Geodesic Optimization for Predictive Shift Adaptation on EEG data

Apolline Mellot[*], **Antoine Collas**[*]
Inria, CEA, Université Paris-Saclay
Palaiseau, France
apolline.mellot@inria.fr
antoine.collas@inria.fr

Sylvain Chevallier
TAU Inria, LISN-CNRS,
University Paris-Saclay, France.
sylvain.chevallier@
universite-paris-saclay.fr

Alexandre Gramfort
Inria, CEA, Université Paris-Saclay
Palaiseau, France
alexandre.gramfort@inria.fr

Denis A. Engemann
Roche Pharma Research and Early Development,
Neuroscience and Rare Diseases,
Roche Innovation Center Basel,
F. Hoffmann–La Roche Ltd., Basel, Switzerland.
denis.engemann@roche.com

Abstract

Electroencephalography (EEG) data is often collected from diverse contexts involving different populations and EEG devices. This variability can induce distribution shifts in the data X and in the biomedical variables of interest y, thus limiting the application of supervised machine learning (ML) algorithms. While domain adaptation (DA) methods have been developed to mitigate the impact of these shifts, such methods struggle when distribution shifts occur simultaneously in X and y. As state-of-the-art ML models for EEG represent the data by spatial covariance matrices, which lie on the Riemannian manifold of Symmetric Positive Definite (SPD) matrices, it is appealing to study DA techniques operating on the SPD manifold. This paper proposes a novel method termed Geodesic Optimization for Predictive Shift Adaptation (`GOPSA`) to address test-time multi-source DA for situations in which source domains have distinct y distributions. `GOPSA` exploits the geodesic structure of the Riemannian manifold to jointly learn a domain-specific re-centering operator representing site-specific intercepts and the regression model. We performed empirical benchmarks on the cross-site generalization of age-prediction models with resting-state EEG data from a large multi-national dataset (HarMNqEEG), which included 14 recording sites and more than 1500 human participants. Compared to state-of-the-art methods, our results showed that `GOPSA` achieved significantly higher performance on three regression metrics (R^2, MAE, and Spearman's ρ) for several source-target site combinations, highlighting its effectiveness in tackling multi-source DA with predictive shifts in EEG data analysis. Our method has the potential to combine the advantages of mixed-effects modeling with machine learning for biomedical applications of EEG, such as multicenter clinical trials.

1 Introduction

Machine learning (ML) has enabled advances in the analysis of complex biological signals, such as magneto- and electroencephalography (M/EEG), in diverse applications including biomarker

[*]Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1033

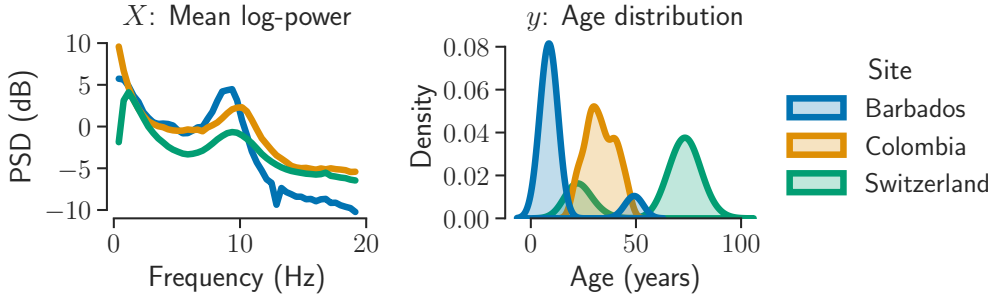

Figure 1: **Joint shift in X and y distributions on the HarMNqEEG dataset [33].** Subset of mean PSDs (**A**) and age distributions (**B**) from three recording sites used for the empirical benchmarks.

exploration [58, 20, 60] or developing Brain-Computer Interface (BCI) [57, 15, 1, 2]. However, a major challenge in applying ML to these signals arises from their inherent variability, a problem commonly referred to as dataset shift [12]. In the case of M/EEG data this variability can be caused by differences in recording devices (electrode positions and amplifier configurations), recording protocols, population demographics, and inter-subject variability [36, 14, 23, 29]. Notably, shifts can occur not only in the data X but also in the biomedical variable y we aim to predict, further complicating the use of ML algorithms.

Riemannian geometry has significantly advanced EEG data analysis by enabling the use of spatial covariance matrices as EEG descriptors [5, 4, 6, 38, 31, 35, 33, 48, 17, 56]. In [4, 6], the authors introduced a classification framework for BCI based on the Riemannian geometry of covariance matrices. These methods classify EEG signals directly on the tangent space using the Riemannian manifold of symmetric positive definite (SPD) matrices ($\mathbb{S}_d^{++}$), effectively capturing spatial information. More recently, [47, 48, 7] extended this framework to regression problems from M/EEG data in the context of biomarker exploration. Furthermore, [47, 48] proved that Riemannian metrics lead to regression models with statistical guarantees in line with log-linear brain dynamics [10] and are, therefore, well-suited for neuroscience applications. Across various biomarker-exploration tasks and datasets, recent work has shown that Riemannian M/EEG representations offer parameter-sparse alternatives to non-Riemannian deep learning architectures [13, 40, 17].

Domain adaptation addresses the challenges posed by differences in data distributions between source and target domains, e.g., when data are recorded with different cameras in computer vision [55], different writing styles in natural language processing [32], or varying sensor setups in time series analysis [22]. In particular, DA considers *target shift* where the shift is in the outcome variable y. For classification it means source and target data share the same labels but in different proportions [34]. Target shift is also frequent in the context of multicenter neuroscience studies, as the studied population of one site may vary significantly from the studied population of another site (cf. Figure 1). To tackle various sources of variability in neurophysiological data like EEG, there is a need for a DA approach that can deal with a joint shift in X and y.

