Transfer Learning for P300 Brain-Computer Interfaces by Joint Alignment of Feature Vectors

Fatih Altindis, Antara Banerjee, Ronald Phlypo, Bulent Yilmaz, Marco Congedo

Transfer Learning for P300 Brain-Computer Interfaces by Joint Alignment of Feature Vectors

Fatih Altindis, Antara Banerjee, Ronald Phlypo, Bulent Yilmaz and Marco Congedo

***Abstract*— This paper presents a new transfer learning method named group learning, that jointly aligns multiple domains (many-to-many) and an extension named fast alignment that aligns any further domain to previously aligned group of domains (many-to-one). The proposed group alignment algorithm (GALIA) is evaluated on brain-computer interface (BCI) data and optimal hyper-parameter values of the algorithm are studied for classification performance and computational cost. Six publicly available P300 databases comprising 333 sessions from 177 subjects are used. As compared to the conventional subject-specific train/test pipeline, both group learning and fast alignment significantly improve the classification accuracy except for the database with clinical subjects (average improvement: 2.12±1.88%). GALIA utilizes cyclic approximate joint diagonalization (AJD) to find a set of linear transformations, one for each domain, jointly aligning the feature vectors of all domains. Group learning achieves a many-to-many transfer learning without compromising the classification performance on non-clinical BCI data. Fast alignment further extends the group learning for any unseen domains, allowing a many-to-one transfer learning with the same properties. The former method creates a single machine learning model using data from previous subjects and/or sessions, whereas the latter exploits the trained model for an unseen domain requiring no further training of the classifier.**

***Index Terms*— Brain-computer interface (BCI), transfer learning, domain adaptation, Riemannian geometry, electroencephalography (EEG).**

I. INTRODUCTION

A BRAIN-Computer Interface (BCI) is a system capable of predicting or classifying cognitive states and intentions of the user through the analysis of neurophysiological signals. BCI systems based on electroencephalography (EEG) have a wide range of applications thanks to the fact that EEG is completely non-invasive, features high temporal resolution, suits mobile usage and requires inexpensive equipment [1], [2]. Over the past 30 years EEG-based BCIs, which are the concern of the present article, have been showcased in a number of disparate proof-of-concepts, including wheelchair and prosthetic control, moving a cursor on a screen, spellers, gaming and artistic expression [3]–[5]. Still, current systems are far from being production-ready for a number of reasons, among which the most fundamental are probably the difficulty in reaching a 'plug & play' mode of operation and the insufficient accuracy currently achieved on single-trial decoding.

Traditionally, BCI operates in two phases: in the *training phase* a machine learning model (MLM) is calibrated in a supervised manner (with labeled EEG data), while in the *test phase*, *i.e.*, the actual use, the BCI must classify EEG data to infer the classes they belong to (unsupervised mode of operation). Generally, BCIs require a calibration session before every use, because EEG data is highly variable, thus a pre-trained MLM struggles in the subsequent sessions of the same user, let alone in other subjects.

The necessity of frequent calibrations is a cumbersome ritual for potential users, causing a waste of time and energy [6]. This is particularly blocking for the clinical population, whose cognitive resources are limited [6]. In view of this situation, the research community is currently focusing on so-called *transfer learning* methods, which aim at MLMs capable of overcoming the cross-session and cross-subject dependency [7]–[14], [16]–[21], [23]–[30]. In this field of research, the domain we ought to use for learning is referred to as the *source* and the domain we want to apply the learning to as the *target*. If the transfer is operated without using any label from the target domain, it is called *unsupervised*. If some labels are used, it is called *semi-supervised*. In this article we address the latter scenario.

This work has been partially supported by ANR grant Hifi (ANT-20-CE17-0023) and TÜBİTAK-2214A grant (1059B142100364). (*Corresponding author: Fatih Altindis).*

Fatih Altindis is with the Electrical & Electronics Engineering Department, Abdullah Gül University, Kayseri, 38080 Türkiye (e-mail: fatih.altindis@agu.edu.tr).

Antara Banerjee is with the Department of Electrical Engineering, Indian Institute of Technology, Varansi, India (e-mail: antara.banerjee.cd.eee19@ithbu.ac.in).

Ronald Phlypo is with the GIPSA-Lab, University Grenoble-Alpes, CNRS, Grenoble INP, Grenoble, 38400 France (e-mail: ronald.phlypo@gipsa-lab.grenoble-inp.fr).

Bulent Yilmaz is with the Electrical & Electronics Engineering Department, Abdullah Gül University, Kayseri, 38080 Türkiye and also with the Electrical Engineering Department, Gulf University for Science and Technology, Hawally, 32093, Kuwait (e-mail: bulent.yilmaz@agu.edu.tr).

Marco Congedo is with the GIPSA-Lab, University Grenoble-Alpes, CNRS, Grenoble INP, Grenoble, 38400 France (e-mail: marco.congedo@gipsa-lab.grenoble-inp.fr).

Two transfer learning approaches can be found in the literature: *rule adaptation* and *domain adaptation* [7]. Rule adaptation tries to reduce the calibration time for the new tasks by adapting classification rule. This is particularly useful when the available data is scarce or the labeling is insufficient. The current trend following this approach consists in fine-tuning, for the target data at hand, deep neural networks (DNN) pre-trained on source domains [8]–[10]. The main limitations are the computational cost and the fact that DNNs act as black boxes, thus it is in general hard to understand what they actually learn [11]. Although recent studies on interpretable neural networks have started to unveil this ambiguity, the research on neural networks is still in an early phase [12], [13]. Besides, training a DNN model requires a very large amount of data, a requirement that is still difficult to meet with EEG [14].

The other approach, *domain adaptation*, tries to 'shift' the source and target domain toward a *homogeneous space* so as to minimize the discrepancy between the two. In the literature we find this approach operating either from one source domain to one target domain (one-to-one), or from several source domains to one target domain (many-to-one) (Fig. 1). Early attempts have adapted the well-known common spatial pattern (CSP) filter [15], [16]. For one-to-one transfer learning, penalty terms have been introduced to regularize the CSP objective function [17], [18]. One-to-one transfer learning in regularized CSP (RCSP) can be expanded to the many-to-one setting by either taking a weighted average of multiple source data to form a composite spatial covariance matrix or by aggregating RCSP coefficients from each source iteratively [19], [20]. In [21] the authors whitened the arithmetic average of the covariance matrices estimated on the EEG trials of each subject. This amounts to a Euclidean recentering of the covariance matrices around the identity matrix.

A peculiar line of research has arisen thanks to the utilization of Riemannian geometry [22]. The first attempt has recentered the covariance matrices estimated on the EEG trials of each subject using parallel-transport on the Riemannian manifold of symmetric positive-definite (SPD) matrices [23]. The method, intrinsically one-to-one, can be easily coupled with spatial filtering [24]. In a subsequent attempt, a *stretching* and a *rotation* steps have been added to the recentering step, yielding a more precise one-to-one transfer learning method named Riemannian Procrustes Analysis (RPA) [25]. The stretching step equalizes the Riemannian dispersion of the observed source and target points (*i.e.*, covariance matrices) around the identity, which after recentering is the barycenter of both domains. Like recentering, the stretching steps is unsupervised. The rotation step, which instead is semi-supervised, tries to align as much as possible the intra-class barycenters of the two domains. In [26], recentering on the manifold is followed by lifting onto the tangent space of the SPD manifold. The tangent vectors are then stretched and rotated as in the RPA, effectively aligning the tangent vectors of the source and target domain. This method, named tangent space alignment (TSA), also boils down to a one-to-one Procrustes procedure. However, as it acts on the tangent space of the SPD manifold, it is computationally simpler and faster.

A more recent trend in the domain adaptation literature is the many-to-one strategies. In [27], a one-to-one domain adaptation

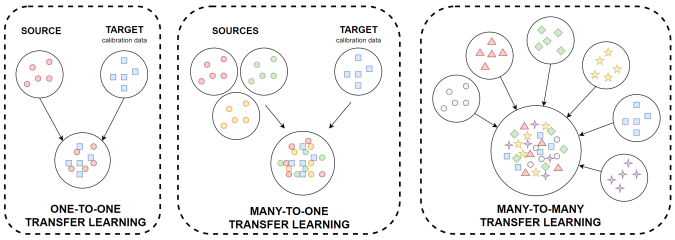

Fig. 1. Schematic representation of different transfer learning strategies for domain adaptation. Circles represents domains, in general, data of a particular subject or a particular session. The goal is shifting the source and target domains in a homogeneous space. From left to right: the one-to-one, many-to-one and many-to-many strategy. See text for details.

on the Riemannian manifold is followed by training one classifier for each of several source domains. Then, a voting algorithm is employed to combine the prediction of each classifier for the classification of target data. The authors in [28] proposed to first select source domains (*e.g.*, subjects) based on their one-to-one transfer learning classification performance. Then, the distribution of the tangent space features of selected sources are aligned using the maximum mean discrepancy criterion to train a single classifier. Inspired from the composite CSP, in [29] a regularization term is introduced to estimate weighted Riemannian means for each class using multiple source domains. The transfer learning methods described so far are in general very much *source-dependent,* that is to say, a careful selection of the source domain(s) is required in order to avoid *negative learning* [20], [24], [27]–[29].

