TRANSFER LEARNING BASED ON OPTIMAL TRANSPORT FOR MOTOR IMAGERY BRAIN-COMPUTER INTERFACES

Victoria Peterson*[1], Nicolás Nieto[2], Dominik Wyser[3], Olivier Lambercy[3], Roger Gassert[3], Diego H. Milone[2] and Rubén D. Spies[1]

[1]*Instituto de Matemática Aplicada del Litoral, IMAL, UNL, CONICET, Santa Fe, Argentina*
[2]*Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional, sinc(i), FICH-UNL, CONICET, Santa Fe, Argentina.*
[3]*Rehabilitation Engineering Laboratory, Department of Health Sciences and Technology, ETH Zurich, Zurich, Switzerland*

Abstract: Objective: This paper tackles the cross-sessions variability of electroencephalography-based brain-computer interfaces (BCIs) in order to avoid the lengthy recalibration step of the decoding method before every use. Methods: We develop a new approach of domain adaptation based on optimal transport to tackle brain signal variability between sessions of motor imagery BCIs. We propose a backward method where, unlike the original formulation, the data from a new session are transported to a calibration session, and thereby avoiding model retraining. Several domain adaptation approaches are evaluated and compared. We simulated two possible online scenarios: i) block-wise adaptation and ii) sample-wise adaptation. In this study, we collect a dataset of 10 subjects performing a hand motor imagery task in 2 sessions. A publicly available dataset is also used. Results: For the first scenario, results indicate that classifier retraining can be avoided by means of our backward formulation yielding to equivalent classification performance as compared to retraining solutions. In the second scenario, classification performance rises up to 90.23% overall accuracy when the label of the indicated mental task is used to learn the transport. Adaptive time is between 10 and 80 times faster than the other methods. Conclusions: The proposed method is able to mitigate the cross-session variability in motor imagery BCIs. Significance: The backward formulation is an efficient retraining-free approach built to avoid lengthy calibration times. Thus, the BCI can be actively used after just a few minutes of setup. This is important for practical applications such as BCI-based motor rehabilitation.

Keywords:
Brain-Computer Interfaces, Domain Adaptation, Motor Imagery, Optimal Transport, Transfer Learning.

1 INTRODUCTION

Brain-computer interfaces (BCI) based on electroencephalography (EEG) can be used as a rehabilitation approach to improve the functional ability of people with severe sensorimotor impairments [1–4]. In this context, the brain activity associated to a mental motor task, or an actual motor attempt, must be decoded to control an external device, such as a virtual reality avatar providing visual feedback [5], a robotic device providing direct physical support [6], or the combination of both [7]. In order to implement and validate a robust BCI, two phases need to be completed: i) a *calibration* phase, in which the subject-specific decoding model is learned, mapping electrical brain activity patterns to output commands, and ii) a *testing* phase, in which the learned model is applied and the actual command is generated [8]. However, different sources of varibility in the EEG signal affect the ability to robustly detect the intended mental task. Changes at the level of the subject, the electrode position or the type of feedback presented, are potential sources of variability. The lack of stationary of the EEG can be observed between subjects but also within and between sessions of the same subject [9]. Thus, in view of real-life EEG-based BCI applications, the decoding algorithm must be able to deal with such variability in a non time-consuming manner.

Within the BCI community, a standard approach to tackle EEG variability consists of learning a new decoding algorithm before each and every use (recalibration). Although such an approach can lead to good classification performance (accuracy $> 70\%$), it is not only time-consuming (generally a large amount of data must be recorded) but it also neglects all the information from previously collected data. Another strategy consists of finding the shared structure in the feature space across training data of multiple sessions and subjects [10].

[*] Corresponding author: vpeterson@santafe-conicet.gov.ar

Manuscript accepted for publication in IEEE Transactions on Biomedical Engineering on August 09, 2021

The EEG's nonstationarity can be thought of as a data distribution drift between calibration and testing recordings [11]. Domain adaptation, a particular case of transfer learning (TL), provides a way of devising models that can cope with such data shift [12]. In the context of BCIs, domain adaptation aims at addressing the within or between-subjects variability by adapting features and/or classifier parameters from one domain, e.g. one subject or session, to another domain, e.g. another subject or session of the same subject.

The feature extraction method most widely used in rehabilitative BCIs is the common spatial pattern (CSP) algorithm, first used to distinguish between the mental imagination (motor imagery, MI) of opening and closing the right vs. left hand [13]. In short, CSP spans band-pass filtered EEG data into a discriminative subspace by applying a linear mapping which maximizes the variance of one class and minimizes it for the other class [14]. Despite its popularity, CSP is sensitive to both data variations and small training datasets [15]. Different solutions have been proposed to tackle these drawbacks, ranging from adaptations at the level of the method itself [15], or changes in the classifier being used [16], to transformations of the feature space [9, 17].

In the specific context of neurorehabilitation applications, a subject-specific MI-BCI system is expected to be used across several sessions (days). The MI-decoding model will be subject to not only physiological noise, but potentially also to the brain pattern changes induced with recovery. In this context, in order to provide accurate feedback to the user as soon as the session begins, the decoding algorithm can be either *domain-invariant* or *domain-adaptive* to cross-sessions distribution drifts. Whereas in the former the decoding algorithm is designed so as to find common components between domains, in the latter the data distribution drift is learned so as to make the calibration and testing distributions more similar. This approach considers that samples (EEG trials) from two different domains (sessions) may have different data distributions, but have the same conditional distributions of the mental tasks with respect to the data [18].

Every domain-adaptive method seeks to "match" the probability distributions coming from two different domains: the *source* and the *target*. A recent approach within the BCI community for matching those distributions consists of domain alignments based on simple geometrical transformations of the data. In particular, the authors in [19] proposed what is called Riemannian alignment, a TL framework that consists of re-centering the covariance matrices of both domains in order to make them comparable. The Riemannian procrustes analysis (RPA) [11], which can be viewed as an improvement of the aforementioned TL method, also re-centers data as the first step of a series of geometrical transformations. More recently, the Euclidean alignment (EA) [20] was proposed with the main advantage of transforming the EEG trials in the Euclidean space, and thus any machine learning pipeline can then be applied to the aligned trials. Another relevant difference of EA as compared to RPA is that while the latter makes use of the available label information, the former is a completely unsupervised TL method. Regardless of how the alignment is made, all these methods transform data from both domains, and then a new classifier must be learned with the transformed training data.

Within the machine learning community attention has been paid to the use of optimal transport (OT) [21] as a TL approach. The use of optimal transport for domain adaptation (OTDA) was first proposed in [22], where the authors make use of OT theory to compute the optimal coupling between two probability distributions in a cost-effective manner. Once the optimal coupling is computed, the optimal transport is constructed in terms of it, and the data from the source domain are then transformed to make its probability distribution more "similar" to the distribution in the target domain. A new classifier is then learned using the transported source data such that the label prediction in the target domain improves. The use of OTDA for transfer learning in BCI has already been evaluated in [23] for P300-BCI applications. Although the authors show that data distribution drift can potentially be addressed by keeping the classifier fixed and transforming the testing data, the transportation plan was learned by using testing data, an approach that is not applicable to online scenarios.

In this work, we provide a novel framework of domain adaptation based on OT for addressing the cross-session variability in online MI-BCI. Considering real-time applications of BCI systems for motor rehabilitation, we are interested in finding a solution to the problem of long calibration times and classifier retraining between sessions. In order to do so, we propose a new backward method, called backward optimal transport for domain adaptation (BOTDA), which transforms target samples to boost the performance of the already trained classifier. Unlike the approach followed in [23], the transport operator is directly calculated in an optimal way, without inverting the forward operator. By using BOTDA, both classifier retraining and long calibration data before each session can be avoided. In addition, a complete online compatible workflow for applying TL based on OTDA is presented here. This work is a contribution towards optimal strategies for the development of robust MI-BCI systems. It opens new avenues for increasing time-efficiency and performance of BCI-based neurorehabilitation approaches by better addressing EEG variability.