Related work In [62], the authors addressed DA for EEG-based BCI using re-centering affine transformation of covariance matrices (Section 2) to align data from different sessions or human participants, improving classification accuracies. Yair et al. [59] extended this with parallel transport showing its effectiveness in EEG analysis, whereas, Peng et al [43] introduced a domain-specific regularizer based on the Riemannian mean. Notably, this parallel transport approach reduces to [62] when the common reference point is the identity. In a deep learning context, Kobler et al. [31] proposed to do a per-domain online re-centering which can be seen as a domain specific Riemannian batch norm. Going beyond re-centering, Riemannian Procrustes Analysis (RPA) [45] was proposed for EEG transfer learning, using three steps: mean alignment, dispersion matching, and rotation correction. However, the rotation step is unsuitable for regression problems and RPA adapts only a single source to a target domain. Recently, [36] demonstrated the benefits of re-centering for regression problems, showing improvements in handling task variations in MEG and enhancing across-dataset inference in EEG.

On the other hand, mixed-effects models (or multilevel models) have been successfully used to tackle data shifts in X and y [16, 25]. In biomedical data, mixed-effects models are crucial due to the presence of common effects, such as disease status and age. Riemannian mixed-effects models have been used to analyze observations on Riemannian manifolds, accommodating individual trajectories with mixed effects at both group and individual levels [30, 49, 50]. These models adapt a base point on the manifold for each data point and utilize parallel transport for this adaptation, which is necessary for accurate trajectory modeling. However, they differ significantly from the problem we address in this work. Notably, the input data X are covariates (e.g., age or disease status) which belong to a Euclidean space and the variables y to predict belong to the manifold (e.g., MRI diffusion tensors on $\mathbb{S}_d^{++}$) which is the opposite of the paper's studied problem. This distinction is critical as it highlights that while both methods use the geometry of Riemannian manifolds, the nature of the predicted variables and the type of data used differ from existing Riemannian mixed-effects models.

Contributions In this work, we address the challenging problem of multi-source domain adaptation with predictive shifts on the SPD manifold, focusing on distribution shifts in both the input data X and the variable to predict y. We propose a novel method called Geodesic Optimization for Predictive Shift Adaptation (`GOPSA`). It enables mixed-effects modeling by jointly learning parallel transport along a geodesic for each domain and a global regression model common to all domains, with the assumption that the mean $\bar{y}_{\mathcal{T}}$ of the target domain is known. `GOPSA` aims to advance the state of the art by: *(i)* addressing shifts in both covariance matrices and the outcome variable y, *(ii)* being tailored for regression problems, and *(iii)* being a multi-source test-time domain adaptation method, meaning that once trained on source data, it can generalize to any target domain without requiring access to source data or retraining a new model.

We first introduce in Section 2 how to do regression from covariance matrices on the Riemannian manifold of $\mathbb{S}_d^{++}$. We also interpret classical learning methods on $\mathbb{S}_d^{++}$ from heterogeneous domains as parallel transports combined with Riemannian logarithmic mappings. This leads us to `GOPSA` in Section 3, which learns to parallel transport each domain, with algorithms at train and test times. Finally, in Section 4, we apply `GOPSA` as well as different baselines on simulated data and the HarMNqEEG dataset.

Notations Vectors and matrices are represented by small and large cap boldface letters respectively (e.g., $\boldsymbol{x}$, $\boldsymbol{X}$). The set $\{1, ..., K\}$ is denoted by $[\![1, K]\!]$. $\mathbb{S}_d^{++}$ and $\mathbb{S}_d$ represent the sets of $d \times d$ symmetric positive definite and symmetric matrices. $\mathrm{uvec} : \mathbb{S}_d \to \mathbb{R}^{d(d+1)/2}$ vectorizes the upper triangular part of a symmetric matrix. Frobenius and 2-norms are denoted by $\|.\|_F$ and $\|.\|_2$, respectively. $\triangleq$ defines the left part of the equation as the right part. The transpose and Euclidean inner product operations are represented by $\cdot^\top$. $\mathbf{1}_n \triangleq [1, \ldots, 1]^\top$ denotes an n-dimensional vector of ones. For a loss function $\mathcal{L}$, $\nabla\mathcal{L}$ and $\mathcal{L}'$ denote its gradient and derivative. We consider K labeled source domains, each consisting of N_k covariance matrices, along with their corresponding outcome values, denoted by $\{(\boldsymbol{\Sigma}_{k,i}, y_{k,i})\}_{i=1}^{N_k}$. The target domain is $(\boldsymbol{\Sigma}_{\mathcal{T},i})_{i=1}^{N_{\mathcal{T}}}$ with the average outcome $\bar{y}_{\mathcal{T}}$.

2 Regression modeling from covariance matrices using Riemannian geometry

Riemannian geometry of $\mathbb{S}_d^{++}$ The covariance matrices belong to the set of $d \times d$ symmetric positive definite matrices denoted $\mathbb{S}_d^{++}$ [51, 44]. The latter is open in the set of $d \times d$ symmetric matrices denoted $\mathbb{S}_d$, and thus $\mathbb{S}_d^{++}$ is a smooth manifold [9]. A vector space is defined at each $\boldsymbol{\Sigma} \in \mathbb{S}_d^{++}$, called the tangent space, denoted $T_{\boldsymbol{\Sigma}}\mathbb{S}_d^{++}$, and is equal to $\mathbb{S}_d$, the ambient space. Equipped with a smooth inner product at every tangent space, a smooth manifold b ecomes a Riemannian manifold. To do so, we make use of the affine invariant Riemannian metric [51, 44]. Given $\boldsymbol{\Gamma}, \boldsymbol{\Gamma}' \in T_{\boldsymbol{\Sigma}}\mathbb{S}_d^{++}$, this metric is $\langle \boldsymbol{\Gamma}, \boldsymbol{\Gamma}' \rangle_{\boldsymbol{\Sigma}} = \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Gamma}'\right)$.

Riemannian mean The Riemannian distance (or geodesic distance) associated with the affine invariant metric is $\delta_R(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}') \triangleq \left\|\log\left(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-1/2}\right)\right\|_F$ with $\log : \mathbb{S}_d^{++} \to \mathbb{S}_d$ being the matrix logarithm (see Appendix A.1 for a definition). This distance is used to compute the Riemannian mean $\overline{\boldsymbol{\Sigma}}$ defined for a set $\{\boldsymbol{\Sigma}_i\}_{i=1}^N \subset \mathbb{S}_d^{++}$ as $\overline{\boldsymbol{\Sigma}} \triangleq \arg\min_{\boldsymbol{\Sigma} \in \mathbb{S}_d^{++}} \sum_{i=1}^N \delta_R(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_i)^2$.