Generalizing the Procrustes cost function of TSA [26], we have recently ended up taking a route rather different from all previous domain adaptation attempts: in [30], *all available domains* are first jointly aligned in the tangent space of the Riemannian SPD manifold, a *group MLM* is then estimated and finally this model is applied without further tuning to test all (aligned) domains. This way, we have effectively achieved a *many-to-many* transfer learning method, which we have named *group learning* (Fig. 1). To the best of our knowledge the group learning strategy in [30] is the first one jointly aligning many domains to train a single classifier for all data. In the present article we refine and push further this idea. We show here how to align any further (target) domain to the group MLM without retraining or tuning the classifier at all (*fast alignment*). Accordingly, numerous databases can be used to build a massive group MLM that can be applied *as it is* in a semi-supervised fashion on any target data. This effectively allows a "many-to-one" transfer learning strategy, which is what is useful in practice. We show that group MLMs are little sensitive to the inclusion of low-scoring subjects, *i.e.*, that the method is not source-dependent. We also show that choosing optimal hyper-parameters for the algorithm is not cumbersome. As compared to the formulation in [30], we also improve the convergence rate of the gradient descent optimization introducing a smart initialization.

We test the proposed method on six P300 databases including 333 sessions recoded on 177 subjects. We demonstrate cross-subject and cross-session group transfer learning. In contrast to the state-of-the-art subject-specific learning, the proposed group transfer learning significantly improves the accuracy (2.12±1.88% average improvement) in five out of the six BCI

databases that we considered (the non-clinical ones), that is, it features *positive learning* for non-clinical data. The runtime of GALIA on a regular PC we have observed is below 10 s for an 8-subject database and up to 1000 s for a 126-subject database. The complete Julia computer programming code to replicate our analysis is available in a GitHub repository [31].

The reminder of the paper is organized as it follows: The Materials and Methods section summarizes useful elements of Riemannian geometry. Then it describes the proposed group learning algorithm, the fast alignment method, the processing pipeline and the BCI databases we have used. The Results section reports the accuracy of the proposed group learning algorithm under different hyper-parameters settings as compared to subject-specific train-test accuracy. The Discussion section considers advantages and limitations of this study. It also points to several possible directions for future research on group learning.

II. Methods and Materials

Hereby we present the group learning method for the case of BCIs based on event-related potentials (ERPs). However, the method is very general in that it may apply to any collection of feature vectors (not just BCI data).

Throughout this paper we denote matrices by upper case bold letters (**A**), vectors by lower case bold letters (**a**), variables by lower case italic letters (a) and constants by upper case italic letters (A). Set of objects are denoted with curly brackets such as, $n \in \{1,\ldots,N\}$. The matrix operators $(\cdot)^T$, $\mathrm{tr}(\cdot)$, $(\cdot)^{-1}$, $(\cdot)^{-1/2}$, $\log()$, $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the transpose, trace, inverse, inverse of the principal square root, matrix logarithm, Frobenius norm and 2-norm of the argument, respectively. The matrix operators *off*$(\cdot)$ and *uvec*$(\cdot)$ nullify the diagonal elements and vectorize the upper triangle of the argument, respectively. The symbol $\circ$ indicates the Hadamard product. The $N \times N$ identity matrix is denoted by $\mathbf{I}_N$.

A. ERP-based BCI data

In ERP-based BCIs a continuous stream of discrete sensory stimuli (symbols) is flashed in turn on a screen. The user is allowed to select symbols one-by-one by just focusing on them. In P300 spellers, the symbols are a subset of those found on a computer keyboard. The symbol the user focuses on for a given selection is named *salient,* whereas all other symbols are named *non-salient*. Upon flashing, the symbols evoke stereotypical electrical potential in the brain, lasting up to 1 second [32]. EEG records are segmented into 1-s epochs, named *trials*, starting at the exact moment of the flash. The aim of the BCI is to determine what stimulus is salient at a given time, that is, what symbol the user wish to select, using only the ERP responses of the user. Hence, given a sequence of acquired ERPs, a binary class problem (salient vs. non-salient) is posed.

B. Preprocessing

EEG signals are filtered by a second-order Butterworth filter (1-16 Hz) featuring linear phase response. Trials are extracted by segmenting EEG data into 1-s epochs. Those featuring excessive artefacts are excluded from the analysis by applying a data-driven amplitude-thresholding procedure. No further artefact rejection nor artefact correction procedure has been used so as to mimic the real-world usage of BCIs.

C. Encoding ERP Trials

In order to encode ERP trial we follow the well-established Riemannian approach [24], [26], [30], [33]–[35]. Let $m \in \{1,\ldots,M\}$ be the index of M domains (*i.e.*, subjects and/or sessions) and $l \in \{1,\ldots,L_m\}$ be the index of L_m available ERP trials for the m^{th} domain. Therefore, let $\mathbf{X}_{ml} \in \mathbb{R}^{N_m \times T_m}$ be the EEG data of a trial, with N_m and T_m the number of channels and the number of time samples for the m^{th} domain, respectively. First, for each domain a *prototype trial* (mean) for *salient* ERP trials is estimated by using the weighted least-square estimation of the salient ERP detailed in [32]. We retain the first $D=N_m/2$ principal components, denoted as $\mathbf{Y}_m \in \mathbb{R}^{D \times T_m}$, of the prototype trial of each domain. Then, the so-called *super-trials* [35] are created as $\mathbf{X}'_{ml} = [\,\mathbf{X}^T_{ml}\ \mathbf{Y}^T_m]^T$.

Next, the covariance matrices of all trials are estimated using the linear Ledoit and Wolf shrinkage estimator [36]. The form of the covariance matrices of super-trials reads

$$\mathbf{X}'_{ml}\mathbf{X}'^T_{ml} = \begin{bmatrix} \mathbf{X}_{ml}\mathbf{X}^T_{ml} & \mathbf{X}_{ml}\mathbf{Y}^T_m \\ \mathbf{Y}_m\mathbf{X}^T_{ml} & \mathbf{Y}_m\mathbf{Y}^T_m \end{bmatrix}, \tag{1}$$

from which we see that the second diagonal block, $\mathbf{Y}_m\mathbf{Y}^T_m$, is the same for all trials of the given domain and therefore does not hold relevant discriminant information.

Following [23], [34], all regularized covariance matrices for a given domain, denoted $\mathbf{C}_{ml}$, are recentered around the identity using parallel transport and lifted onto the tangent space therein [33], such as

$$\mathbf{S}_{ml} = \log\left(\mathbf{G}_m^{-1/2}\mathbf{C}_{ml}\mathbf{G}_m^{-1/2}\right), \tag{2}$$

where $\mathbf{G}_m$ is the Riemannian mean of the set $\{\mathbf{C}_{m1},\ldots,\mathbf{C}_{mL}\}$, estimated using the gradient descent algorithm GM-GD given in [37].

D. Feature Vectors and Surrogate Feature Vectors

In order to obtain feature vectors, the symmetric matrices $\mathbf{S}_{ml}$ are vectorized such as

$$\mathbf{v}'_{ml} = uvec\left(\mathbf{S}_{ml} \circ \mathbf{H}\right),$$

where $\mathbf{H}$ is a matrix holding 1 on the diagonal elements and $\sqrt{2}$ elsewhere. The normalization enforced by $\mathbf{H}$ ensures that the 2-norm of feature vectors $\mathbf{v}'_{ml}$ is equal to the Frobenius norm of $\mathbf{S}_{ml}$ [33], that is, they are such that $\|\mathbf{v}'_{ml}\|_2 = \|\mathbf{S}_{ml}\|_F$.

In order to equalize the norm of the tangent vectors across domains, all tangent vectors are normalized so as to have unit mean norm for each domain, that is

$$\mathbf{v}_{ml} = \mathbf{v}'_{ml}\left(\frac{1}{L_m}\sum_{l=1}^{L_m}\left\|\mathbf{v}'_{ml}\right\|_2\right)^{-1}.$$

Furthermore, since the $\mathbf{Y}_m\mathbf{Y}^T_m$ block in (1) do not hold any discriminant information, its elements are removed from $\mathbf{v}_{ml}$, after the vectorization of $\mathbf{S}_{ml}$. Hence, the number of elements of vectors $\mathbf{v}_{ml}$ is $E_m=(N_m^2+N_m+2DN_m)/2$. Notice that while it is not clear whether the removed elements pertain only to the $\mathbf{Y}_m\mathbf{Y}^T_m$ block after non-linear transformation (2), our tests indicate that keeping these elements does not improve the classification performance.