The organization of this article is as follows. Section 2 describes the problem, introduces OTDA as well as our backward approach and explains the use of OT in the context of BCI. Section 3 describes the two real multiple-sessions MI-BCI datasets used throughout this work. Experiments and results are presented in Section 4 whereas discussions and conclusions, including future works, are in Sections 5 and 6, respectively.

2 Transfer learning and domain adaptation based on optimal transport

This section introduces some of the mathematical assumptions and foundations of domain adaptation based on optimal transport. The traditional as well as the proposed backward formulation are detailed.

2.1 Assumptions and main definitions

Throughout this work, we consider that a domain is a BCI session, and thus, the *source* domain Ω_s refers to the calibration session, whereas the *target* domain Ω_t refers to the testing session. We analyze different regularized OTDA versions proposed by Courty et al. [22] from two perspectives: i) the forward approach (original), in which a forward transport mapping F is learned from the source to the target domain and ii) the backward approach, our proposed new alternative, which learns a backward transport mapping B directly from the target to the source domain. Fig. 1 illustrates the two aforementioned OTDA alternatives: i) forward OTDA (FOTDA) and ii) backward OTDA (BOTDA). For learning the transportation plan by either of the two OTDA alternatives, data from the new session is needed. With real-life applications in mind, we provide a domain adaptation algorithm based on optimal transport for MI-BCI that can be completely implemented online. In this context, we simulate two possible online scenarios: block-wise adaptation and sample-wise adaptation. In the first scenario we hypothesize that the data distribution drift remains unchanged from one data-block (testing run) to another, and therefore it can be learned from the previous available data. On the contrary, in the second scenario, we assume that the data distribution is continuously changing and thus the current testing trial should be considered in the transport learning process. Although experiments were conducted as online simulations, throughout this work we shall indistinguishably refer to them as online testing session since this is how our workflow was designed. That is, all the proposed methods can be applied in fully online scenarios.

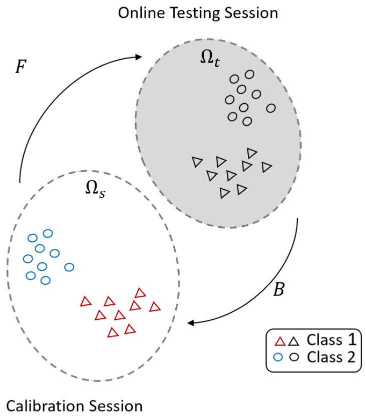

Figure 1: Illustration of two different domain adaptation approaches: forward OTDA, where the distribution drift is learned from *source* to *target* domain and backward OTDA, where the transportation mapping is learned from *target* to *source* domain. Here Ω_s and Ω_t denote the source and target domains, respectively, and F and B denote the forward and backward transport mappings, respectively.

2.2 Problem formulation

Consider two datasets, the *source* ($\mathcal{S}$) and the *target* ($\mathcal{T}$) dataset, coming from two different MI-BCI sessions (e.g., days). Each dataset is composed of EEG trials belonging to one and only one of two MI classes. Let N_s and N_t denote the number of trials in the source and target domains, respectively, and $\mathbf{x}_i^s \in \Omega_s, \mathbf{x}_i^t \in \Omega_t$ the feature vectors, where $\Omega_s,\, \Omega_t \subset \mathbb{R}^d$, $\mathcal{K} \doteq \{k_1, k_2\}$ and $y_i^s, y_i^t \in \mathcal{K}$ are class labels. Hence our datasets are

$$\mathcal{S} \doteq \{(\mathbf{x}_i^s, y_i^s), i = 1, \ldots, N_s\} \subset \Omega_s \times \mathcal{K},$$
$$\mathcal{T} \doteq \{(\mathbf{x}_i^t, y_i^t), i = 1, \ldots, N_t\} \subset \Omega_t \times \mathcal{K}.$$

We shall denote with $X_s \in \mathbb{R}^{N_s \times d}$ and $X_t \in \mathbb{R}^{N_t \times d}$ the feature matrices whose i^{th} rows are $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$, respectively.

In every standard learning paradigm, the discriminative model is constructed using a training set (with label information) and then evaluated on the unseen testing set. This learning-evaluation framework assumes that training and testing sets are drawn from the same distribution. However, in multiple-sessions EEG-based BCIs this assumption cannot be guaranteed since training and testing sets are collected at different times by different system conditions, which may lead to poor classification performances. This issue can be modeled as a covariate shift problem, in which changes in the distributions in the two domains are considered ($\mathbf{P}_s(\mathbf{x}^s) \neq \mathbf{P}_t(\mathbf{x}^t)$), but it is assumed that the conditional distributions of the labels with respect to the data remain the same ($\mathbf{P}_s(y|\mathbf{x}^s) = \mathbf{P}_t(y|\mathbf{x}^t)$). Assuming that the domain drift is given by a certain transformation $F : \Omega_s \to \Omega_t$, if such a transformation is known, one can then "adapt" the domains so as to make the distributions similar, and thus prevent classification failure. The domain adaptation approach based on optimal transport (OTDA) consists of estimating such a transformation F in a cost-effective manner [22], as described in the following subsection.

2.3 A brief introduction to OTDA

OT theory studies a problem known as the Monge-Kantorovich transportation problem [21], which, roughly speaking, seeks to find a cost-effective way to transport mass between two probability distributions. In this direction, it is said that OT solves and optimization problem which minimizes what is called transportation cost. For $\Omega \subset \mathbb{R}^d$ let $\mathcal{P}(\Omega)$ denote the space of all probability measures with support in Ω. Given $\Omega_s,\, \Omega_t \subset \mathbb{R}^d$, a measurable mapping $F : \Omega_s \to \Omega_t$, as above, and a measure $\alpha \in \mathcal{P}(\Omega_s)$, the measure $\sigma \in \mathcal{P}(\Omega_t)$ defined by $\sigma(A) \doteq \alpha\left(F^{-1}(A)\right)$ for every σ-measurable set $A \subset \Omega_t$, is denoted by $F\#\alpha$. The mapping F is said to be a push-forward or a transport map of α in σ, and σ is referred to as the *push-forward* of α by F.

In this context, addressing the data distribution drift by the forward OTDA problem consists of finding a transport map F of the source data distribution $\mu_s \in \mathcal{P}(\Omega_s)$ in the target data distribution $\mu_t \in \mathcal{P}(\Omega_t)$, i.e. such that $F\#\mu_s = \mu_t$. This adaptation problem can be tackled by following the next three steps: first, estimate the measures μ_s and μ_t from the feature matrices X_s and X_t, then find a transport map F from μ_s in μ_t, and finally use F to transport X_s. Although that sounds simple, searching for F in the space of all possible transformations turns out to be an absolutely unmanageable problem and appropriate restrictions on F must be imposed. In this line of work, the Monge approach [24] to the optimal transport problem consists of finding, among all the possible transports F from μ_s in μ_t, an optimal transformation F_0 as the solution of the minimization problem

$$F_0 \;\doteq\; \underset{F \ s.t.\ F\#\mu_s = \mu_t}{\operatorname{argmin}} \int_{\Omega_s} c(\mathbf{x}, F(\mathbf{x}))\, d\mu_s(\mathbf{x}),$$

where $c : \Omega_s \times \Omega_t \to \mathbb{R}^+$ is a given cost function.