This mean is efficiently computed with a Riemannian gradient descent [44, 63].

https://doi.org/10.52202/079017-1033

Riemannian logarithmic mapping The idea of the covariance-based approach is to define non-linear feature transformations into vectors that can be used as input for classical linear machine learning models. To do so, the Riemannian logarithmic mapping of $\boldsymbol{\Sigma}'$ at $\boldsymbol{\Sigma}$ is defined as

$$\log_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}') \triangleq \boldsymbol{\Sigma}^{1/2} \log\left(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-1/2}\right)\boldsymbol{\Sigma}^{1/2} \in T_{\boldsymbol{\Sigma}}\mathbb{S}_d^{++} \,. \tag{1}$$

Thus, matrices in the Riemannian manifold $\mathbb{S}_d^{++}$ are transformed into tangent vectors.

Parallel transport A classical practice to align distributions is parallel transport of covariance matrices from their mean to the identity and then apply the logarithmic mapping (1). Parallel transport along a curve allows to move SPD matrices from one point on the curve to another point on the curve while keeping the inner product between the logarithmic mappings with any other vector transported along the same curve constant. The following lemma gives the parallel transport of $\boldsymbol{\Sigma}'$ from $\boldsymbol{\Sigma}$ to $\boldsymbol{I}_d$ along the geodesic between these two points (See proof in Appendix A.2).

Lemma 2.1 (Parallel transport to the identity). *Given $\boldsymbol{\Sigma}, \boldsymbol{\Sigma}' \in \mathbb{S}_d^{++}$, the parallel transport of $\boldsymbol{\Sigma}'$ along the geodesic from $\boldsymbol{\Sigma}$ to the identity $\boldsymbol{I}_d$ at $\alpha \in [0,1]$ is*

$$\mathrm{PT}\left(\boldsymbol{\Sigma}', \boldsymbol{\Sigma}, \alpha\right) \triangleq \boldsymbol{\Sigma}^{-\alpha/2}\boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-\alpha/2} \,.$$

Learning on $\mathbb{S}_d^{++}$ The logarithmic mapping (1) at the identity is simply the matrix logarithm. Thus, a classical non-linear feature extraction [6, 36, 8] of a dataset $\{\boldsymbol{\Sigma}_i\}_{i=1}^N$ of Riemannian mean $\overline{\boldsymbol{\Sigma}}$ combines parallel transport and logarithmic mapping at the identity,

$$\phi\left(\boldsymbol{\Sigma}_i, \overline{\boldsymbol{\Sigma}}\right) \triangleq \mathrm{uvec}\left(\log_{\boldsymbol{I}_d}\left(\mathrm{PT}\left(\boldsymbol{\Sigma}_i, \overline{\boldsymbol{\Sigma}}, 1\right)\right)\right) = \mathrm{uvec}\left(\log\left(\overline{\boldsymbol{\Sigma}}^{-1/2}\boldsymbol{\Sigma}_i\overline{\boldsymbol{\Sigma}}^{-1/2}\right)\right) \in \mathbb{R}^{d(d+1)/2} \tag{2}$$

where uvec vectorizes the upper triangular part with off-diagonal elements multiplied by $\sqrt{2}$ to preserve the norm. Correcting dataset shifts by re-centering all source datasets [62], corresponds to parallel transporting data $\{\boldsymbol{\Sigma}_{k,i}\}_{i=1}^{N_k}$ of each domain $k \in [\![1, K]\!]$ from its Riemannian mean $\overline{\boldsymbol{\Sigma}}_k$ to the identity,

$$\phi(\boldsymbol{\Sigma}_{k,i}, \overline{\boldsymbol{\Sigma}}_k) = \mathrm{uvec}\left(\log\left(\overline{\boldsymbol{\Sigma}}_k^{-1/2}\boldsymbol{\Sigma}_{k,i}\overline{\boldsymbol{\Sigma}}_k^{-1/2}\right)\right) \,. \tag{3}$$

This method is the go-to approach for reducing shifts of the covariance matrix distributions coming from different domains and has been applied successfully for brain-computer interfaces [45, 59] and age prediction from M/EEG data [36].

3 Learning to recenter from highly shifted y distributions with GOPSA

In this section, we develop a novel multi-source domain adaptation method, called Geodesic Optimization for Predictive Shift Adaptation (GOPSA), that operates on the $\mathbb{S}_d^{++}$ manifold and is capable of handling vastly different distributions of y. Our approach implements a Riemannian mixed-effects model, which consists of two components: a single parameter estimating a geodesic intercept specific to each domain and a set of parameters shared across domains.

At train-time, GOPSA jointly learns the parallel transport of each of the K source domains and the regression model shared across domains. At test-time, we assume having access to the target mean response value $\bar{y}_{\mathcal{T}}$ and predict on the unlabeled target domain of covariance matrices $(\boldsymbol{\Sigma}_{\mathcal{T},i})_{i=1}^{N_{\mathcal{T}}}$. GOPSA focuses solely on learning the parallel transport of the target domain so that the mean prediction, using the regression model learned at train-time, matches $\bar{y}_{\mathcal{T}}$.