As for the group learning method, we propose to align these feature vectors, whereas the alignment operation itself is

estimated on *surrogate feature vectors*. These are obtained for each domain separately by bootstrapping average estimations of $\mathbf{v}_{ml}$ for each class separately, and then normalize them so as to have a global unitary mean norm within each domain. In this work, such bootstraps are obtained for each class by averaging A vectors $\mathbf{v}_{ml}$ randomly drawn (with replacement) from the same class. E_m bootstraps are stacked horizontally to form matrices $\mathbf{T}_{mk}\in\mathbb{R}^{E_m \text{ x } E_m}$, for each domain $m\in\{1,\ldots,M\}$ and each classes $k\in\{1,\ldots,K\}$, separately. Surrogate feature vectors are used to ensure that the number of vectors is the same within the same class for all domains. Since we are drawing the same number of bootstraps for all classes, in this work the dimension of the $\mathbf{T}_{mk}$ matrices is equal across classes, but this is not a requirement of the group learning algorithm. Notice also that bootstrapping is not a hardcore necessity to form surrogate feature vectors $\mathbf{T}_{mk}$; other methods to generate them from feature vectors $\mathbf{v}_{ml}$ may be used.

The bootstraps are then whitened by finding M matrices $\{\mathbf{W}_1,\ldots,\mathbf{W}_M\}$ such that

$$\mathbf{W}_m^T\left[\sum_k\left(\mathbf{T}_{mk}\mathbf{T}_{mk}^T\right)\right]\mathbf{W}_m=\mathbf{I}_P,\ \text{for all } m\in\{1,\ldots,M\},$$

where P is the pre-whitening dimension, chosen to be equal for all domains. Finally, we compute pre-whitened cross outer products

$$\mathbf{R}_{ijk}=\mathbf{W}_i^T\mathbf{T}_{ik}\mathbf{T}_{jk}^T\mathbf{W}_j,\ \forall i\neq j\in\{1,\ldots,M\},k\in\{1,\ldots,K\}.$$

E. Group Learning

The general idea to achieve group learning is to find M linear transformation $\{\mathbf{U}_1,\ldots,\mathbf{U}_M\}$ such that all full-rank matrices $\mathbf{U}^T_i\mathbf{R}_{ijk}\mathbf{U}_j$ are as diagonal as possible under appropriate constraints on $\{\mathbf{U}_1,\ldots,\mathbf{U}_M\}$. For M=2 with a single class (K=*1*) and under the constraint of orthogonality, $\mathbf{U}_1$ and $\mathbf{U}_2$ are given in deterministic form as the matrices holding in columns the left and right singular vectors of $\mathbf{R}_{12}$. This is known as *maximum covariance analysis* (MCA) [38], the normalized version of which is the better-known *canonical correlation analysis*. The MCA is the essence of the Tangent Space Alignment (TSA) method [26]. Thus, our proposition can be understood as a generalization of TSA to M>2 subjects.

Group learning can indeed be seen as a generalization of MCA to the much more general (and useful) case (M>2, K>1), which is the case of concern here. Another way to see it is as a special case of the *joint blind source separation* problem, which allows us to readily exploit the extensive treatment this problem has received by the signal processing community (*e.g.*, [39]–[42]). In particular, in this work we adopt the cyclic *approximate joint diagonalization* (AJD) gradient descend optimization scheme proposed in [41]. Once found the $\mathbf{U}_m$ matrices, the alignment matrices $\mathbf{B}_m$ are obtained as $\mathbf{B}_m$=$\mathbf{W}_m\mathbf{U}_m$ for all $m\in\{1,\ldots,M\}$. Then, the feature vectors of each domain, regardless of their class, are *aligned* by means of the transformation

$$\mathbf{z}_{ml}\leftarrow\mathbf{B}_m^T\mathbf{v}_{ml}. \tag{3}$$

F. Optimization Scheme

As it is typical in AJD algorithms, the estimation of alignment matrices $\mathbf{B}_m$ is factorized in two stages: a pre-whitening stage (Section II-D) and an optimization stage. The optimization problem is formulated as

$$\underset{\mathbf{U}_i,\ldots,\mathbf{U}_M}{\operatorname{argmin}}\sum_k\sum_{i\neq j=1}^{M}\left\|off\left(\mathbf{U}_i^T\mathbf{R}_{ijk}\mathbf{U}_j\right)\right\|_F^2 \tag{4}$$

with appropriate constraints on the $\mathbf{U}_m$ matrices.

Following [41], a possible strategy for solving (4) is to cyclically solve it for each matrix $\mathbf{U}_m$ (each domain), holding the others fixed, until convergence. The cost function to be solved for each domain separately reads then

$$\underset{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}{\operatorname{argmin}}\left(\Psi^{off}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}\right),\ \text{with}\ \Psi^{off}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}=2\sum_k\sum_{j\neq i}\left\|off\left(\mathbf{Q}_{ijk}\right)\right\|_F^2, \tag{5}$$

where for convenience we have posed $\mathbf{Q}_{ijk}$=$\mathbf{U}^T_i\mathbf{R}_{ijk}\mathbf{U}_j$ and $\mathbf{U}_{\setminus\{i\}}$=$\{\mathbf{U}_m \mid m=1,\ldots,M, m\neq i\}$. The off-diagonal elements of the $\mathbf{Q}_{ijk}$ matrices are given by

$$\Psi^{off}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}=\Psi^{total-diag}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}=\Psi^{total}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}-\Psi^{diag}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}, \tag{6}$$

where the *total* and *diagonal* parts are

$$\Psi^{total}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}=\sum_k\sum_{j\neq i}2tr\left(\mathbf{Q}_{ijk}\mathbf{Q}_{ijk}^T\right)\ \text{and}$$

$$\Psi^{diag}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}}=2\sum_k\sum_{j\neq i}\sum_p\left(\mathbf{u}^T_{i(p)}\mathbf{R}_{ijk}\mathbf{u}_{j(p)}\right)^2, \tag{7}$$

respectively. In (7) and everywhere hereafter, $\mathbf{u}_{i(p)}$ and $\mathbf{u}_{j(p)}$ are the p^{th} column vector of $\mathbf{U}_i$ and $\mathbf{U}_j$, respectively. We constraint the norm of the column vectors of $\mathbf{U}_i$, such that

$$\Psi^{off}_{\mathbf{U}_i|\mathbf{U}_{\setminus\{i\}}},\ \text{subject to}\ \mathbf{u}^T_{i(p)}\mathbf{M}_{i(p)}\mathbf{u}_{i(p)}=1,\ \forall p=1,\ldots,P, \tag{8}$$

where matrices $\mathbf{M}_{i(p)}$ are defined as

$$\mathbf{M}_{i(p)}=\sum_k\sum_{j\neq i}\left(\mathbf{R}_{ijk}\mathbf{u}_{j(p)}\mathbf{u}^T_{j(p)}\mathbf{R}^T_{ijk}\right). \tag{9}$$

In the AJD literature, constraint (9) is known as the *intrinsic constraints* [43].

As shown in [38], [41], the solution for the $\mathbf{u}_{i(p)}$ vectors of each matrix $\mathbf{U}_i$ is the principal generalized eigenvector of matrix $\mathbf{M}_{i(p)}$ in the metric of $\mathbf{M}_i$. We do not find such eigenvectors explicitly, rather, we limit ourselves to a single power iteration per update step, followed by a normalization enforcing the constraint. This yields the simple updating rule:

$$\text{For } i=1,\ldots,M,\ p=1,\ldots,P \text{ do}\ \begin{cases}\mathbf{u}_{i(p)}\leftarrow\mathbf{M}_i^{-1}\mathbf{M}_{i(p)}\mathbf{u}_{i(p)}\\ \mathbf{u}_{i(p)}\leftarrow\mathbf{u}_{i(p)}\left(\mathbf{u}^T_{i(p)}\mathbf{M}_{i(p)}\mathbf{u}_{i(p)}\right)^{-1/2}\end{cases}$$

As a computation shortcut, in the update rule we do not need to compute the inverse of $\mathbf{M}_i$; instead we proceed equivalently by computing its Cholesky decomposition and solving two triangular system of equations. Notice that the updating rule is applied cyclically on all $\mathbf{U}_m$ matrices.

Following [42], we initialize each matrix $\mathbf{U}_m$ by

$$\mathbf{U}_i=\mathbf{\Pi},\ \text{where SVD}\left(\sum_k\sum_{j\neq i=1}^{M}\mathbf{R}_{ijk}\right)=\mathbf{\Pi\Sigma V}^T$$

in order to hasten convergence. We name this *smart initialization*. Upon convergence of all matrices $\mathbf{U}_m$ we normalize all their columns such as

$$\mathbf{u}_{i(n)}\leftarrow\left(\mathbf{u}^T_{i(n)}\sum_k\left(\mathbf{R}_{ijk}\right)\mathbf{u}_{i(n)}\right)^{-1/2}\mathbf{u}_{i(n)}\ \text{ for all }\ \begin{matrix}n\in\{1,\ldots,N\}\\ i\neq j\in\{1,\ldots,M\}\end{matrix}$$

and we obtain the M alignment matrices as $\mathbf{B}_m$=$\mathbf{W}_m\mathbf{U}_m$.