A convex relaxation of the OT problem was introduced by Kantorovich et al. [24]. Given $\mu_s \in \mathcal{P}(\Omega_s)$ and $\mu_t \in \mathcal{P}(\Omega_t)$, let $\Upsilon = \Upsilon(\mu_s, \mu_t)$ be the set of all couplings between μ_s and μ_t, i.e. all joint probability measures γ in $\mathcal{P}(\Omega_s \times \Omega_t)$ whose respective marginals are μ_s and μ_t. The Kantorovich approach to OT consists of finding the optimal transportation plan, which is defined as the coupling γ_0 given by

$$\gamma_0 \ \doteq \ \operatorname*{argmin}_{\gamma\in\Upsilon} \int_{\Omega_s\times\Omega_t} c(\mathbf{x}^s,\mathbf{x}^t)\ d\gamma(\mathbf{x}^s,\mathbf{x}^t). \tag{1}$$

When the measures μ_s and μ_t are to be estimated from the feature matrices X_s and X_t respectively, as in our case, the corresponding sample distributions take the form

$$m_s \ = \ \sum_{i=1}^{N_s} p_i^s \delta_{\mathbf{x}_i^s} \quad \text{and} \quad m_t \ = \ \sum_{j=1}^{N_t} p_j^t \delta_{\mathbf{x}_j^t}, \tag{2}$$

where p_i^s and p_j^t are the probability masses at the points $\mathbf{x}_i^s$ and $\mathbf{x}_j^t$, respectively, and $\delta_x \in \mathcal{P}(\Omega)$ denotes the unit Dirac delta measure at the point $\mathbf{x} \in \Omega$. Usually $p_i^s = \frac{1}{N_s}\ \forall i$ and $p_j^t = \frac{1}{N_t}\ \forall j$. The continuous Kantorovich formulation of OT (1) has an immediate correlate in this discrete case. In fact, if we now denote by $A_s \doteq \Pi_1\mathcal{S}$ and $A_t \doteq \Pi_1\mathcal{T}$ the projections of the datasets $\mathcal{S}$ and $\mathcal{T}$ into Ω_s and Ω_t, respectively, and with Γ the set of all discrete probabilistic couplings between the discrete measures m_s and m_t, i.e.

$$\Gamma = \left\{ \gamma = (\gamma_{ij}) \in \mathcal{P}(A_s \times A_t) \text{ s.t.} \sum_j \gamma_{ij} = m_s, \sum_i \gamma_{ij} = m_t \right\},$$

then the discrete Kantorovich formulation of OT boils down to finding the optimal discrete transportation plan, defined as

$$\gamma_0 \ \doteq \ \operatorname*{argmin}_{\gamma\in\Gamma} \langle \gamma, C\rangle_F, \tag{3}$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product, and $C = (c_{ij}),\ c_{ij} \geq 0$ is a cost function matrix with $c_{ij} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ representing the cost of moving a unit probability mass from $\mathbf{x}_i^s \in \Omega_s$ to $\mathbf{x}_j^t \in \Omega_t$. Quite often c is simply the square L_2 norm or the Euclidean distance.

Several regularized versions of the discrete OT problem (3) exist. In [25], the authors proposed to add a regularizer to the formulation in (3) in order to reduce the sparsity of the transportation plan. The optimal coupling was then defined as:

$$\gamma_0 \ \doteq \ \operatorname*{argmin}_{\gamma\in\Gamma} \langle \gamma, C\rangle_F + \lambda W_e(\gamma), \tag{4}$$

where λ is a positive constant called regularization parameter, and $W_e(\gamma) \doteq \sum_{ij} \gamma_{ij}\log(\gamma_{ij})$ is the negentropy of γ. This formulation of OT not only favors smoother versions of the transport by reducing its sparsity, but it also allows for the use of computationally efficient algorithms based on the Sinkhorn-Knopp's scaling matrix approach [26]. In the sequel, we shall refer to this regularized OT version as OT-S.

In addition, to take advantage of the label information available in the source domain, an extra penalizer term can be added to (4) in the following way:

$$\gamma_0 \ \doteq \ \operatorname*{argmin}_{\gamma\in\Gamma} \langle \gamma, C\rangle_F + \lambda W_e(\gamma) + \eta W_c(\gamma), \tag{5}$$

where $\eta > 0$ is also a regularization parameter and $W_c(\gamma)$ is a regularization term which introduces label information into the OT formulation. For instance, the group-lasso regularizer [27] on the columns of γ can be used to penalize couplings that take any two samples in the source domain having different labels to the same sample in the target domain, as follows:

$$W_c(\gamma) \doteq \sum_j \sum_k \|\gamma(\mathcal{I}_k, j)\|_2,$$

where $\mathcal{I}_k$ denotes the set of indices of all the rows of γ corresponding to all the source domain samples of class $k \in \mathcal{K}$ [22].

For domain adaptation, once the optimal transportation plan γ_0 is found, source samples have to be transported to the target domain. For that, we need to define a transport map F in terms of γ_0. This mapping can be conveniently expressed for each $\mathbf{x}_i^s$ as the following c-based barycentric mapping:

$$\hat{\mathbf{x}}_i^s = F_{\gamma_0}(\mathbf{x}_i^s) \doteq \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{j=1}^{N_t} \gamma_0(i,j)\, c(\mathbf{x},\mathbf{x}_j^t), \qquad \text{for all } i = 1,\ldots,N_s. \tag{6}$$

In the discrete, regularized with class-labeled information formulation of OT (5), to which we shall refer in the sequel as OT-GL, it is expected that each target sample receives masses from source samples in the source domain that have the same label. For domain adaptation it is thus expected that the prior distributions of the labels are preserved in both domains. Fortunately, this assumption is not a problem for the current MI-BCI application considered throughout this work, as it will be explained in Subsection 2.5.

2.4 Backward OTDA

The forward OTDA maps source samples into the target domain by the transport map F_{γ_0} and then a new classifier is trained with those transformed source samples. If classifier retraining is to be avoided, rather than transforming the source samples, target samples can be transported to fit the source data distribution. However, we can not guarantee that the transport map F_{γ_0} defined in (6) be invertible. Although clearly a transport map can be defined in terms of γ_0^T as $\hat{\mathbf{x}}_i^t = F_{\gamma_0^T}(\mathbf{x}_i^t)$, it might happen that γ_0^T as a coupling between m_t and m_s is not optimal. With this in mind, we provide an alternative way to learn the mapping from the target to the source domain, in which the inversion of the F operator is avoided, ensuring that the learned mapping is optimal.

Given that in BCI applications it is common to acquire a set of data at the beginning of each new session for updating the model (recalibration), the transportation plan can be learned from the target to the source domains (see Fig. 1). This novel approach addresses the data drift from the target domain (new session) into the source domain (old session), reason for which we shall refer to it as backward OTDA (BOTDA). With ζ_0 and B_{ζ_0} we will denote the optimal transport plan as obtained by (4) or (5) when the backward approach is used, and the corresponding transport map, respectively. Hence, target samples can be transformed as follows:

$$\hat{\mathbf{x}}_i^t = B_{\zeta_0}(\mathbf{x}_i^t) \doteq \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{j=1}^{N_s} \zeta_0(i,j)\, c(\mathbf{x},\mathbf{x}_j^s), \qquad \text{for all } i = 1,\ldots,N_t. \tag{7}$$

In summary, BOTDA learns a mapping from the target to the source domain in a cost-effective manner, and the classifier (already trained with the source data) is tested with the transformed target data. Since the implementation of BOTDA relies on the same optimizer as FOTDA [25, 28], from the algorithmic point of view the novelty relies on how to build the source and the target datasets.

2.5 OTDA applied to BCI

In the present work it will always be assumed that the discriminative model is learned using data from the source domain (calibration session) and then evaluated in the target domain, representing a new BCI session. In spite of the OTDA approach that we investigate within this paper, in order to learn the transportation plan, data from the target domain are always needed. In this work, we shall refer as *transportation set*, $\mathcal{V} = \{(\mathbf{x}_i^v, y_i^v)\}_{i=1}^{N_v} \subset \mathcal{T}$, $N_v \ll N_t$, to such portion of data coming from the target domain which is used to learn the transportation plan. With $X_v \in \mathbb{R}^{N_v \times d}$ we shall denote the transportation feature matrix whose i^{th} row is $\mathbf{x}_i^v$.