Parallel transport along the geodesic In Section 2, we presented how domain adaptation is performed on $\mathbb{S}_d^{++}$. In particular, (3) presents how to account for data shifts of each domain. However, this operator can only work if the variability between domains is considered as noise. As explained earlier, we are interested in shifts in both features and the response variable. Thus, (3) discards shift coming from the response variable and hence harms the performance of the predictive model. Based on the Lemma 2.1, we propose to parallel transport features to any point on the geodesic between a domain-specific Riemannian mean $\overline{\boldsymbol{\Sigma}}_k$ and the identity. Indeed, GOPSA parallel transports $\boldsymbol{\Sigma}_{k,i}$ on this geodesic with $\alpha \in [0,1]$ and then applies the Riemannian logarithmic mapping (1) at the identity,

$$\phi(\boldsymbol{\Sigma}_{k,i}, \overline{\boldsymbol{\Sigma}}_k, \alpha) \triangleq \mathrm{uvec}\left(\log_{\boldsymbol{I}_d}\left(\mathrm{PT}\left(\boldsymbol{\Sigma}_{k,i}, \overline{\boldsymbol{\Sigma}}_k, \alpha\right)\right)\right) = \mathrm{uvec}\left(\log\left(\overline{\boldsymbol{\Sigma}}_k^{-\alpha/2}\boldsymbol{\Sigma}_{k,i}\overline{\boldsymbol{\Sigma}}_k^{-\alpha/2}\right)\right). \tag{4}$$

This allows each domain to undergo parallel transport to a certain degree, effectively moving it toward the identity.

Algorithm 1: Train-Time GOPSA

Input: For all $k \in [\![1, K]\!]$, $\{(\boldsymbol{\Sigma}_{k,i}, y_{k,i})\}_{i=1}^{N_k}$, initialization of $\boldsymbol{\gamma}_{\mathcal{S}}$, step-sizes $\{\xi_t\}_{t\geq 1}$

for $k = 1 \to K$ **do**

  $\overline{\boldsymbol{\Sigma}}_k \leftarrow$ Riemannian mean of $\{\boldsymbol{\Sigma}_{k,i}\}_{i=1}^{N_k}$

end

$t \leftarrow 1$

while *not converged* **do**

  $\boldsymbol{Z}_{\mathcal{S}}(\boldsymbol{\gamma}_{\mathcal{S}}) \leftarrow$ Compute features with (7)

  $\boldsymbol{\beta}^{\star}_{\mathcal{S}}(\boldsymbol{\gamma}_{\mathcal{S}}) \leftarrow$ Compute Ridge coeff. with (8)

  $\nabla\mathcal{L}_{\mathcal{S}}(\boldsymbol{\gamma}_{\mathcal{S}}) \leftarrow$ Compute loss gradient of (8)

  $\boldsymbol{\gamma}_{\mathcal{S}} \leftarrow \boldsymbol{\gamma}_{\mathcal{S}} - \xi_t \nabla\mathcal{L}_{\mathcal{S}}(\boldsymbol{\gamma}_{\mathcal{S}})$

  $t \leftarrow t + 1$

end

return $\boldsymbol{\beta}^{\star}_{\mathcal{S}}(\boldsymbol{\gamma}^{\star}_{\mathcal{S}})$

Algorithm 2: Test-Time GOPSA

Input: $\{\boldsymbol{\Sigma}_{\mathcal{T},i}\}_{i=1}^{N_{\mathcal{T}}}$, mean outcome value $\bar{y}_{\mathcal{T}}$, initialization of $\gamma_{\mathcal{T}}$, trained Ridge coeff. $\boldsymbol{\beta}^{\star}_{\mathcal{S}}(\boldsymbol{\gamma}^{\star}_{\mathcal{S}})$, step-sizes $\{\xi_t\}_{t\geq 1}$

$\overline{\boldsymbol{\Sigma}}_{\mathcal{T}} \leftarrow$ Riemannian mean of $\{\boldsymbol{\Sigma}_{\mathcal{T},i}\}_{i=1}^{N_{\mathcal{T}}}$

$t \leftarrow 1$

while *not converged* **do**

  $\boldsymbol{Z}_{\mathcal{T}}(\gamma_{\mathcal{T}}) \leftarrow$ Compute features with (9)

  $\mathcal{L}'_{\mathcal{T}}(\gamma_{\mathcal{T}}) \leftarrow$ Compute loss derivative of (10)

  $\gamma_{\mathcal{T}} \leftarrow \gamma_{\mathcal{T}} - \xi_t \mathcal{L}'_{\mathcal{T}}(\gamma_{\mathcal{T}})$

  $t \leftarrow t + 1$

end

$\widehat{\boldsymbol{y}}_{\mathcal{T}} \leftarrow \boldsymbol{Z}_{\mathcal{T}}(\gamma^{\star}_{\mathcal{T}})\boldsymbol{\beta}^{\star}_{\mathcal{S}}(\boldsymbol{\gamma}^{\star}_{\mathcal{S}})$

return $\widehat{\boldsymbol{y}}_{\mathcal{T}}$

3.1 Train-time

GOPSA aims to learn simultaneously features from (4) and a regression model. To do so, we solve the following optimization problem

$$\underset{\substack{\boldsymbol{\beta}_{\mathcal{S}} \in \mathbb{R}^{d(d+1)/2} \\ \boldsymbol{\alpha}_{\mathcal{S}} \in [0,1]^K}}{\text{minimize}} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \left(y_{k,i} - \boldsymbol{\beta}_{\mathcal{S}}^{\top} \phi\left(\boldsymbol{\Sigma}_{k,i}, \overline{\boldsymbol{\Sigma}}_k, \alpha_k\right)\right)^2 \tag{5}$$

with $\boldsymbol{\alpha}_{\mathcal{S}} = [\alpha_1, \ldots, \alpha_K]^{\top}$. This cost function is decomposed into three key aspects. First, covariance matrices undergo parallel transported using Lemma 2.1 to account for shifts between domains. Second, they are vectorized, and a linear regression predicts the output variable from these vectorized features. Third, the coefficients of the linear regression $\boldsymbol{\beta}_{\mathcal{S}}$ and the $\boldsymbol{\alpha}_{\mathcal{S}}$ are learned jointly so that the predictor is adapted to the parallel transport and reciprocally. Besides, to enforce the constraint on $\boldsymbol{\alpha}_{\mathcal{S}}$, we re-parameterize it using the sigmoid function, which defines a bijection between $\mathbb{R}$ and $(0, 1)$, thereby ensuring that the resulting $\boldsymbol{\alpha}_{\mathcal{S}}$ values lie within the desired range: $\alpha_k = \sigma(\gamma_k) \triangleq (1 + \exp(-\gamma_k))^{-1}$. Thus, the constrained problem (5) can be formulated as the following unconstrained optimization problem