Algorithm 1 GALIA (**G**roup **ALI**gnment **A**lgorithm)

Input: $\mathbf{T}_{mk}$ $m \in \{1,...,M\}$ and $k \in \{1,...,K\}$

subspace dimension $P << E$

Output: $\mathbf{B}_m = \mathbf{W}_m \mathbf{U}_m \quad \forall m \in \{1,...,M\}$

Begin:

$\mathbf{W}_m^T \left[\sum_k \left(\mathbf{T}_{mk} \mathbf{T}_{mk}^T \right) \right] \mathbf{W}_m = \mathbf{I}_P \quad \forall m \in \{1,...,M\}$

for all i=1 to M **do**

  $\mathbf{R}_{ijk} = \mathbf{W}_i^T \mathbf{T}_{ik} \mathbf{T}_{jk}^T \mathbf{W}_j \quad \forall i \neq j \in \{1,...,M\}$

end for

Initialize all matrices $\mathbf{U}_m$:

$\mathbf{U}_m = \mathbf{\Pi}$, where SVD$(\sum_k \sum_{m \neq j=1}^{M} \mathbf{R}_{ijk}) = \mathbf{\Pi \Sigma V} \quad \forall m$

repeat

  for all m=1 to M **do**

    $\mathbf{M}_{m(p)} = \sum_k \sum_{j \neq m=1}^{M} \left(\mathbf{R}_{mjk} \mathbf{u}_{j(p)} \mathbf{u}_{j(p)}^T \mathbf{R}_{mjk}^T \right) \quad \forall p$

    $\sum_{p=1}^{P} \mathbf{M}_{m(p)} = LL^T$ (Cholesky decomposition)

    for all p=1 to P **do**

      solve $Lf = \mathbf{M}_{m(p)} \mathbf{u}_{m(p)}$ and $L^T g = f$

      update $\mathbf{u}_{m(p)} \leftarrow g \left(g^T \mathbf{M}_{m(p)} g \right)^{-1/2}$

    end for

  end for

until convergence of all $\mathbf{U}_m$ matrices.

Normalize all columns of matrices $\mathbf{U}_m$:

$\mathbf{u}_{i(p)} \leftarrow \left(\mathbf{u}_{i(p)}^T \sum_k \left(\mathbf{R}_{ijk} \right) \mathbf{u}_{i(p)} \right)^{-1/2} \mathbf{u}_{i(p)} \quad \forall \begin{matrix} p \in \{1,...,P\} \\ i \neq j \in \{1,...,M\} \end{matrix}$

The pseudo-code is given in Algorithm 1. We name this alignment algorithm GALIA (***G**roup **ALI**gnment **A**lgorithm*). For further details on the optimization scheme, cost function and initialization procedure, the reader is referred to [38]-[41]. Notice that the bootstrapping size (A) and the pre-whitening dimension (P) are the only hyper-parameters of the algorithm. An efficient implementation of GALIA is provided in a GitHub repository [31].

G. Fast Alignment of New Domains

Suppose we have aligned the data of M subjects (and/or sessions), which is possibly very large, and trained a complex classifier on them. Then suppose that we want to apply this mighty classifier to a new subject (or new session) that was not available when we constructed the model. As we have stated the group learning problem, the M+1 domains must be re-aligned altogether, running again the group alignment algorithm. Fortunately, this is not strictly necessary, as we can align the new subject (or session) to the existing model in a much faster way. The problem can be posed this way: given M pre-computed alignment matrices $\mathbf{U}_m$, we want to find a matrix $\mathbf{U}_x$ for the new domain so as to maximize the cross outer-product between the new (pre-whitened) domain and all (pre-whitened) existing domains. That is to say, we force the new domain to adapt itself to the group, while the group is left untouched. The optimization problem is formulated then as

$$\underset{\mathbf{U}_x | \{\mathbf{U}_1,...,\mathbf{U}_M\}}{\operatorname{argmin}} \sum_k \sum_{j=1}^{M} \left\| off\left(\mathbf{U}_x^T \mathbf{R}_{xjk} \mathbf{U}_j \right) \right\|_F^2 . \tag{10}$$

The functional in (10) has the exact same form of the one in (5). As we have seen, the solution for each of the P columns of $\mathbf{U}_x$ is given by the principal generalized eigenvector of matrix $\mathbf{M}_{x(p)} = \sum_m (\mathbf{R}_{xm} \mathbf{u}_{m(p)} \mathbf{u}_{m(p)}^T \mathbf{R}_{xm}^T)$ in the metric of $\mathbf{M}_x = \sum_p \mathbf{M}_{x(p)}$.

In contrast to Algorithm 1, where (5) is one of M nested cost functions, here we only need to solve (10), therefore here we explicitly compute the P generalized eigenvalues. Thus, once we pre-whiten the surrogated feature vectors of the new domain we find $\mathbf{U}_x$ by a deterministic solution and we can construct matrix $\mathbf{B}_x$. We then align the feature vectors of the new domain using the same projection (3) namely,

$$\mathbf{z}_{xl} \leftarrow \mathbf{B}_x^T \mathbf{v}_{xl}$$

and apply to them the pre-computed machine learning model as it is. Note that group learning operates by cyclically optimizing the functional in (5). In each of these functionals matrix $\mathbf{U}_i$ is optimized given all the others. It is easy to see that as the number of domains goes to infinity, the influence of a single domain vanishes. This implies that the fast alignment is asymptotically equivalent to group learning. In practice, this means that group learning models trained on large databases can be expected to yield excellent results on any unseen target data.

H. Description of Data

We tested the proposed algorithm on six P300 databases including 333 sessions recorded from 177 subjects. The main characteristics of the databases are given in Table I. For details on the databases and for the experimental procedure, the reader is referred to [44]–[49]. All six databases are publicly available on the *MOABB* framework [50].

TABLE I
MAIN CHARACTERISTICS OF THE BCI DATABASES USED IN THIS STUDY

Databases	Subjects (Sessions)	Num. of Channels	Channel Type	Sampling Rate
bi2013a	22(1)	16	Ag/AgCl	128 Hz
bi2014a	64(1)	16	Gold	512 Hz
bi2014b	31(3)	31	Ag/AgCl	512 Hz
bi2015a	42(3)	31	Ag/AgCl	512 Hz
BNCI2014008	8(1)	8	Ag/AgCl	256 Hz
BNCI2015003	10(2)	8	Ag/AgCl	256 Hz

I. Pipelines

Each recorded session defines a separate domain, in which we have divided the available tangent vectors $\mathbf{v}_{ml}$ into train-test splits with random shuffling. The sizes of the training splits are arranged such that they feature the same percentage of trials (starting from 20% up to 90%, with 10% increment) of all available trials from each class for the given domain. The remaining trials are retained as test splits. This allows to test the classification performance of the group learning algorithm for different amounts of available training data. The splitting procedure is repeated five times; the reported accuracies are the average of those five folds.

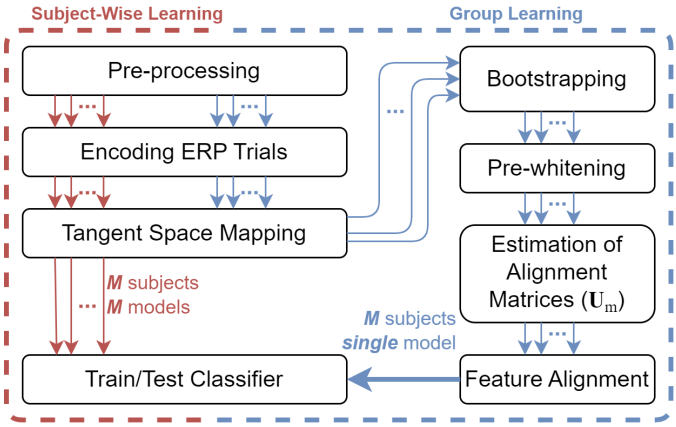

Fig. 2. Pipeline steps of subject-wise train-test learning and group learning summarized as a flowchart. Notice that not only that group learning pipeline has a few extra steps before cross-validation the classifier, but also it uses single classifier for all domains.

All computations were carried out using the Julia programming language (release v1.7.3) on a computer equipped with Windows 10 OS, an Intel i9-10900K @3.7GHz CPU and 64 GB of RAM. For classification, the linear support vector machine (LinearSVC) of the scikit-learn (version 1.2.2) Python library was exported to Julia.

Since the classes are unbalanced, balanced accuracy is employed as a performance index. Apart from the class weights parameter which is set to *balanced*, the default values are used for the parameters of the LinearSVC classifier.

Importantly, the train-test splits are always identical in the comparisons of pipelines. For the group learning and fast alignment, only *one* classifier is trained using data from all sessions of all subjects, whereas in the subject-specific case M classifiers are trained, each one using only the training data of the subject/session under test. Exactly the same training data is used for training the subject-specific classifiers, to align the group learning model and to train the group learning/fast alignment model, yielding a fair comparison of the three methods. The step-by-step summary of group learning and subject-specific train-test (subject-wise learning) pipelines are shown in Fig. 2. Fast alignment is detailed in Section II-G.

III. Results

In order to compare the decoding performances of the group learning, fast alignment and subject-wise learning pipelines, we consider subject-by-subject classification accuracies.