Unlike other classification problems in which target data are fully available, for online brain signal detection the target data become available one trial at a time. This is undoubtedly a restriction to the way the data distribution drift can be learned. With this in mind, we simulate two possible online scenarios to learn the mapping: block-wise adaptation and sample-wise adaptation. In the first scenario we assume that the new BCI session is divided into data blocks of n_t trials each, which we call runs.With the aim of keeping the recalibration time as short as possible while having enough data to learn the mapping, we set $n_t = 20$, which comprises around 10 trails per class[1]. In addition, we hypothesize that the distribution drift of a testing run is equal to the distribution drift of the previous runs. We also assume here that the first session is meant for calibration whereas the first run of the new session (target domain) is meant for recalibration purposes, $R_0 = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{n_t}$. The remaining n_r runs are considered as online testing runs ($R_1, \ldots, R_{n_r}$). In this context, the prior available data in the target domain must be wisely used to learn the OT mapping. In this direction, for every OTDA alternative, we decide to learn the mapping using all the available testing data up to the current r^{th} run, as follows:

$$\mathcal{V}_r \doteq R_0 \cup \bigcup_{j=1}^{r-1} R_j, \quad r = 1, 2, \ldots, n_r. \tag{8}$$

Now, how can we make sample-wise learning of EEG distribution drift? In rehabilitative BCIs, a synchronous paradigm is used for both the calibration and the testing-with-feedback phases. In a synchronous BCI, generally by means of a visual cue, the system indicates the user when to start making a certain mental task [29], and thus the feedback provided to the user in the evaluation phase is based on what he/she did and can be compared to what he/she should have done. Given that in such cue-based environments the indicated mental task of a trial is always known, we provide a way of continuously adapting the data by considering the current i^{th} trial as part of the transportation set, i.e. we define:

$$\mathcal{V}_i \doteq R_0 \cup \bigcup_{j=n_t+1}^{i} \{(\mathbf{x}_j^t, y_j^t)\} \subset \mathcal{T}, \quad \forall i = n_t + 1, \ldots, N_t \tag{9}$$

It is important to highlight here that the current label information included in $\mathcal{V}_i$ is only used in the BOTDA with group-lasso penalty.

For MI-BCI, a simple but effective discriminative model can be built based on CSP and a linear classifier, such as linear discriminative analysis (LDA) [30]. Within this standard setup, applying OTDA involves four main steps (calibration, mapping learning, data transformation and classifier evaluation), as shown in Fig. 2. In the calibration phase ($\mathbf{C}$), the calibration data from the source domain are used for learning the CSP features ($\mathbf{C_1}$) and an initial LDA classifier ($\mathbf{C_2}$). In the online testing session ($\mathbf{T}$), the trained CSP model (represented here by the matrix W) is applied at the target domain ($\mathbf{T_1}$) in both the transportation and the testing sets. Both OTDA mappings (forward and backward) are learned using the transportation and the calibration sets ($\mathbf{T_2}$). While in the BOTDA alternative (light yellow arrow) testing features are transformed ($\mathbf{T_3}$) to feed the already trained classifier ($\mathbf{T_4}$), in the case of FOTDA (light orange arrow) the transformation is applied on the calibration feature set ($\mathbf{T_5}$) and then a new classifier $\hat{g}(\cdot)$ is trained using the transformed data matrix $\hat{X}_s$ ($\mathbf{T_6}$). This classifier is evaluated with the original testing feature set X_t ($\mathbf{T_7}$). Note that the label information will only be used for learning the mapping when the OT-GL formulation is used in either of the two OTDA alternatives, as explained below.

In this work four OTDA alternatives are tested. Although for each method the training or testing features can be different, the discriminative model is always CSP+LDA. We compare the classification performances of each OTDA approach against: a BCI model without transfer learning, a recalibration procedure already used in rehabilitative BCI [31], and two state-of-the-art data alignment methods. Subsequently, all investigated processing methods are summarized:

- **Standard with calibration (SC):** train the model with source data (calibration) and do the classification on the target dataset (new session) with no transformation whatsoever.

[1] Larger n_t values were tested, namely n_t equals to 40 and 80, yielding to similar overall classification results as when $n_t = 20$ but with more associated computational time.

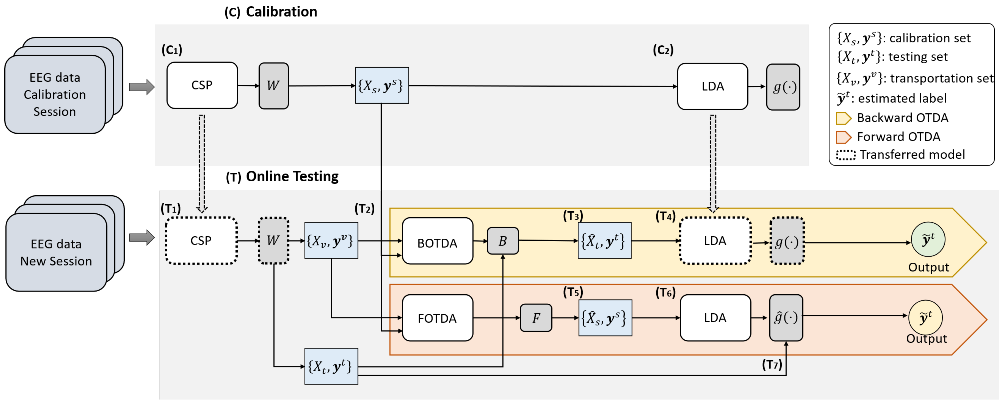

Figure 2: Schematic representation of the OTDA pipeline from calibration $(\mathbf{C})$ to online testing $(\mathbf{T})$ when either the backward (light yellow arrow) or the forward approach (light orange arrow) is being used. Note that: already trained models from Calibration Session to the New Session are drawn with dashes lines; methods are depicted in white boxes; the outputs of each model are shown in gray boxes; and feature sets are drawn in light blue rectangles. Since the outputs (predicted class) of the forward and the backward alternative may not be the same, two different colors were used to show the output label in each path.

- **Standard with recalibration (SR):** employ model retraining (CSP+LDA) by updating the training data with the previous testing run and eliminating the oldest one, as proposed by Ang et al. [31]. Note that this strategy is made on the raw data space, before feature extraction, and thus a new CSP model is constructed at each data updating step.

- **Riemannian procrustes analysis (RPA):** the sample covariance matrices are geometrically transformed (translated, scaled and rotated) in a semi-supervised way [11]. The approach is performed in the Riemannian manifold. After the source and the target data are transformed, the Minimum-Distance to Mean (MDM) classifier [32] is trained and tested on the transformed target dataset.

- **Euclidean alignment (EA):** performs an unsupervised data alignment using the arithmetic mean of all covariance matrices to transformation the data in the Euclidean space [20]. Like RPA, both source and target data are adapted. The decoding model is then learned in the transformed source samples and applied to the transformed target dataset.

- **FOTDA-S:** employ Sinkhorn OTDA formulation (4) to learn the mapping between the *source* and *target* domains. Source samples are then transformed to train a new classifier in which target samples are tested.

- **FOTDA-GL:** employ group-lasso OTDA formulation (5) to learn the mapping between the *source* and *target* domains. Source samples are then transformed to train a new classifier in which target samples are tested.

- **BOTDA-S:** employ Sinkhorn OTDA formulation (4) in the backward approach so as to learn the mapping from the *target* to the *source* domains. Target samples are transformed to feed the already trained classifier with the source data.

- **BOTDA-GL:** employ group-lasso OTDA formulation (5) in the backward approach so as to learn the mapping from *target* to *source* domains. Target samples are transformed to feed the already trained classifier with the source data.

3 Databases

Two different EEG-based BCI datasets were used in this study, each comprising at least two MI-BCI sessions of multiple participants. Since we aimed at providing an adaptive strategy for online testing in a motor rehabilitation scenario, where typically only binary decisions are required (e.g., to trigger an external device), both datasets comprised two MI classes.