$$\underset{\substack{\boldsymbol{\beta}_{\mathcal{S}} \in \mathbb{R}^{d(d+1)/2} \\ \boldsymbol{\gamma}_{\mathcal{S}} \in \mathbb{R}^K}}{\text{minimize}} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \left(y_{k,i} - \boldsymbol{\beta}_{\mathcal{S}}^{\top} \phi\left(\boldsymbol{\Sigma}_{k,i}, \overline{\boldsymbol{\Sigma}}_k, \sigma(\gamma_k)\right)\right)^2, \tag{6}$$

with $\boldsymbol{\gamma}_{\mathcal{S}} = [\gamma_1, \ldots, \gamma_K]^{\top}$. Let us define the matrix $\boldsymbol{Z}_{\mathcal{S}}(\boldsymbol{\gamma}) \in \mathbb{R}^{N_{\mathcal{S}} \times d(d+1)/2}$, with $N_{\mathcal{S}} = \sum_{k=1}^{K} N_k$, as the concatenation of the source data, where each row corresponds to a feature vector:

$$\boldsymbol{Z}_{\mathcal{S}}(\boldsymbol{\gamma}) = \left[\phi\left(\boldsymbol{\Sigma}_{1,1}, \overline{\boldsymbol{\Sigma}}_1, \sigma(\gamma_1)\right), \ldots, \phi\left(\boldsymbol{\Sigma}_{K,N_K}, \overline{\boldsymbol{\Sigma}}_K, \sigma(\gamma_K)\right)\right]^{\top}. \tag{7}$$

In the same manner, the source labels are concatenated to $\boldsymbol{y}_{\mathcal{S}} = [y_{1,1}, \ldots, y_{K,N_K}]^{\top} \in \mathbb{R}^{N_{\mathcal{S}}}$. Given a fixed $\boldsymbol{\gamma}_{\mathcal{S}}$, the problem (6) is solved with the ordinary least squares estimator. In practice, we choose to regularize the estimation of the linear regression with a Ridge penalty. Thus, (6) is rewritten as

$$\begin{aligned} \boldsymbol{\gamma}^{\star}_{\mathcal{S}} \triangleq \underset{\boldsymbol{\gamma} \in \mathbb{R}^K}{\arg\min} \left\{ \mathcal{L}_{\mathcal{S}}(\boldsymbol{\gamma}) \triangleq \|\boldsymbol{y}_{\mathcal{S}} - \boldsymbol{Z}_{\mathcal{S}}(\boldsymbol{\gamma})\boldsymbol{\beta}^{\star}_{\mathcal{S}}(\boldsymbol{\gamma})\|_2^2 \right\} \\ \text{subject to} \quad \boldsymbol{\beta}^{\star}_{\mathcal{S}}(\boldsymbol{\gamma}) \triangleq \boldsymbol{Z}_{\mathcal{S}}(\boldsymbol{\gamma})^{\top}(\lambda \boldsymbol{I}_N + \boldsymbol{Z}_{\mathcal{S}}(\boldsymbol{\gamma})\boldsymbol{Z}_{\mathcal{S}}(\boldsymbol{\gamma})^{\top})^{-1}\boldsymbol{y}_{\mathcal{S}} \ , \end{aligned} \tag{8}$$

where $\boldsymbol{\beta}^{\star}_{\mathcal{S}}(\boldsymbol{\gamma}) \in \mathbb{R}^{d(d+1)/2}$ are the Ridge estimated coefficients given a fixed $\boldsymbol{\gamma}$ and $\lambda > 0$ is the regularization hyperparameter. The problem (8) is efficiently solved with any gradient-based solver [39].

The train-time of GOPSA is summarized in Algorithm 1. The proposed training algorithm begins by calculating the Riemannian mean of covariance matrices for each domain k. It then iteratively optimizes the parameters $\boldsymbol{\gamma}_{\mathcal{S}}$ by computing the feature matrix (7), determining Ridge regression coefficients (8), and updating $\boldsymbol{\gamma}_{\mathcal{S}}$ using a gradient descent step on the loss function (8) until convergence. The output result is the optimized Ridge regression coefficients. For clarity of presentation,

https://doi.org/10.52202/079017-1033

Algorithm 1 employs a gradient descent. In practice, we use L-BFGS and obtain the gradient using automatic differentiation through the Ridge solution that is plugged into the loss in (8).

3.2 Test-time

At test-time, we now have a fitted linear model on source data with coefficients $\boldsymbol{\beta}^\star_{\mathcal{S}}(\boldsymbol{\gamma}^\star_{\mathcal{S}})$. The goal is to adapt a new target domain $(\boldsymbol{\Sigma}_{\mathcal{T},i})_{i=1}^{N_\mathcal{T}}$ for which the average outcome $\bar{y}_\mathcal{T}$ is assumed to be known. First, let us define the matrix $\boldsymbol{Z}_\mathcal{T}(\gamma) \in \mathbb{R}^{N_\mathcal{T} \times d(d+1)/2}$ as the concatenation of the target data

$$\boldsymbol{Z}_\mathcal{T}(\gamma) = \left[\phi\left(\boldsymbol{\Sigma}_{\mathcal{T},1}, \overline{\boldsymbol{\Sigma}}_\mathcal{T}, \sigma(\gamma)\right), \ldots, \phi\left(\boldsymbol{\Sigma}_{\mathcal{T},N_\mathcal{T}}, \overline{\boldsymbol{\Sigma}}_\mathcal{T}, \sigma(\gamma)\right)\right]^\top . \tag{9}$$