A. Hyper-parameters and Computational Cost

Our first analysis addresses the choice of the best pre-whitening dimension (P) and bootstrapping size (A). We have tested the group learning pipeline with six different pre-whitening dimension values $P \in \{4, 8, 16, 24, 32, 48\}$ for the whitening matrices $\mathbf{W}_m$. As for the bootstrapping size (A) of the surrogate features, we tested four different values $A \in \{1, 2, 10, 25\}$. When A=1, surrogate features are created with randomly sampled (with replacement) feature vectors directly. For A>1, the surrogate features are estimated by taking the mean of A randomly selected feature vectors. A grid search for the hyper-parameters of GALIA are thereby created.

In Fig. 3, the average classification changes of all split sizes obtained with the group learning pipeline is shown. Line plots

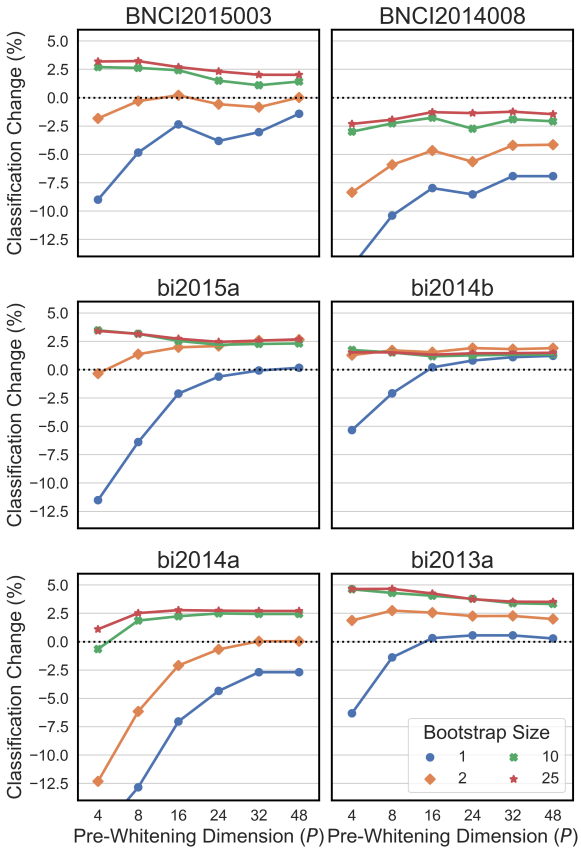

Fig. 3. Each plot shows how much in percentage the average classification accuracy changes when using the group learning model (with the given hyper-parameters) as compared to the subject-wise train-test model for each database. The horizontal dotted black line indicates no change (0%).

for each A value are shown along ascendingly ordered pre-whitening dimensions for all bootstrapping sizes and for each database separately. It can be seen that for the smaller bootstrapping size (A=1 or A=2) the group learning yields lower classification performance, suggesting that choosing a higher bootstrap size better retains distinctive features of the classes. Also, the classification performance of the group learning becomes less sensitive to pre-whitening dimension as the bootstrapping size increases. Given a sufficient value of A, the average classification change is nearly flat across all pre-whitening dimensions and for all databases.

As we have described in Section II-F, the dimensions of the $\mathbf{U}_m$ matrices are defined by pre-whitening dimension. Consequently, the computational cost of GALIA (*Algorithm 1*) depends on the choice of P. Fig. 4 shows the computational cost of GALIA in terms of runtime for each value of P. We should note that the computational costs shown in Fig. 4 depends not only on P, but also on the number of subjects (M) composing the group. Therefore, the nominal runtime of GALIA is higher for larger databases such as bi2015a and bi2014b. Nevertheless, the cost of doubling the pre-whitening dimension is nearly 10 times in terms of the runtime of the algorithm, regardless the group size. In absolute terms, the runtime is modest in our Julia implementation as it is comprised in between 10 and 1000 seconds for P=16, depending on the group size.

As illustrated in this section, the classification performance of group learning does not require separate fine tuning of hyper-

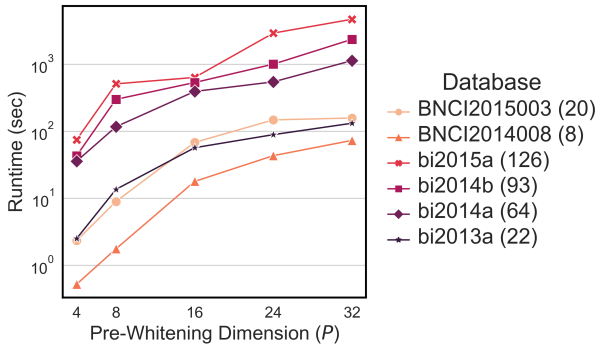

Fig. 4. Average runtime (s) to run the group alignment algorithm versus *P*. The average runtime axis (y-axis) is displayed on a logarithmic scale. The total number of subjects/sessions of each database reported in parenthesis.

parameters A and P for each database we have used. Instead, a suboptimal hyper-parameter configuration with low computational cost and generalizing well across all databases regardless of their recording sensor type, number of sensors and number of subjects can be used for all. Consequently, in this study we set the bootstrapping size to A=25 and pre-whitening dimension to P=16 as the hyper-parameters of GALIA.

B. Comparison of Group Learning vs Subject-Wise Learning

In Fig. 5, group learning versus subject-wise learning classification accuracies of each subject are presented. The train and test splits have equal numerosity in each class (50%, 50%). It can be seen that for the vast majority of subjects group learning yields higher classification accuracy as compared to subject-wise learning. If we take a closer look at bi2014b database, only one third of the subjects display subject-wise classification accuracy that is equal or higher than 60%. Nevertheless, group learning improves the classification accuracy of the two thirds of the subjects by 1–5%, yielding 1.4% improvement on average. Conversely, in BNCI2015003, bi2013a, bi2014a and bi2015a databases, subject-wise classification accuracies are higher than 60% for the vast majority of subjects. On average, classification performances of these databases are improved by 2.7%, 4.2%, 2.81% and 2.93% respectively, as shown in Table II. Wilcoxon signed-rank tests (Table III) reveal that group learning significantly improves the classification performance of all databases except BNCI2014008 ($p<0.001$) in almost all cases. This is a clear evidence of positive learning achieved by GALIA, especially for those databases (BNCI2015003, bi2013a, bi2014a, bi2015a) displaying high subject-specific accuracy.

In addition to the subject-by-subject comparison, we present the average classification accuracy of both pipelines with respect to increasing training data size in Fig 6. It can be seen that in all databases the average classification accuracy of group learning matches -if not surpasses- the average classification accuracy of subject-wise learning, regardless the train-test split size. As the training data increases, group learning yields significantly higher classification accuracy than subject-wise learning for all databases except for the BNCI2014008 database.

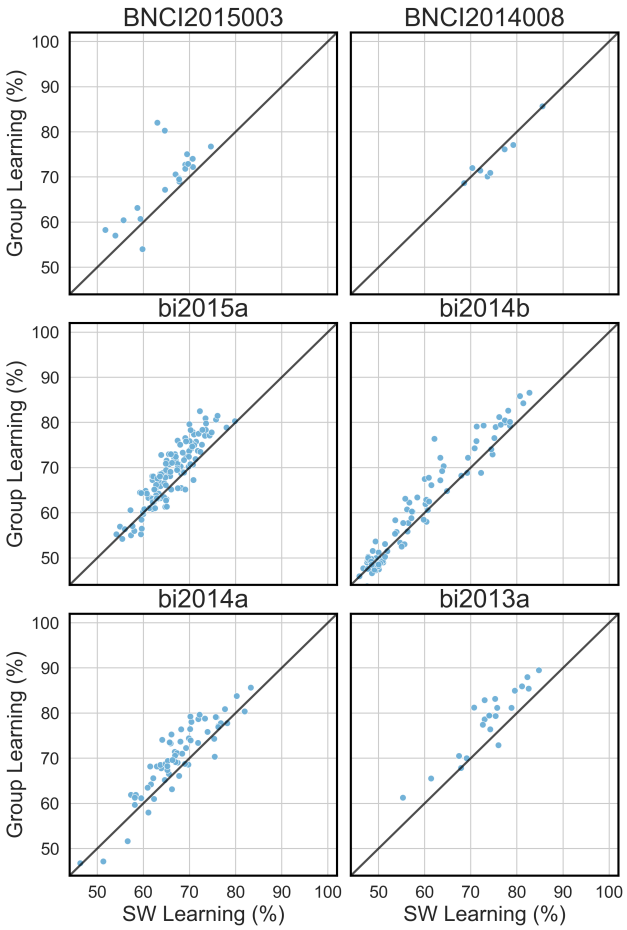

Fig. 5. Scatter plots comparing individual-subject accuracies of group learning versus subject-wise learning pipelines for each database.

TABLE II
AVERAGE CLASSIFICATION ACCURACIES OF EACH DATABASE

Databases	SW Learning (%)	Group Learning (%)	Fast Alignment (%)
bi2013a	76.10±0.30	80.30±0.30	80.34±0.30
bi2014a	68.48±0.11	71.29±0.13	71.28±0.13
bi2014b	60.31±0.38	61.69±0.43	61.74±0.47
bi2015a	67.81±0.14	70.74±0.15	70.59±0.22
BNCI2014008	75.93±0.60	74.66±0.66	74.22±0.66
BNCI2015003	67.64±0.61	70.34±0.71	69.03±0.69

Classification accuracies averaged over all train-test split sizes for each database and corresponding standard errors.