Dataset-1: this dataset was collected from 10 naive able-bodied BCI users (3 females, 4 left-handed, mean age ± SD = 25.45 ± 2.50 years) on two different days (sessions) with a session-to-session separation of 5 days maximum. The experiment was approved by the local ethics committee (BASEC-Nr. Req-2017-00631, Cantonal Ethics Commission, Zurich, Switzerland). A portable 64-channel EEG system (eegort Ant Neuro, Netherlands) was used for brain signal recording, with a sampling frequency of 512 Hz. Surface electrodes were placed in accordance with the international 10-20 system, using CPz as reference and AFz as ground electrodes. EEG signals were band-pass filtered between 0.5 Hz and 40 Hz. Two mental tasks were performed by the subjects: i) the kinesthetic imagination of movement of their dominant hand (grasping movement) and ii) a rest/relax condition. Each session was composed of four runs separated by short breaks. Each run consisted of 40 trials (20 for each condition), yielding a total of 160 trials at the end of each session. No feedback was provided to the subject during the sessions. For more information refer to [33]. For the experiments detailed below, we downsampled the EEG signals to 128 Hz, extracted EEG segments from 0.5 to 2.5 s after the onset of the visual cue, and selected 28 electrodes covering the sensorimotor areas, in accordance to [34].

Dataset-2: known in the literature as BNCI2015001. It is a publicly available dataset[2] by Faller et al. [35]. The EEG data were obtained from 12 able-bodied BCI-naive volunteers (5 female, age 24.8 ± 3.0 years) which participated in at least two MI-BCI sessions (maximum day time frame was 5 days). The EEG recordings were acquired with the g.GAMMAsys active electrode system (electrode positions at FC3, FCz, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CPz and CP4) along with a g.USBamp amplifier (g.tec, Guger Technologies OEG, Graz, Austria) at a sampling frequency of 512 Hz. During acquisition, the signals were bandpass filtered between 0.5 and 100 Hz with an additional notch filter at 50 Hz. In each session, the participants performed 5 runs of 40 trials (i.e. 200 trials) of hand MI vs. feet MI. Only the first run of each session was without providing visual feedback. As before, the EEG signals were windowed from 0.5 to 2.5 s after the onset of the visual cue. Signals were band-pass filtered between 0.5 and 40 Hz, and then downsampled to 128 Hz. Although for some participants three BCI sessions were acquired, for fair comparison purposes, only the first two sessions were used for the experiments.

4 Experiments and Results

In order to keep the focus on the transfer learning rather than on the classifier, we used the traditional CSP and LDA frameworks with diagonal loading [15]. Thus, after filtering each EEG trial between 8 and 30 Hz, six spatial filters were used for feature learning with CSP. A subject-specific CSP+LDA model using calibration data was built to analyze the impact of OTDA. Each OTDA alternative was applied at the CSP feature level space, as shown in Fig. 2. Two different online scenarios were tested: i) block-wise adaptation and ii) sample-wise (continuous) adaptation, as described in subsections 4.1 and 4.2, respectively. The simulations as well as the corresponding source codes implemented in this work are publicly available[3]. We used the POT library [28] for learning the regularized discrete transportation plans, the MNE library [36] for implementing CSP and filtering the EEG data, the Scikit-learn library [37] for learning the linear classifier and the open Python code of RPA [11].

4.1 Data drift learned from previous data

As explained in Section 2.5, for this part of the study the new BCI session was divided into runs of 20 trials each. Given the number of trials of each dataset, we ended up having 7 testing runs for Dataset-1 and 9 for Dataset-2. We tested the four OTDA alternatives (FOTDA-S, FOTDA-GL, BOTDA-S and BOTDA-GL) against the two not domain-adaptive methods (SC and SR) as well as the two data alignment methods (RPA and EA). For the OTDA alternatives, to learn the transportation plan at each r^{th} testing run, we used the transportation set $\mathcal{V}_r$ as defined in (8). In order to boost the transport learning process, we selected a small

[2] Available at http://bnci-horizon-2020.eu/database/data-sets

[3] https://github.com/vpeterson/otda-mibci. The provided examples were run over Subject S9 of Database-1.

number of source samples to learn the mapping. Since it has been shown that the proportions of samples between the source and the target distributions influence the transport [38], this source subset consists of $M = N_v$ trials randomly selected from the source dataset. This process was repeated 20 times, and the best subset of samples was selected based on classification accuracy. Within this validation procedure, the regularization parameters were also selected by means of a grid search process ($\lambda, \eta \in \{0.1, 0.5, 1, 2, 5, 10, 20\}$). In order to statistically analyze the differences between the performances yielded by each method, the non-parametric Friedman test and the post-hoc Nemenyi test at level of significance $\alpha = 0.05$ were applied.

For Dataset-1, the overall accuracy across subjects is shown in Table 1. Columns correspond to runs, organized in increasing order, while the last column shows the average performance across runs. Rows are organized in three subgroups: standard, data alignment and OTDA methods. It can be seen that all domain adaptive methods despite EA achieved similar classification performance between each other (p-value > 0.11), whereas all transfer learning method are significantly different than the non-adaptive one (p-value < 0.02). Significant differences where found between EA and all OTDA alternatives (p-value < 0.01). For the first testing run, accuracy improvements of 9.5%, 5.5% and 6.5% as compared to SC, SR and RPA, respectively, can be achieved by means of the free re-training model based on BOTDA. A similar tendency between the full retraining model (SR) and the different domain-adaptive alternatives is observed in most of the testing runs.

Table 2 shows the classification results achieved on average across subjects for Dataset-2. It is organized as Table 1, where rows and columns separate methods and testing runs, respectively. As before, there is not a clear winner between the OTDA alternatives. In addition, note BOTDA-S present competitive classification performance as compared to the RPA and EA method along the different testing runs (p-value > 0.07). Detailed plots of the results as well as the statistics made can be found in the supplementary material.

Table 1: Overall classification results (accuracy in %) yielded by each tested method for Dataset-1 in the block-wise adaptation scenario. Columns correspond to testing runs. Last column aggregates the average across runs.

	$\mathbf{R_1}$	$\mathbf{R_2}$	$\mathbf{R_3}$	$\mathbf{R_4}$	$\mathbf{R_5}$	$\mathbf{R_6}$	$\mathbf{R_7}$	**av**
SC	64.0	65.0	66.5	70.5	71.0	70.0	72.0	68.4
SR	68.0	69.0	71.5	81.0	74.0	75.5	71.5	72.9
RPA	67.5	70.5	71.5	75.5	75.0	68.0	76.5	72.14
EA	77.0	73.5	71.5	75.5	73.0	74.0	70.5	73.6
FOTDA-S	73.5	68.5	70.0	72.5	73.5	71.0	70.5	71.4
FOTDA-GL	71.5	68.5	70.5	73.5	72.5	70.5	70.5	71.1
BOTDA-S	73.0	71.5	71.0	73.0	70.5	72.5	69.0	71.5
BOTDA-GL	73.5	66.5	71.5	74.5	71.5	70.5	69.5	71.1

Table 2: Overall classification results (accuracy in %) yielded by each tested method for Dataset-2 in the block-wise adaptation scenario. Columns correspond to testing runs. Last column aggregates the average across runs.

	$\mathbf{R_1}$	$\mathbf{R_2}$	$\mathbf{R_3}$	$\mathbf{R_4}$	$\mathbf{R_5}$	$\mathbf{R_6}$	$\mathbf{R_7}$	$\mathbf{R_8}$	$\mathbf{R_9}$	**av**
SC	77.9	78.8	74.6	72.5	74.6	75.8	74.6	77.5	74.6	75.6
SR	80.8	81.7	78.8	80.0	79.2	79.6	79.6	85.0	78.8	80.4
RPA	77.5	76.3	72.9	80.8	77.5	79.6	80.0	83.3	79.2	78.6
EA	77.1	75.8	83.3	82.9	78.8	78.8	79.6	79.2	80.0	79.9
FOTDA-S	74.2	80.2	78.3	78.7	75.4	77.5	77.5	77.5	76.7	77.4
FOTDA-GL	77.1	79.6	79.6	77.9	76.25	78.8	78.8	75.8	76.7	77.8
BOTDA-S	78.3	76.3	78.8	77.9	75.4	76.7	80.0	76.7	78.8	77.6
BOTDA-GL	73.8	77.1	77.5	76.7	75.0	76.3	77.9	75.8	75.4	76.2

4.2 Data drift learned from current and prior data

In the previous experiment we considered that the estimated distribution drift of prior runs was similar to the data variation of a posterior set of trials. Although this assumption allows to learn the mapping for online data adaptation, it may not always be true given the high nonstationarity of the EEG signals, and thus continuous drift learning should be performed.