Then, GOPSA adapts to this new target domain by minimizing the error between $\bar{y}_\mathcal{T}$ and its estimation computed with the fitted linear model. This minimization is performed with respect to $\gamma_\mathcal{T} \in \mathbb{R}$ that parametrizes the parallel transport of the target domain, i.e.,

$$\gamma^\star_\mathcal{T} = \underset{\gamma \in \mathbb{R}}{\arg\min} \left\{ \mathcal{L}_\mathcal{T}(\gamma) \triangleq \left(\bar{y}_\mathcal{T} - \frac{1}{N_\mathcal{T}} \mathbf{1}_{N_\mathcal{T}}^\top \boldsymbol{Z}_\mathcal{T}(\gamma) \boldsymbol{\beta}^\star_{\mathcal{S}}(\boldsymbol{\gamma}^\star_{\mathcal{S}}) \right)^2 \right\} . \tag{10}$$

Finally, the predictions on the target domain are

$$\widehat{\boldsymbol{y}}_\mathcal{T} = \boldsymbol{Z}_\mathcal{T}(\gamma^\star_\mathcal{T}) \boldsymbol{\beta}^\star_{\mathcal{S}}(\boldsymbol{\gamma}^\star_{\mathcal{S}}) \in \mathbb{R}^{N_\mathcal{T}} . \tag{11}$$

The test-time procedure of GOPSA is summarized in Algorithm 2. The algorithm begins by calculating the Riemannian mean of the target covariance matrices $\{\boldsymbol{\Sigma}_{\mathcal{T},i}\}_{i=1}^{N_\mathcal{T}}$. It then iteratively optimizes the parameter $\gamma_\mathcal{T}$ by computing the feature matrix (9), the derivative of the loss function (10), and updating $\gamma_\mathcal{T}$ using a gradient descent step until convergence. The algorithm determines the estimated target outcomes, $\widehat{\boldsymbol{y}}_\mathcal{T}$, by using the optimized $\gamma^\star_\mathcal{T}$ on the feature matrix, combined with the pre-trained regression coefficients $\boldsymbol{\beta}^\star_{\mathcal{S}}(\boldsymbol{\gamma}^\star_{\mathcal{S}})$. The output result is the predicted target outcomes $\widehat{\boldsymbol{y}}_\mathcal{T}$. It should be noted that, once again, for clarity of presentation, Algorithm 2 employs a gradient descent, but other derivative-based optimization methods can be used.

4 Empirical benchmarks

In this section, we built empirical benchmark to evaluate the performance of GOPSA. We first present the simulated data that we used to illustrate the relevance of our method when there is a joint distribution shift of the data and the labels. Then, we present the EEG dataset that we used to evaluate the performance of GOPSA with real data from different recording sites. Finally, we present the baseline methods that are compared with GOPSA.

Simulated data To generate simulated data, we used the generative model described in [47, 48, 36]. The data follow the classical instantaneous mixing model:

$$\boldsymbol{x}_i(t) = \boldsymbol{A}\boldsymbol{s}_i(t) \tag{12}$$

where $\boldsymbol{x}_i(t) \in \mathbb{R}^d$ are the observed time-series, $\boldsymbol{s}_i(t) \in \mathbb{R}^d$ are the underlying signal of the neural generators and $\boldsymbol{A}$ is the mixing matrix whose columns are the observed spatial patterns of the neural generators. Furthermore, we assume that y follow a log-linear model:

$$y_i = \beta_0 + \sum_{\ell=1}^{d} \beta_\ell \log(p_{\ell i}) \tag{13}$$

where $p_{\ell i} > 0$ is the variance of the ℓ-th element of the underlying signal $\boldsymbol{s}_i(t)$ as introduced in [47, 48, 36]. From this, we generate domains (source and target) by applying shifts on X and y. To do so, we introduced a per-domain shift in the data distribution by applying an affine transformation to the covariance matrices $\boldsymbol{\Sigma}_i \triangleq \mathbb{E}[\boldsymbol{x}_i(t)\boldsymbol{x}_i(t)^\top]$:

$$\boldsymbol{\Sigma}_i \mapsto \boldsymbol{B}_k^\xi \boldsymbol{\Sigma}_i \boldsymbol{B}_k^\xi \tag{14}$$

with $\boldsymbol{B}_k \in \mathbb{S}_d^{++}$ and $\xi \geq 0$ controlling the amplitude of the shift. Then, we shifted the label distribution by modifying the variance of the underlying signal $p_{\ell i}$:

$$p_{\ell i} \mapsto p_{\ell i}^{1+k\xi} \tag{15}$$

with $\xi \geq 0$ still controlling the amplitude of the shift. Thus, the distribution of y is shifted per domain because of the log-linear relationship of (13). It should be noted that $\boldsymbol{\beta}$ is kept constant across domains.

HarMNqEEG dataset The HarMNqEEG dataset [33] was used for our numerical experiments. This dataset includes EEG recordings collected from 1564 participants across 14 different study sites, distributed across 9 countries. In our analysis, we consider each study site as a distinct domain. Appendix A.5 provides detailed demographic information. The EEG data were recorded with the same montage of 19 channels of the 10/20 International Electrodes Positioning System. The dataset provides pre-computed cross-spectral tensors for each participant rather than raw data, and anonymized metadata including the age and the sex of the participants. More precisely, the shared data consists of cross-spectral matrices with a frequency range of 1.17 Hz to 19.14 Hz, sampled at a resolution of 0.39 Hz. A standardized recording protocol was enforced to ensure the consistency across EEG recording of the dataset. In addition to recording constraints, this protocol included artifact cleaning procedures. The cross-spectrum were computed using Bartlett's method (See Appendix A.3). Our pre-processing steps were guided by the pre-processing pipeline outlined in [33]. First, we performed a common average reference (CAR) on all cross-spectrum (See Appendix A.3) as different EEG references were used across domains. Subsequently, we extracted the real part of the cross-spectral tensor to obtain co-spectrum tensors containing frequency-specific covariance estimates along the frequency spectrum. Due to the linear dependence between channels introduced by the CAR, the covariance matrices are rank deficient. To address this, we applied a shrinkage regularization with a coefficient of 10^{-5} to the data. Additionally, we implemented a global-scale factor (GSF) correction, which compensates for amplitude variations between EEG recordings by scaling the covariance matrices with a subject-specific factor [24, 33] (See Appendix A.3). Following these pre-processing steps, we obtained a set of 49 covariance matrices for each EEG recording, with each matrix corresponding to a specific frequency bin of the EEG signal. This pre-processed co-spectrum served as the input data for our domain adaptation study.