TABLE III
WILCOXON SIGNED-RANK TEST P-VALUES COMPARING THE AVERAGE ACCURACY OF GROUP LEARNING (GL) AND SUBJECT-WISE LEARNING (SW). SIGNIFICANT RESULTS AFTER CORRECTION BY THE BONFERRONI METHOD (GL>SW) AT THE α=0.05 LEVEL ARE PRINTED IN BOLD.

Databases	Training Split Size							
	20	30	40	50	60	70	80	90
bi2013a	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
bi2014a	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
bi2014b	0.946	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	0.046
bi2015a	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
BNCI2014008	0.008	0.770	0.844	0.926	0.992	1.000	1.000	0.973
BNCI2015003	0.004	**<0.001**	**<0.001**	**<0.001**	**<0.001**	0.002	**<0.001**	0.285

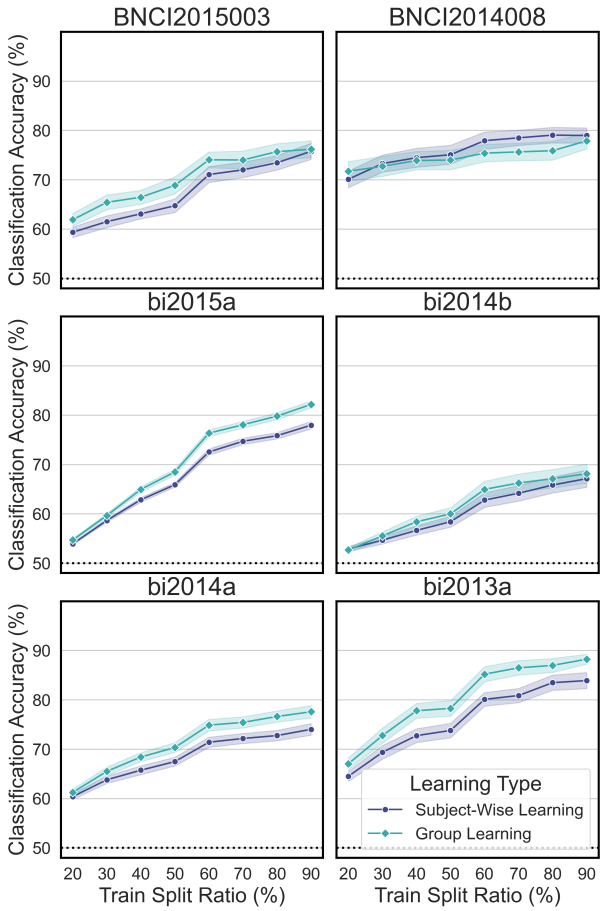

Fig. 6. Average classification accuracy of group learning and subject-wise learning across all training split sizes for all databases. The horizontal dotted black line indicates chance level (50%).

C. Robustness to Negative Learning of GALIA

In the above analysis, none of the subjects are excluded from group learning. However, most previous Riemannian transfer learning methods were shown to be sensitive to the choice of the source domains [25]. In order to verify whether GALIA also displays such behavior, we exclude subjects that have less than 60% subject-wise learning classification accuracy. Next, the group learning pipeline is run with exactly the same parameters on the remaining subjects. In Fig. 7, we compare the average classification accuracies of group learning for each database using all retained subjects before and after excluding low-scoring subjects. Considering that the BNCI2014008 database has no subjects scoring below 60% and that the bi2013a database has only one subject scoring barely lower than 60%, these two databases are excluded from this analysis. As for the remaining databases, the average classification accuracy does not display significant differences according to the Wilcoxon signed-rank test (p-value not shown). This analysis stresses the robustness of GALIA against the inclusion in the group model of low-scoring subjects, and suggests that it does not require pre-selection of subjects to be included in the group learning in order to avoid negative learning. This is a distinctive advantage over previous attempts, similar to what has been noticed in [26].

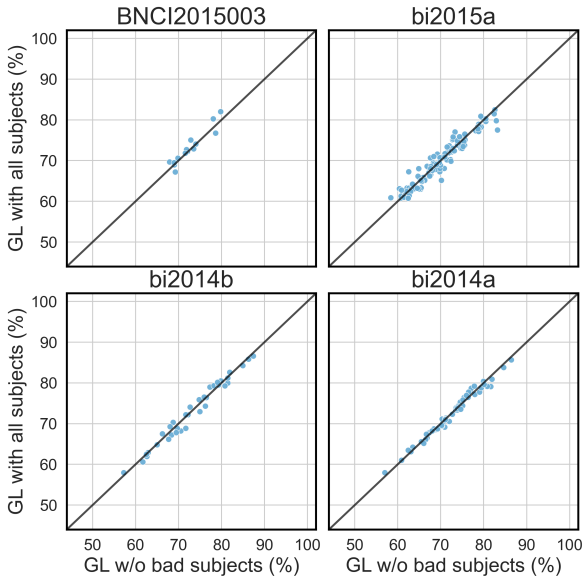

Fig. 7. Scatter plots comparing individual-subject accuracies of subjects for group learning models trained with and without low-scoring subjects.

D. Fast Alignment

To reiterate briefly, GALIA applies to a set of subjects/sessions, which are treated simultaneously as source and target domains. If the size of the group changes, all of the alignment matrices need to be recalculated again. Considering the long-run aim of the group learning (that is to create an MLM that can be applied on virtually any unseen data), it is undesirable to start over the estimation of alignment matrices every time a new target subject has to be treated when we need to apply the model on an unseen subject.

Here we present a *leave-one-out* (LOO) analysis in which an unseen subject is aligned to the set of subjects without re-running GALIA from scratch. In each turn, one subject/session is relegated from the group whereas the remaining subjects/sessions are used to train GALIA. The relegated subject/session is adapted to the group with fast alignment using its training data as described in Section II-G. Then, the classifier trained on the group is used to classify the data of the newly aligned subject. Notice that in this analysis the classifier has never seen any of the trials of the target data, effectively achieving a many-to-one domain adaptation mode of operation. In Fig. 8, subject-by-subject comparisons of fast alignment, group learning and subject-wise learning are shown when 50% training split size is used. It can be seen that fast alignment of a new subject yields approximately the same classification performance as compared to the situation where the subject is included in the group model. We repeat this analysis for all training split sizes and compute p-values using the Wilcoxon signed-rank test (Table IV). For all tests the null hypothesis cannot be rejected, suggesting that the average classification accuracy achieved by fast alignment is equivalent to the one achieved by the group learning pipeline, even with a small amount of training trials.

TABLE IV
WILCOXON SIGNED-RANK TEST P-VALUES COMPARING THE AVERAGE ACCURACY OF GROUP LEARNING (GL) AND FAST ALIGNMENT (FA). SIGNIFICANT RESULTS AFTER CORRECTION BY THE BONFERRONI METHOD (GL>FA) AT THE α=0.05 LEVEL ARE PRINTED IN BOLD.

Databases	Training Split Size							
	20	30	40	50	60	70	80	90
bi2013a	0.194	0.960	0.947	0.555	0.358	0.524	0.228	0.564
bi2014a	0.504	0.336	0.421	0.372	0.613	0.130	0.376	0.160
bi2014b	0.915	0.869	0.978	0.975	0.355	0.198	0.424	0.757
bi2015a	0.989	0.927	1.000	0.553	0.319	0.697	0.705	0.775
BNCI2014008	0.422	0.191	**0.004**	0.055	0.680	0.098	0.020	0.074
BNCI2015003	0.806	0.215	0.507	0.715	0.378	0.595	0.273	0.778

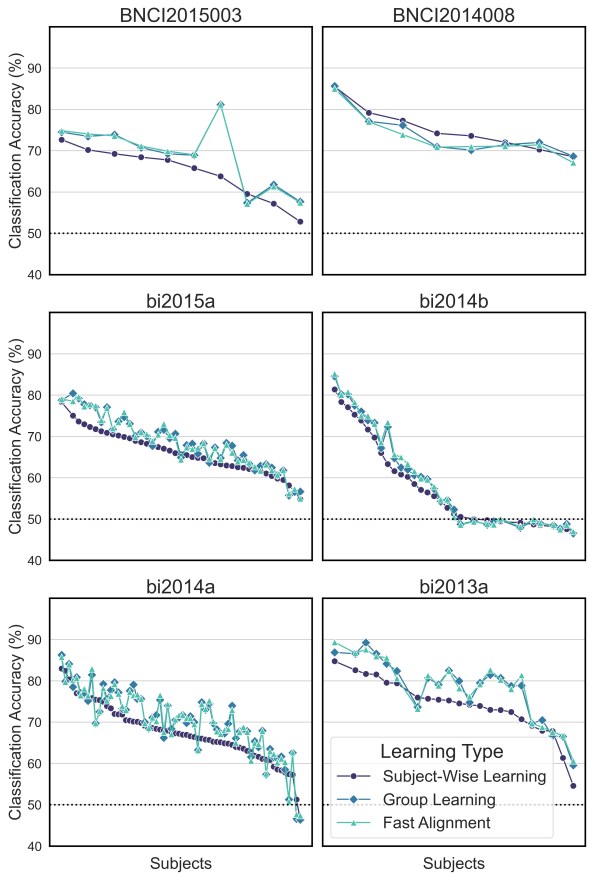

Fig. 8. Subject-by-subject classification accuracies of subject-wise learning, group learning and fast alignment for all databases. Subjects are sorted based on their subject-wise learning accuracies. The horizontal dotted black line indicates chance level (50%).