Since for learning the transportation plan a set of trials from the target domain is always needed, as before, we used the first 20 trials of the new session for recalibration purposes. For every testing trial, we learned the mapping by using not only all prior available data, but also the current trial, as defined in (9). Following the same methodology described in Subsection 4.1, the transportation plan was learned by using the corresponding transportation set $\mathcal{V}_i$ and a small subset of M trials of the source dataset. Considering the online time processing restrictions, this small subset of M trials was selected following the same selection process explained in Subsection 4.1 but using only R_0, thus $M = 20$. Here we simulated a continuous online session. At the end of this experiment, the classification accuracy, per each subject and processing method, was evaluated based upon the label assigned to each trial during the testing phase. For completeness we also evaluated the performance of the calibrated classifier with no adaptation (SC), the standard with recalibration (SR) method and the two data-alignment methods (RPA and EA). Note that the information of the indicated mental task is actually used only by SR, RPA and BOTDA-GL. Statistical analyses were performed also here by means the Friedman test and the post-hoc Nemenyi test at level of significance $\alpha = 0.05$.

Tables 3 and 4 show the accuracy reached in the testing phase by each tested method for each subject of Dataset-1 and Dataset-2, respectively. In both tables the average accuracy across subjects (last row denoted as 'av') shows that BOTDA-GL performs better (p-value < 0.05) than the rest of the OTDA alternatives as well as the SC method, reaching mean accuracy levels of 88.85% and 90.23% for Dataset-1 and Dataset-2, respectively. These values correspond to relative improvements of 20% and 14% with respect to the method without adaptation (SC) for each dataset, respectively. Although, no significant differences were found between BOTDA-GL and SR, RPA and EA, for most of the cases BOTDA-GL achieved the highest accuracy values (7/10 for Dataset-1 and 9/12 for Dataset-2). The rest of the OTDA alternatives achieve similar classification results as the standard calibration method (SC).

Table 3: Accuracy reached by each subject for Dataset-1 (S1-S10) for each one of the evaluated methods in the sample-wise adaptation scenario. The average across subjects is aggregated at the last column. Best results appear in bold.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	av
SC	77.86	82.14	48.57	57.86	65.71	87.86	52.14	87.86	72.14	52.14	68.43
SR	85.00	87.86	81.43	77.86	78.57	89.29	81.43	**92.86**	82.14	71.43	82.79
RPA	97.14	89.29	**80.71**	**81.43**	84.29	95.71	87.86	90.71	88.57	87.86	88.357
EA	81.43	84.29	70.00	77.14	77.86	89.29	82.14	**92.86**	80.71	65.71	80.14
FOTDA-S	78.57	82.86	47.86	55.00	70.71	84.29	76.43	82.86	78.57	54.29	71.14
FOTDA-GL	78.57	82.14	47.86	55.00	71.43	85.00	75.71	83.57	78.57	54.29	71.21
BOTDA-S	68.57	78.57	50.00	52.86	73.57	87.86	72.14	88.57	75.71	55.00	70.29
BOTDA-GL	**100.00**	**95.00**	55.71	79.29	**85.00**	**97.14**	**88.57**	91.43	**98.57**	**97.86**	**88.86**

It is also interesting to compare the computational cost associated to running each adaptive method. We ran our experiments on an Intel® Core$^{\text{TM}}$ i7-6700K CPU @ 4.00 GHz × 8 with 56 GB of RAM. For the sake of comparison, the optimal hyper-parameters search was not included for these time measurements. Therefore, Table 5 shows the computational time of adapting each new trial in this sample-wise scenario for the SR, RPA, EA and BOTDA-GL methods. For Dataset-1, our proposed BOTDA-GL method is 9.6, 79.9 and 18.7 times faster than the SR, RPA and EA methods, respectively. Those values are 11.65, 38.04 and 18.6 for Dataset-2. Although here we only show the computational time of the BOTDA-GL method, it is worth mentioning that all the other OTDA alternatives result in similar computational times.

Table 4: Accuracy reached by each subject for Dataset-2 (S1 - S12) for each one of the evaluated methods in the sample-wise adaptation scenario. The average across subjects is aggregated at the last column. Best results appear in bold.

	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**av**
SC	96.67	96.67	90.00	88.33	86.11	71.11	87.22	57.22	71.11	60.56	53.89	48.89	75.65
SR	98.89	97.78	91.67	89.44	92.78	84.44	94.44	68.89	83.89	70.00	73.89	66.67	84.40
RPA	98.33	97.22	92.78	90.00	93.89	**85.56**	**93.33**	84.44	86.67	75.56	79.44	**76.67**	87.83
EA	98.89	97.22	91.11	90.56	88.89	81.67	91.67	68.33	82.78	68.33	69.44	72.78	83.47
FOTDA-S	98.33	96.67	88.89	82.78	62.22	67.78	88.33	60.00	68.33	65.00	63.33	64.44	75.51
FOTDA-GL	99.44	96.67	89.44	85.00	85.56	74.44	90.00	62.22	66.11	65.00	63.33	64.44	78.47
BOTDA-S	98.33	96.11	87.22	88.33	86.67	72.22	86.67	62.78	71.67	65.00	64.44	52.78	77.69
BOTDA-GL	**100.00**	**98.89**	**96.11**	**91.67**	**100.00**	85.00	88.33	**93.33**	**100.00**	**76.11**	**100.00**	53.33	**90.23**

Table 5: Overall computational time (seconds) of the SR, RPA, EA and BOTDA-GL methods for online adaptation.

	Dataset-1	**Dataset-1**
SR	0.326 ± 0.008	0.337 ± 0.015
RPA	2.711 ± 0.557	1.099 ± 0.324
EA	0.634 ± 0.109	0.537 ± 0.089
BOTDA-GL	0.034 ± 0.010	0.029 ± 0.014

5 Discussions

In this work we investigated the use of optimal transport as a transfer learning approach for addressing EEG variability between sessions of MI-BCI, aiming at avoiding classifier retraining. We have provided a complete framework for applying OTDA, taking into account the constraints of rehabilitative MI-BCI applications. Moreover, we have also proposed a new backward alternative which allows for learning the distribution matching from the target domain (new session) to the source domain (calibration session). In addition, two online simulated adaptive scenarios were tested: i) block-wise adaptation and ii) sample-wise adaptation, with the main difference between these two scenarios being the way that the distribution mapping is learned. Finally, we have also explored two regularized OT formulations for learning the transportation plan ((4) and (5)).

In the first part of this work, we assumed that data distribution drift of a block of trials (run) can be estimated using previous available data. The results showed that for most of the cases there are no significant differences between any of the OTDA alternatives and both SR and RPA. It is important to underline here that BOTDA has the advantage of avoiding classifier retraining, and thus, from the machine learning viewpoint, BOTDA presents a more challenging problem than either of the other adaptive strategies. The lack of significant difference between BOTDA and both SR and RPA indicates that similar classification performance can be achieved avoiding classifier retraining when BOTDA is applied. With regard to the EA approach, we observed that it outperforms all other methods, being as good as SR. This method, as well as all the other considered adaptive methods, requires the training of a new classifier before a new testing block starts. Furthermore, in this scenario, we also note that adding the group-lasso penalty to the BOTDA formulation does not contribute to increasing classification performance. This can be explained by the fact that without classifier retraining, BOTDA-GL only relies on the optimal transportation plan learned with label information from a set of previous trials different from the current testing run. Finally, the fact that after a certain point, domain adaptation does not contribute towards classification improvement can be explained by the "saturation" effect in transfer learning [39], which establishes that when a sufficient amount of data from the target domain is available, a good classifier can already be trained without transfer learning. However, for MI-BCIs applications in motor rehabilitation, therapy session time is typically limited. Thus, having a machine-learning solution for avoiding long calibration times before each session is of utmost importance. In fact, for such practical application, it could allow dedicating more therapy time to provide the biofeedback to the patient rather than to retrain the decoding model, where no feedback can be provided. The improvements found by using BOTDA at the first testing runs without classifier retraining not only mitigates the well-known decrease in classification performance at the beginning of a new session, but it also constitutes a step forward in the search for more efficient and robust methodologies for applying BCIs to neurorehabilitation.

