111Hz by 16 wireless electrodes, placed in two rings of 8 electrodes each around the right forearm and the radiohumeral joint. The second ring is shifted by half an electrode relative to the first. We exploit this configuration to simulate electrode shifts, using the data from one ring as training and from the other as test data. The train data is 186,193 data points, and test data is 78040.

We employed the same Siamese network architecture and hyperparameters as in [4]. The training lasts 100 epochs for Ninapro DB2 and 50 epochs for DB8. L1 is chosen as our distance function. The learning rate decays from 0.0002 to 0.00001 for Ninapro DB2 and 0.0002 to 0.00008 for Ninapro DB8. The experiments are implemented with PyTorch and Cuda. All experiments are run on a PC with Intel Core i9-13900 CPU, 32 GB RAM, and NVIDIA RTX 4000 Ada GPU with 20GB VRAM.

Table 1 shows the results for NinaPro DB2 under simulated linear shift for increasing α. We observe that, with increasing α, the classification accuracy of our default classifier drops from 0.36 for $\alpha = 0.2$ to 0.11 for $\alpha = 0.8$. LDAd performs reasonably well for $\alpha = 0.2$ and $\alpha = 0.8$ but drops to an accuracy of 0.24 for intermediate $\alpha \in \{0.4, 0.6\}$. This is likely due to the fact that inverse and transpose of $\boldsymbol{T}(\alpha)^T$ coincide for $\alpha \in \{0, 1\}$ but diverge for intermediate values. By contrast, LInDAd retains the performance of the original model consistently at accuracy ≥ 0.44, which is to be expected given that the inverse perfectly

Table 2: Mean accuracy ($\pm$ std.) across participants of all methods on NinaPro DB8 in negative direction (left) and positive direction (right).

Method	**negative direction**	**positive direction**
base	0.22 ± 0.11	0.44 ± 0.18
LDAd	0.44 ± 0.19	0.50 ± 0.18
LInDAd	0.34 ± 0.22	0.31 ± 0.21
RT	0.43 ± 0.18	0.45 ± 0.19

counteracts the simulated shift.In more detail, we observe that LInDAd was substantially more accurate in estimating α (90% accuracy) whereas LDAd often estimated αs that were off by -0.2 to +0.2 (refer to Fig. 2, left). Even if LDAd estimated the shift correctly, the accuracy was lower, due to the aforementioned difference between transpose and inverse (refer to Fig. 2, right).

Interestingly, RT also performs well, retaining an accuracy of ≥ 0.43 across all shifts. The difference between LDAd and LInDAd as well as between LDAd and RT is significant ($p < 0.05$ in a Wilcoxon signed rank test).

Table 2 shows the results for NinaPro DB8 when adapting from one electrode ring to the other. In the negative direction (left) the baseline models drops to about 0.22 accuracy whereas LDAd and RT achieve accuracies around 0.43. For both methods, improvements are highly significant ($p < 0.005$ in a Wilcoxon signed-rank test). By contrast, LInDAd fails to significantly outperform the baseline model. This is likely due to the fact that the inverse estimates the signal of a clean electrode utilizing data from all other electrodes in the ring, yielding undesirable cross-influences under conditions of actual (rather than simulated) electrode shift. By contrast, LDAd is less affected by violations of the linearity assumption as it only uses data from neighboring electrodes (refer to Eq. (1)).

If we adapt in the other direction in NinaPro DB8, we observe a different effect: Here, the original model still performs well at around 0.44 and no domain adaptation method can improve significantly upon this strong baseline. This is likely due to the fact that the source data in this direction is recorded close to the radiohumeral joint, which decreases the signal quality in the training data, thus forcing the classifier to be more robust in the first place. Still, we observe slight (albeit non-significant) improvements for LDAd and RT.

Given that the goal of our work is to support amputees, Table 3 reports the results of LDAd for the two amputees in NinaPro DB8. We observe that accuracies are notably higher than the average reported in Table 2, likely due to the fact that both amputees already have multiple years of training in controlling a prosthesis, thus yielding sEMG signals that are easier to classify.

The robust training model needs a longer time to train; it takes an average of 650 seconds across patients to train on NinaPro DB8, which is 411 seconds longer than training without the data augmentation. Note that the inference

Table 3: Mean accuracy ($\pm$ std.) across 10 repeats of LDAd for the two amputees in NinaPro DB8 in negative direction (left) and positive direction (right).

cause of amputation	**negative direction**	**positive direction**
Accident	0.619 ± 0.03	0.690 ± 0.01
Epithelioid sarcoma	0.446 ± 0.19	0.567 ± 0.05

times are the same because the model architecture remains constant.

4 Conclusion

In the setting of sEMG classification under electrode shifts, we compared the linear domain adaptation method by [3] to two novel methods: a linear inverse domain adaptation, and a robust training approach. Our experiments covered both simulated (NinaPro DB2) and real (NinaPro DB8) electrode shifts. For simulated shifts, the novel linear inverse model performed best. However, this method broke down for real shifts, where the underlying linearity assumption does not hold. By contrast, linear domain adaptation and robust training (RT) performed better and significantly improved upon the disturbed model. The improved performance for RT is particularly promising, given that this method requires no new data recording and only requires some overhead at training time but no overhead at test time. In future work, further ablation studies should validate the impact of RT hyperparameters; RT should be validated in user studies in an on-line setting, and other adaptation methods should be explored, such as pre-processing to make the linearity assumption plausible; or non-linear domain adaptation methods.

References

[1] I. Trutyak, V. Malickii, M. Samotowka, V. Trunkvalter, R. Trutyak, and V. Ivaschenko. Problematic issues of limb amputation in wounded with combat trauma. *Proceeding of the Shevchenko Scientific Society. Medical Sciences*, 72(2), 2023.

[2] D. Farina, I. Vujaklija, R. Brånemark, A.M.J. Bull, H. Dietl, B. Graimann, L.J. Hargrove, K.-P. Hoffmann, H. Huang, T. Ingvarsson, et al. Toward higher-performance bionic limbs for wider clinical use. *Nature biomedical engineering*, 7(4):473–485, 2023.

[3] C. Prahm, A. Schulz, B. Paaßen, J. Schoisswohl, E. Kaniusas, G. Dorffner, B. Hammer, and O. Aszmann. Counteracting electrode shifts in upper-limb prosthesis control via transfer learning. *IEEE TNSRE*, 27(5):956–962, 2019.

[4] R. Liu and B. Paaßen. Few-shot similarity learning for motion classification via electromyography. In *Proceedings of the ESANN*, 2024.

[5] B. Paaßen, A. Schulz, J. Hahne, and B. Hammer. Expectation maximization transfer learning and its application for bionic hand prostheses. *Neurocomputing*, 298:122–133, 2018.

[6] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv*, 1805.12152, 2019.