IV. DISCUSSION

In this paper, we have expanded the *many-to-many* transfer learning method in [30] by introducing a smart initialization method to hasten the convergence rate of the group alignment algorithm (GALIA) and by showing how to choose optimal hyper-parameters of GALIA generalizing well across databases. We have also introduced the fast alignment method (*many-to-one*) to align a new (unseen) domain with the formerly aligned group of domains without re-running GALIA and re-training or updating the classifier. Finally, we have extensively tested the group alignment and fast alignment method on six publicly available P300 databases comprising 333 sessions from 177 subjects.

Granting that the conventional subject-specific train-test pipeline is the golden standard for the purpose of classification, we showed that the group learning method outperformed the golden standard on all databases except BNCI2014008. The peculiarity of this database is that it is the only one in which the recordings were taken from clinical patients (Amyotrophic lateral sclerosis). Since GALIA seeks linear transformation of the feature vectors to align the data, the negative learning observed in this database may be explained by the fact that a linear transformation does not suffice to align these clinical data. Further studies are needed in order to test this hypothesis and to find a solution.

In order to ascertain whether the inter-domain aligment achieved by GALIA is the essential ingridient in order to achieve the results we report, we have performed an ablation study on the bi2013a and bi2015a databases. This is achieved replacing the matrices $\mathbf{U}_m$, the core of our alignment method, with the identity matrix and performed the cross-validation exactly is the same way otherwise. This procedure effectively disables the alignment, keeping equal all other aspects of the test and amount to a "naïve" group learning consisting in just pooling the data of all subjects in order to train a group classifier. The ablation of GALIA reduced the classification accuracy averaged over all train-test split sizes to below 60 % for both databases, well below the accuracy achieved by either group learning or subject-wise learning (compare these scores to Table II). We conclude in favor of the essentiality of our alignment procedure in our pipelines.

As compared to the previous Riemannian domain adaptation methods acting in either the manifold or in the tangent space [21], [23]–[28], group learning stands out for being the first many-to-many domain adaptation method. Favorably, there is no need to search for the optimal source domains in order to prevent negative learning unlike [23], [24], [27], [28], since the group learning is shown here to be robust against inclusion of low-performing subjects. Unlike the conventional one-to-one domain adaptation or its cascaded many-to-one versions that usually employ voting strategies (majority or weighted) on multiple classifiers each being trained with a different source domain [23], [24], [26]–[28], in group learning a single classifier is trained and tested with the aligned data of all domains. Hence, the data of many domains are truly forged in together for training a single classifier.

Remarkably, the classification performance of our fast alignment method is comparable to the one achieved by group learning, that is, the classifier works as well without seeing the target data *at all*. Nonetheless, labeled trials are needed in order to align the target data. A promising line of research consists in making the fast alignment method fully unsupervised.

As seen in Table V, the alignment matrix of a new domain can be computed in at most three seconds on a regular desktop PC. For any practical purpose this implies that many-to-one domain adaptation can be performed online.

Our results suggest that a single MLM can be trained with the massive amount of data that belongs to the group of aligned domains thanks to the fast alignment method. Once trained, the MLM can serve for the classification of any new domain simply by aligning its feature vectors with the feature vectors of the

MLM, without any further training of the classifier. This may be used as a pre-processing step in deep neural network (DNN) methods such as [51] to tackle domain adaptation on massive amounts of data, which is what a DNN needs. An even more intriguing possibility would be to design a specific DNN layer to perform group data alignment along the lines of GALIA.

This study has several limitations: we did not apply group learning for cross-database domain adaptation due to the incongruence among databases caused by sensor types and numbers. An adaptation of the dimensionality transcending method proposed in [52] is currently under study in order to align incongruent EEG data. This would allow the creation of *universal* MLM based on GALIA fast alignment, that is, pre-trained classifiers that can be used on any target domain data, regardless the number and position of electrodes. Besides, the possible non-linearity of the data will be considered for enhancing the alignment of the domains. Finally, we have tested group learning and fast alignment only on P300 BCI data; future studies are needed to test our proposition on other BCI paradigms, such as motor-imagery and SSVEP.

TABLE V
RUNTIME OF SUBJECT-WISE LEARNING, GROUP LEARNING AND FAST ALIGNMENT

Databases	SW Pipeline	GL Pipeline	Fast Alignment
bi2013a	8.04	+62.49	+0.82
bi2014a	127.26	+301.03	+2.13
bi2014b	125.23	+599.15	+2.71
bi2015a	190.77	+726.67	+2.41
BNCI2014008	26.21	+18.89	+0.74
BNCI2015003	14.74	+71.77	+0.86

Subject-wise learning pipeline runtimes are given in seconds. The runtime values of group learning and fast alignment are in comparison to subject-wise learning.

REFERENCES

[1] J. D. Millan, P. W. Ferrez, F. Galan, E. Lew, and R. Chavarriaga, "Non-invasive brain-machine interaction," *Intern J Pattern Recognit Artif Intell*, vol. 22, no. 5, pp. 959–972, 2008.

[2] H. J. Hwang, S. Kim, S. Choi, and C. H. Im, "EEG-Based Brain-Computer Interfaces: A Thorough Literature Survey," *Int J Hum Comput Interact*, vol. 29, no. 12, pp. 814–826, 2013, doi: 10.1080/10447318.2013.780869.

[3] L. Korczowski, A. Barachant, A. Andreev, C. Jutten, and M. Congedo, "'Brain Invaders 2': an open source Plug & Play multi-user BCI videogame," *6th International Brain-Computer Interface Meeting*, 2016.

[4] D. J. McFarland and J. R. Wolpaw, "Brain-computer interfaces for communication and control," *Commun ACM*, vol. 54, no. 5, p. 60, 2011, doi: 10.1145/1941487.1941506.

[5] A. Nijholt, R. J. K. Jacob, M. Andujar, B. F. Yuksel, and G. Leslie, "Brain-Computer Interfaces for Artistic Expression," *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, Montreal QC, Canada, pp. 1–7, 2018. doi: 10.1145/3170427.3170618.

[6] L. Mayaud *et al.*, "Brain-computer interface for the communication of acute patients: a feasibility study and a randomized controlled trial comparing performance with healthy participants and a traditional assistive device," *Brain-Computer Interfaces*, vol. 3, no. 4, 2016, doi: 10.1080/2326263X.2016.1254403.

[7] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, and M. Grosse-Wentrup, "Transfer Learning in Brain-Computer Interfaces," *IEEE Comput Intell Mag*, vol. 11, no. 1, pp. 20–31, 2016, doi: 10.1109/Mci.2015.2501545.

[8] F. Fahimi, Z. Zhang, W. B. Goh, T. S. Lee, K. K. Ang, and C. Guan, "Inter-subject transfer learning with an end-to-end deep convolutional neural network for EEG-based BCI," *J Neural Eng*, vol. 16, no. 2, 2019, doi: 10.1088/1741-2552/aaf3f6.

[9] P. R. A. S. Bassi, W. Rampazzo, and R. Attux, "Transfer learning and SpecAugment applied to SSVEP based BCI classification," *Biomed Signal Process Control*, vol. 67, 2021, doi: 10.1016/j.bspc.2021.102542.

[10] M. Dehghani, A. Mobaien, and R. Boostani, "A deep neural network-based transfer learning to enhance the performance and learning speed of BCI systems," *Brain-Computer Interfaces*, vol. 8, no. 1–2, 2021, doi: 10.1080/2326263X.2021.1943955.

[11] Z. T. Wan, R. Yang, M. J. Huang, N. Y. Zeng, and X. H. Liu, "A review on transfer learning in EEG signal analysis," *Neurocomputing*, vol. 421, pp. 1–14, 2021, doi: https://doi.org/10.1016/j.neucom.2020.09.017.

[12] I. Dag, L. G. Dui, S. Ferrante, A. Pedrocchi, and A. Antonietti, "Leveraging Deep Learning Techniques to Improve P300-Based Brain Computer Interfaces," *IEEE J Biomed Health Inform*, vol. 26, no. 10, pp. 4892–4902, 2022, doi: 10.1109/JBHI.2022.3174771.

[13] W. Gao *et al.*, "Eliminating or Shortening the Calibration for a P300 Brain–Computer Interface Based on a Convolutional Neural Network and Big Electroencephalography Data: An Online Study," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 1754–1763, 2023, doi: 10.1109/TNSRE.2023.3259991.