In our second experiment we analyzed a way to implement OTDA in online sequential scenarios where labeled data are available after every experimental trial. The classification results shown in tables 3 and 4 clearly indicate the advantages of using the current trial in the BOTDA-GL configuration, whereas for the other tested alternatives of OTDA, the incorporation of the current trial to the transportation set does not improve classification performance. Interestingly, although in this context both SR and RPA strategies use the indicated mental task of the current trial to make the adaptation and learn the decoding model, overall, it is better to use the classifier already trained in the source domain and transport the current trial by means of BOTDA-GL. Additionally, we note here that there are few cases in which classification performance is not improved by BOTDA-GL. These cases are in fact shedding light on the idea that using the indicated mental task (known by the system) does not always represent the mental task truly performed by the user. Thus, only good classification results will be achieved if the patterns provided by the user are discriminative enough for the corresponding mental task. In the case of stroke rehabilitation, patients may exhibit mental fatigue after repetitively performing the intended mental tasks [40]. In such situation the use of BOTDA could be a valuable solution to cope with the lack of ability of stroke survivors to maintain the same mental state throughout the whole BCI session.

Approaches that use the current label information for adaptive BCI have already been evaluated [16, 41], in which the testing class information was used to update the weights of a linear classifier. When BOTDA is being used, the classifier trained with data from the calibration session is kept fixed. Given that BOTDA-GL transformed testing data based on label information, i.e. penalizing the learned mapping so as to transport samples of the same class together, this approach will fail if the features do not match with the indicated mental task. This could explain the reduced decoding performance observed in some subjects (S3 an S4 in Dataset-1, and S7 and S12 in Dataset-2). On the contrary, although RPA also makes use of the indicated mental task, the classifier is re-trained, and thus this issue becomes less evident. As discussed by the authors in [42], if a user is not able to produce stable and distinctive EEG patterns, then the algorithm fails in matching the target sample to its supposed label class. In this direction, the classification reached by each subject in the new session could be interpreted as a quantification of how well the subject performed the indicated mental task. Although more research is needed, we believe that this supervised transfer learning based on our proposed backward OT could be a valuable alternative for avoiding the current pitfalls on how the user is trained and how the feedback is presented to the user during MI-BCI [43].

The computational time required for adapting each trial in the sample-wise scenario revealed that BOTDA is about 10 times faster than the fastest conventional method tested (SR). This result is important if we extrapolate the associated computational time of more complex decoding algorithms. The low computational time can be explained by two reasons: i) the decoding model is kept fixed, without any retraining needs, and ii) once the transportation plan is learned, the sample transportation (see Eq.(6)) consists in simple matrix multiplications [22]. The fact that by means of BOTDA the adaption is less time-consuming than re-training a simple decoding model reveals a huge advantage when considering real-time decoding, where the feedback presenting time should be kept as short as possible [44, 45]. Such an advantage, together with the ability to transport discriminative trials and boost the classification performance of an already trained decoding model, makes BOTDA-GL a promising, effective and efficient method for online applications.

In the last years the use of TL based on deep learning to tackle the variability in BCI has gained more relevance. Although different challenges exist for using deep learning methods to successfully train models in neuroscience, recent works have shown that deep learning architectures can be used for such applications [46–48]. However, most of the existing works aim to address the cross-subject variability and not the cross-session variability, the former being easier to implement since the cost of producing sufficient high-quality data and annotations is lower with respect to multi-subjects multi-sessions datasets. Nevertheless, given the rapid advance in the deep learning community, future research directions should explore the use of such techniques.

In this work, we have presented evidence that OT can be used for TL in MI-BCI in the context of a motor rehabilitation application. When data becomes available in blocks (i.e., group of multiple trials), we showed that classification performance can be improved for the first testing runs. For such scenarios, we suggest to use the BOTDA-S formulation. On the contrary, when data is available after each experimental trial, together with the indicated mental task information, BOTDA-GL should be preferred. Previous works have also shown the impact of using OT for domain adaptation in BCIs. In particular, the work in [23] focused on the use of OTDA for multi-subject P300 based-BCIs. Although the beforementioned work described a simulated online scenario, our approach cannot be directly compared, since, as opposed to our work, all available target data was used to learn the transport. Here, we have provided a workflow for using OTDA in real-life MI-BCI applications. Additionally, the authors in [11] introduce the RPA method and compared its performance against OTDA, applied in the Riemannian manifold. Unfortunately, there is no enough information on how the method is modified and implemented for running it in such domains. In regard to this, an interesting paper showing how to use OTDA in the Riemannian manifold has recently appeared [49]. Future work should explore further analyses in this direction.

Although we have focused on synchronized MI-BCI for motor rehabilitation, the proposed BOTDA method is not restricted to such applications. For asynchronous BCI, where the user himself/herself chooses which task he/she wants to do and when to do it, the block-wise adaptation scenario can still be used by relying on a short calibration block of few trials to learn the transportation map. For such asynchronous BCI applications, BOTDA-S can still be used without any limitation. This method yields comparable classification results as compared to BOTDA-GL in the block-wise adaptation. Moreover, since BOTDA is applied at the level of the feature space, the workflow presented here can be easily adapted for any other BCI paradigm with either two or more classes.

6 Conclusions

In this paper we described and studied different strategies based on optimal transport for tackling cross-session variability in MI-BCI. Our results show that when data from the new BCI session is available in blocks (runs), classification accuracy can be increased by up to 9.5% by BOTDA as compared to the traditional recalibration approach, with only 20 trials of the new session required for the adaptation. In addition, when considering the current trial with its mental task information within the OT learning procedure, we found that BOTDA-GL improves up to perfect classification when the user is able to produce discriminative brain patterns, trimming down the computational time to only 30 ms. These findings indicate that BOTDA-GL is a promising approach for providing accurate and rapid feedback in cue-based environments. Further research is needed to demonstrate the applicability of BOTDA-GL as a user performance metric, which is part of our future research plans.

The use of transfer learning for tackling the EEG nonstationarity is gaining more and more attention within the BCI community. Methods based on data alignment and deep learning are at the top of the list. In the future, we plan to combine the advantages of these transfer learning strategies to try to further improve classifier robustness. In addition we plan to conduct multiple-session MI-BCI experiments for further validating the proposed strategies in real online scenarios.

Acknowledgments

This work was supported in part by the Consejo Nacional de Investigaciones Científicas y Técnicas, CONICET, through PUE-IMAL No. 22920180100041-CO, Universidad Nacional del Litoral, UNL, through CAI+D-UNL 2016 PIC No. 50420150100036LI, and by the ETH Zurich Foundation in collaboration with Hocoma AG

References

[1] K. K. Ang, C. Guan, K. S. G. Chua, B. T. Ang, C. W. K. Kuah, C. Wang, K. S. Phua, Z. Y. Chin, and H. Zhang, “A large clinical study on the ability of stroke patients to use an EEG-based motor imagery brain-computer interface,” *Clinical EEG*

and Neuroscience, vol. 42, no. 4, pp. 253–258, 2011.

[2] L. Van Dokkum, T. Ward, and I. Laffont, "Brain computer interfaces for neurorehabilitation–its current status as a rehabilitation strategy post-stroke," *Annals of Physical and Rehabilitation Medicine*, vol. 58, no. 1, pp. 3–8, 2015.

[3] I. Lazarou, S. Nikolopoulos, P. C. Petrantonakis, I. Kompatsiaris, and M. Tsolaki, "EEG-based brain–computer interfaces for communication and rehabilitation of people with motor impairment: a novel approach of the 21st century," *Frontiers in Human Neuroscience*, vol. 12, p. 14, 2018.