Performance evaluation and hyperparameter selection To evaluate the performance of the compared methods, we conducted experiments across several combinations of source and target sites. We selected source domains such that the union distribution of their predictive variable y encompasses a broad age range. All remaining sites were assigned as target domains. For each source-target combination we performed a stratified shuffle split approach with 100 repetitions on the target data. Stratification was based on the recording sites to ensure that each split contained a balanced proportion of participants from each site. The regularization parameter λ in Ridge regression was selected with a nested cross-validation (grid search) over a logarithmic grid of values from 10^{-1} to 10^{5}. To evaluate the benefit of `GOPSA`, we compared it against four baselines. Detailed mathematical formulations of these baselines can be found in Appendix A.4. For each baseline method, the regression task was performed with Ridge regression.

Domain-aware dummy model (`DO Dummy`) As `GOPSA` requires access to the mean $\bar{y}_k$ of each domain, we used a domain-aware dummy model predicting always the mean $\bar{y}_k$ of each domain.

No re-center / No domain adaptation (`No DA`) This second baseline method involves applying the regression pipeline outlined in [47, 48] without any re-centering. In this setup, all covariance matrices are projected to the tangent space at the source geometric mean $\overline{\boldsymbol{\Sigma}}$ computed from all source points, no matter their recording sites.

Re-center to a common reference point (`Re-center` and `Re-scale`) As introduced in Section 2, a common transfer learning approach is a Riemannian re-centering of all domains to a common point on the manifold [62, 36]. This baseline thus correspond to re-centering each domain k, source and target, independently by whitening them by their respective geometric mean $\overline{\boldsymbol{\Sigma}}_k$. An extension of this approach is to perform a Riemannian re-scaling of all domains to a common dispersion, as presented in [45, 36].

Domain-aware intercept (`DO Intercept`) This method consists in fitting one intercept β_0 per domain. In practice since we assume to know $\bar{y}_{\mathcal{T}}$, we correct the predicted values so that their mean is equal to $\bar{y}_{\mathcal{T}}$. This approach is in line with defining mixed-effects models on the Riemannian manifold [33].

Deep learning (`GREEN`) The `GREEN` model [40] is a deep-learning architecture tailored for EEG applications like age prediction. Since the HarMNqEEG dataset consists of covariance matrices, we

https://doi.org/10.52202/079017-1033

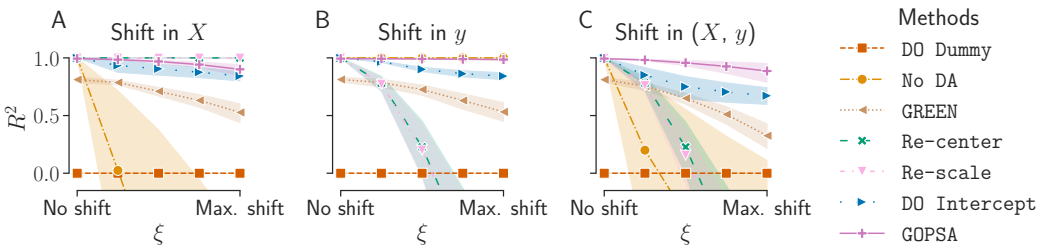

Figure 2: R^2 **scores ↑ for different methods on simulated data.** Performance is measured across 5 source domains and 1 target domain, with shifts controlled by ξ (0 to maximum). Data are generated 100 times, with 5 sensors and 300 covariance matrices per domain. The target domain is randomly selected between the 6 domains generated as presented in Section 4, with the remaining domains used as sources. **(A)** A shift is applied on the covariance matrices following (14). **(B)** A shift is applied on the variances following (15). **(C)** Both shifts from (14) and (15) are applied simultaneously.

used the 'G2' variant of `GREEN`, which starts at the covariance matrices level and includes pooling layers. This variant is designed for SPD matrices, making it an SPD network [26]. Although `GREEN` has been evaluated on multiple datasets for various predictive tasks, it has not yet been applied in a domain adaptation context and does not include an adaptation layer.

We applied the domain-adaptation methods independently to each of the 49 frequency bins, resulting in 49 geometric means per domain, except for `GREEN`, which processes all frequency bands simultaneously. $\bar{y}_{\mathcal{T}}$ of each domain was estimated on target splits (50% of the data) that do not overlap with the evaluation target splits (50% remaining). Statistical inference for model comparisons was implemented with a corrected t-test following [37]. Experiments with 100 repetitions and all site combinations have been run on a standard Slurm cluster for 12 hours with 250 CPU cores.

5 Results

Simulated data Figure 2 presents the results of simulated experiments where shifts are applied on either X, y, or both (X, y) as presented in Section 4. All methods were evaluated in three simulation scenarios: shift in X only, shift in y only, and joint shift in X and y. The intensity of the shift was controlled by ξ in all scenarios. If there is no shift in X, we observe that `No DA` perfectly estimates the y because the log-linear model is easily estimated across domains even when the y distribution changes (Figure 2 B). The performance of `No DA` however drops when a shift in X is introduced (Figure 2 A and C). `Re-center` and `Re-scale` led to the same results as no scaling shift was applied in the simulation. Both were able to correct the shift in X, but performed poorly when a shift in y was added (Figure 2 B and C). `GREEN` notably showed consistant performance across all scenarios, and was relatively resistant to both types of shifts given it is not designed for domain adaptation. `DO Intercept` and `GOPSA` showed the best performance across all scenarios, with an advantage for `GOPSA`. The interest of `GOPSA` is to estimate this log-linear model with shifts in (X, y) per domain (Figure 2 C) which other methods were not able to do. These experiments demonstrate the efficiency of the proposed method in estimating shifts in X between domains even in the presence of a shift in y, contrary to the baseline methods. Theoretically, based on the generative model of the simulated data, the data X and outcome y are linked by a log-linear relationship. This implies that, knowing the shift in X for the target domain, predictions can be made even when y distributions do not overlap between the source and target. Since `GOPSA` estimates the target shift in X by minimizing $(\bar{y} - \text{mean}(\hat{y}_i))^2$, it is capable of handling such scenarios.