[14] A. M. Azab, J. Toth, L. S. Mihaylova, and M. Arvaneh, "A review on transfer learning approaches in brain-computer interface," in *Signal Processing and Machine Learning for Brain-Machine Interfaces*, 2018. doi: 10.1049/PBCE114E_ch5.

[15] A. Kai Keng, C. Zheng Yang, Z. Haihong, and G. Cuntai, "Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2390–2397. doi: 10.1109/IJCNN.2008.4634130.

[16] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms," *IEEE Trans Biomed Eng*, vol. 58, no. 2, pp. 355–362, 2011, doi: 10.1109/TBME.2010.2082539.

[17] K. Hyohyeong, N. Yunjun, and C. Seungjin, "Composite Common Spatial Pattern for Subject-to-Subject Transfer," *IEEE Signal Process Lett*, vol. 16, no. 8, pp. 683–686, doi: 10.1109/lsp.2009.2022557.

[18] H. Cho, M. Ahn, K. Kim, and S. Chan Jun, "Increasing session-to-session transfer in a brain-computer interface with on-site background noise acquisition," *J Neural Eng*, vol. 12, no. 6, 2015, doi: 10.1088/1741-2560/12/6/066009.

[19] Y. Xu *et al.*, "Transfer Learning Based on Regularized Common Spatial Patterns Using Cosine Similarities of Spatial Filters for Motor-Imagery BCI," *Journal of Circuits, Systems and Computers*, vol. 28, no. 7, 2019, doi: 10.1142/S0218126619501238.

[20] A. M. Azab, L. Mihaylova, K. K. Ang, and M. Arvaneh, "Weighted Transfer Learning for Improving Motor Imagery-Based Brain-Computer Interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 7, pp. 1352–1359, 2019, doi: 10.1109/TNSRE.2019.2923315.

[21] H. He and D. Wu, "Transfer Learning for Brain-Computer Interfaces: A Euclidean Space Data Alignment Approach," *IEEE Trans Biomed Eng*, vol. 67, no. 2, pp. 399–410, 2020, doi: 10.1109/TBME.2019.2913914.

[22] M. Congedo, A. Barachant, and R. Bhatia, "Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review," *Brain-Computer Interfaces*, vol. 4, no. 3, pp. 155–174, 2017, doi: 10.1080/2326263x.2017.1297192.

[23] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, "Transfer Learning: A Riemannian Geometry Framework With Applications to Brain-Computer Interfaces," *IEEE Trans Biomed Eng*, vol. 65, no. 5, pp. 1107–1116, 2018, doi: 10.1109/Tbme.2017.2742541.

[24] F. Li, Y. Xia, F. Wang, D. Zhang, X. Li, and F. He, "Transfer Learning Algorithm of P300-EEG Signal Based on XDAWN

Spatial Filter and Riemannian Geometry Classifier," *Applied Sciences*, vol. 10, no. 5, p. 1804, 2020.

[25] P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Riemannian Procrustes Analysis: Transfer Learning for Brain-Computer Interfaces," *IEEE Trans Biomed Eng*, vol. 66, no. 8, pp. 2390–2401, 2019, doi: 10.1109/Tbme.2018.2889705.

[26] A. Bleuze, J. Mattout, and M. Congedo, "Transfer Learning for the Riemannian Tangent Space: Applications to Brain-Computer Interfaces," in *7th International Conference on Engineering and Emerging Technologies, ICEET 2021*, 2021. doi: 10.1109/ICEET53442.2021.9659607.

[27] Q. She, Y. Cai, S. Du, and Y. Chen, "Multi-source manifold feature transfer learning with domain selection for brain-computer interfaces," *Neurocomputing*, vol. 514, pp. 313–327, 2022, doi: 10.1016/j.neucom.2022.09.124.

[28] Y. Liang and Y. Ma, "Calibrating EEG features in motor imagery classification tasks with a small amount of current data using multisource fusion transfer learning," *Biomed Signal Process Control*, vol. 62, 2020, doi: ARTN 102101 10.1016/j.bspc.2020.102101.

[29] E. K. Kalunga, S. Chevallier, and Q. Barthélemy, "Transfer Learning for SSVEP-based BCI using Riemannian similarities between users," in *European Signal Processing Conference*, 2018, pp. 1685–1689. doi: 10.23919/EUSIPCO.2018.8553441.

[30] M. Congedo, A. Bleuzé, and J. Mattout, "Group Learning by Joint Alignment in the Riemannian Tangent Space," in *XXVIIIème Colloque Francophone de Traitement du Signal et des Images (GRETSI)*, Nancy, France, 2022.

[31] F. Altindis and M. Congedo, "Group Alignment Algorithm (GALIA)." [Online]. Available: https://github.com/fatihaltindis/groupLearning

[32] M. Congedo, L. Korczowski, A. Delorme, and F. Lopes da silva, "Spatio-temporal common pattern: A companion method for ERP analysis in the time domain," *J Neurosci Methods*, vol. 267, 2016, doi: 10.1016/j.jneumeth.2016.04.008.

[33] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain-computer interface classification by Riemannian geometry," *IEEE Trans Biomed Eng*, vol. 59, no. 4, pp. 920–928, 2012, doi: 10.1109/TBME.2011.2172210.

[34] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, doi: 10.1016/j.neucom.2012.12.039.

[35] M. Congedo, A. Barachant, and A. Andreev, "A New Generation of Brain-Computer Interface Based on Riemannian Geometry." p. arXiv:1310.8115, 2013.

[36] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *J Multivar Anal*, vol. 88, no. 2, 2004, doi: 10.1016/S0047-259X(03)00096-4.

[37] M. Congedo, B. Afsari, A. Barachant, and M. Moakher, "Approximate joint diagonalization and geometric mean of symmetric positive definite matrices," *PLoS One*, vol. 10, no. 4, 2015, doi: 10.1371/journal.pone.0121423.

[38] M. Congedo, "EEG Source Analysis ," Université de Grenoble, 2013.

[39] M. Anderson, T. Adali, and X. L. Li, "Joint blind source separation with multivariate Gaussian model: Algorithms and performance analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, 2012, doi: 10.1109/TSP.2011.2181836.

[40] J. Vía, M. Anderson, X. L. Li, and T. Adali, "Joint blind source separation from second-order statistics: Necessary and sufficient identifiability conditions," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2011. doi: 10.1109/ICASSP.2011.5946997.

[41] M. Congedo, R. Phlypo, and J. Chatel-Goldman, "Orthogonal and non-orthogonal joint blind source separation in the least-squares sense," in *European Signal Processing Conference*, 2012.

[42] M. Congedo, R. Phlypo, and D. T. Pham, "Approximate joint singular value decomposition of an asymmetric rectangular matrix set," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, 2011, doi: 10.1109/TSP.2010.2087018.

[43] F. Bouchard, J. Malick, and M. Congedo, "Riemannian optimization and approximate joint diagonalization for blind source separation," *IEEE Transactions on Signal Processing*, vol. 66, no. 8, 2018, doi: 10.1109/TSP.2018.2795539.

[44] L. Korczowski *et al.*, "Brain Invaders Solo versus Collaboration: Multi-User P300-based Brain-Computer Interface Dataset (bi2014b)," 2019. doi: 10.5281/zenodo.3267301.

[45] L. Korczowski *et al.*, "Brain Invaders calibration-less P300-based BCI with modulation of flash duration Dataset (bi2015a)," 2019. doi: 10.5281/zenodo.3266930.

[46] G. F. P. Van Veen, A. Barachant, A. Andreev, G. Cattan, P. Rodrigues, and M. Congedo, "Building brain invaders: EEG data of an experimental validation," *arXiv*. 2019.

[47] L. Korczowski, E. Ostaschenko, A. Andreev, G. Cattan, P. Rodrigues, and M. Congedo, "Brain Invaders 2014a," May 2019, doi: 10.5281/ZENODO.2669495.

[48] C. Guger, G. Edlinger, W. Harkam, I. Niedermayer, and G. Pfurtscheller, "How many people are able to operate an EEG-based brain-computer interface (BCI)?," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2003, doi: 10.1109/TNSRE.2003.814481.

[49] A. Riccio *et al.*, "Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis," *Front Hum Neurosci*, vol. 7, 2013, doi: 10.3389/fnhum.2013.00732.

[50] V. Jayaram and A. Barachant, "MOABB: Trustworthy algorithm benchmarking for BCIs," *J Neural Eng*, vol. 15, no. 6, 2018, doi: 10.1088/1741-2552/aadea0.

[51] R. Kobler, J. Hirayama, Q. Zhao, and M. Kawanabe, "SPD domain-specific batch normalization to crack interpretable unsupervised domain adaptation in EEG," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., Curran Associates, Inc., 2022, pp. 6219–6235. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/28ef7ee7cd3e03093acc39e1272411b7-Paper-Conference.pdf

[52] P. L. C. Rodrigues, M. Congedo, and C. Jutten, "Dimensionality Transcending: A Method for Merging BCI Datasets with Different Dimensionalities," *IEEE Trans Biomed Eng*, vol. 68, no. 2, 2021, doi: 10.1109/TBME.2020.3010854.