[4] R. Mane, T. Chouhan, and C. Guan, "BCI for stroke rehabilitation: motor and beyond," *Journal of Neural Engineering*, vol. 17, no. 4, p. 041001, 2020.

[5] A. Vourvopoulos, O. M. Pardo, S. Lefebvre, M. Neureither, D. Saldana, E. Jahng, and S.-L. Liew, "Effects of a brain-computer interface with virtual reality (VR) neurofeedback: A pilot study in chronic stroke patients," *Frontiers in Human Neuroscience*, vol. 13, p. 210, 2019.

[6] K. K. Ang, K. S. G. Chua, K. S. Phua, C. Wang, Z. Y. Chin, C. W. K. Kuah, W. Low, and C. Guan, "A randomized controlled trial of EEG-based motor imagery brain-computer interface robotic rehabilitation for stroke," *Clinical EEG and Neuroscience*, vol. 46, no. 4, pp. 310–320, 2015.

[7] D. Rathee, A. Chowdhury, Y. K. Meena, A. Dutta, S. McDonough, and G. Prasad, "Brain–machine interface-driven post-stroke upper-limb functional recovery correlates with beta-band mediated cortical networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 5, pp. 1020–1031, 2019.

[8] F. Lotte, L. Bougrain, and M. Clerc, "Electroencephalography (EEG)-based brain–computer interfaces," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–20, 1999.

[9] W. Samek, M. Kawanabe, and K.-R. Müller, "Divergence-based framework for common spatial patterns algorithms," *IEEE Reviews in Biomedical Engineering*, vol. 7, pp. 50–72, 2013.

[10] P. Gaur, K. McCreadie, R. B. Pachori, H. Wang, and G. Prasad, "Tangent space features-based transfer learning classification model for two-class motor imagery brain–computer interface," *International journal of neural systems*, vol. 29, no. 10, p. 1950025, 2019.

[11] P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Riemannian procrustes analysis: Transfer learning for Brain–Computer Interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 8, pp. 2390–2401, 2018.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[13] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 441–446, 2000.

[14] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Muller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41–56, 2007.

[15] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 2, pp. 355–362, 2010.

[16] P. Shenoy, M. Krauledat, B. Blankertz, R. P. Rao, and K.-R. Müller, "Towards adaptive classification for BCI," *Journal of Neural Engineering*, vol. 3, no. 1, p. R13, 2006.

[17] X. Li, C. Guan, K. K. Ang, H. Zhang, and S. H. Ong, "Spatial filter adaptation based on the divergence framework for motor imagery EEG classification," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 1847–1850.

[18] W. M. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," *arXiv preprint arXiv:1812.11806*, 2018.

[19] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, "Transfer Learning: A Riemannian Geometry Framework with Applications to Brain-Computer Interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 5, pp. 1107–1116, 2018.

[20] H. He and D. Wu, "Transfer Learning for Brain-Computer Interfaces: A Euclidean Space Data Alignment Approach," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 2, pp. 399–410, 2020.

[21] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338.

[22] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, 2017.

[23] N. T. Gayraud, A. Rakotomamonjy, and M. Clerc, "Optimal transport applied to transfer learning for P300 detection," in *BCI 2017 - 7th Graz Brain-Computer Interface Conference, Sep 2017, Graz, Austria*, 2017, p. 6.

[24] L. V. Kantorovich, "On the translocation of masses," *Dokl. Akad. Nauk SSSR, 37*, vol. 37, no. 7-8, pp. 227–229, 1942.

[25] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems*, 2013, pp. 2292–2300.

[26] P. A. Knight, "The Sinkhorn–Knopp algorithm: convergence and applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 261–275, 2008.

[27] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[28] R. Flamary and N. Courty, “POT python optimal transport library,” 2017. [Online]. Available: https://github.com/rflamary/POT

[29] J. Wolpaw and E. W. Wolpaw, *Brain-computer interfaces: principles and practice*. OUP USA, 2012.

[30] F. Lotte, “Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces,” *Proceedings of the IEEE*, vol. 103, no. 6, pp. 871–890, 2015.

[31] K. K. Ang and C. Guan, “EEG-based strategies to detect motor imagery for control and rehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 4, pp. 392–401, 2016.

[32] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, “Multiclass brain–computer interface classification by Riemannian geometry,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, 2011.

[33] V. Peterson, D. Wyser, O. Lambercy, R. Spies, and R. Gassert, “A penalized time-frequency band feature selection and classification procedure for improved motor intention decoding in multichannel EEG,” *Journal of Neural Engineering*, vol. 16, no. 1, p. 016019, 2019.

[34] S. Marchesotti, R. Martuzzi, A. Schurger, M. L. Blefari, J. R. del Millán, H. Bleuler, and O. Blanke, “Cortical and subcortical mechanisms of brain-machine interfaces,” *Human brain mapping*, vol. 38, no. 6, pp. 2971–2989, 2017.

[35] J. Faller, C. Vidaurre, T. Solis-Escalante, C. Neuper, and R. Scherer, “Autocalibration and recurrent adaptation: Towards a plug and play online ERD-BCI,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 3, pp. 313–319, 2012.

[36] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen *et al.*, “MEG and EEG data analysis with MNE-Python,” *Frontiers in Neuroscience*, vol. 7, p. 267, 2013.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[38] N. Courty, R. Flamary, and D. Tuia, “Domain adaptation with regularized optimal transport,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 274–289.

[39] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Machine Learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[40] R. Foong, N. Tang, E. Chew, K. S. G. Chua, K. K. Ang, C. Quek, C. Guan, K. S. Phua, C. W. K. Kuah, V. A. Deshmukh, L. H. L. Yam, and D. K. Rajeswaran, “Assessment of the Efficacy of EEG-Based MI-BCI with Visual Feedback and EEG Correlates of Mental Fatigue for Upper-Limb Stroke Rehabilitation,” *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 3, pp. 786–795, 2020.

[41] C. Vidaurre, M. Kawanabe, P. von Bünau, B. Blankertz, and K.-R. Müller, “Toward unsupervised adaptation of LDA for brain–computer interfaces,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 587–597, 2010.

[42] F. Lotte, F. Larrue, and C. Mühl, “Flaws in current human training protocols for spontaneous brain-computer interfaces: lessons learned from instructional design,” *Frontiers in human neuroscience*, vol. 7, p. 568, 2013.

[43] F. Lotte, C. Jeunet, B. Mladenovi, Jelena abd NKaoua, and L. Pillette, *A BCI challenge for the signal processing community: considering the user in the loop*. IET, 2018, pp. 1–33.

[44] E. F. Oblak, J. A. Lewis-Peacock, and J. S. Sulzer, “Self-regulation strategy, feedback timing and hemodynamic properties modulate learning in a simulated fmri neurofeedback environment,” *PLoS computational biology*, vol. 13, no. 7, p. e1005681, 2017.

[45] A. Belinskaya, N. Smetanin, M. Lebedev, and A. Ossadtchi, “Short-delay neurofeedback facilitates training of the parietal alpha rhythm,” *Journal of Neural Engineering*, 2020.

[46] D. Wu, Y. Xu, and B. Lu, “Transfer learning for eeg-based brain-computer interfaces: A review of progresses since 2016,” *arXiv preprint arXiv:2004.06286*, 2020.

[47] N. A. Alzahab, L. Apollonio, A. Di Iorio, M. Alshalak, S. Iarlori, F. Ferracuti, A. Monteriù, and C. Porcaro, “Hybrid Deep Learning (hDL)-Based Brain-Computer Interface (BCI) Systems: A Systematic Review,” *Brain Sciences*, vol. 11, no. 1, p. 75, 2021.

[48] S. Roy, A. Chowdhury, K. McCreadie, and G. Prasad, “Deep learning based inter-subject continuous decoding of motor imagery for practical brain-computer interfaces,” *Frontiers in Neuroscience*, vol. 14, 2020.

[49] O. Yair, F. Dietrich, R. Talmon, and I. G. Kevrekidis, “Domain Adaptation with Optimal Transport on the Manifold of SPD matrices ,” *arXiv*, 2019.