HarMNqEEG data We computed benchmarks for five combinations of source sites and we displayed the results for the three metrics selected for performance evaluation, each colored box representing one method (Figure 3). A min-max normalization was applied to each site combinations separately across methods. We first conducted model comparisons in terms of absolute performance across all baselines (**A**). `No DA`, without domain specific re-centering, performed worse than `DO Dummy` in terms of R^2 score and MAE. `Re-center` and `Re-scale` led to lower performances across all metrics, which can be expected as the Riemannian mean is correlated with age in our problem setting Figure 1. Eventhough its architecture does not include an adaptation layer, `GREEN`

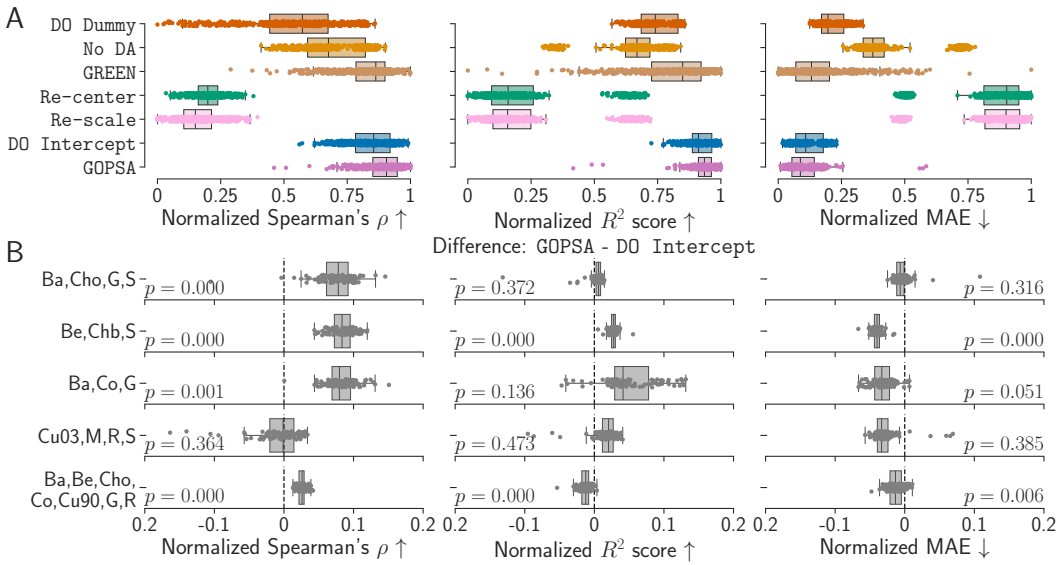

Figure 3: **Normalized performance of the different methods on several source-target combinations for three metrics:** Spearman's $\rho \uparrow$ (left), R^2 score $\uparrow$ (middle) and Mean Absolute Error $\downarrow$ (right). As a large variability in the score values was present between the site combinations, we applied a min-max normalization per combination to set the minimum score across all methods to 0 and the maximum score to 1. (**A**) Boxplot of the concatenated results for the three normalized scores. One point corresponds to one split of one site combination. (**B**) Boxplots of the difference between the normalized scores of `GOPSA` and `DO Intercept`. A row corresponds to one site combination, one point corresponds to one split. For each plot, the associated results of Nadeau's & Bengio's corrected t-test [37] are displayed. A p-value lower than 0.05 indicates a significant difference between the two methods. Ba: Barbados, Be: Bern, Chb: CHBMP (Cuba), Co: Columbia, Cho: Chongqing, Cu03: Cuba2003, Cu90: Cuba90, G: Germany, M: Malaysia, R: Russia, S: Switzerland

reached better performance than the previous methods mentionned, but lacked consistency across site combinations and metrics with large variance especially for the R^2 score and MAE. For all scores, `DO Intercept` and `GOPSA` reached the best average performance with lower variance. A version of Figure 3 **A** without normalization is presented in Appendix A.7. As `DO intercept` and `GOPSA` showed overlapping performance distributions, we investigated their paired split-wise (non-rescaled) score differences (**B**). The site-specific differences of `GOPSA` scores minus `DO Intercept` are displayed with their associated p-values. For one site combination (Ba,Be,Cho,Co,Cu90,G,R), `DO Intercept` yielded higher R^2 scores, and no significant difference was found between the two methods for Ba,Co,G. Similarly, no significant difference was observed on Spearman's ρ results for Cu03,M,R,S. Overall, `GOPSA` significantly outperformed `DO Intercept` in five site combinations for MAE, four for Spearman's ρ and three for R^2 score. Detailed results for each source-target combination are presented in Appendix A.6 for Spearman's ρ, R^2 score, and MAE. The bottom rows correspond to the mean performance of each method of all site combinations, and their average standard deviation (see Appendix A.8 for associated boxplots). We expected `GOPSA` to outperform the baseline methods (e.g. `DO Intercept`) whenever joint (X, y) shifts occur. In our experimental benchmark, `GOPSA` significantly outperformed the baseline methods in some site combinations, but not all. This allows us to assume that not all site combinations show joint shifts.

Model inspection Next, we investigated the impact of the different re-centering approaches on the data Figure 4. Power spectrum densities (PSDs) were computed as the mean across sensors of the diagonals of the covariance matrices Riemannian mean for each site combination after `No DA`, `Re-center` and `GOPSA` (**A**). PSDs for `No DA` display the initial variability between sites without recentering (cf. Figure 1). `Re-center` resulted in flat PSDs because all data were re-centered to the identity. PSDs produced by `GOPSA` are flattened and more similar across sites compared to `No DA` without removing too much information, unlike the un-effective `Re-center method` (cf. Figure 3). The alpha values are inspected as a function of the site mean age (**B**). `Re-center` leads to alpha values all equal to one as all sites are re-centered to the identity. For `GOPSA`, we observed a linear

https://doi.org/10.52202/079017-1033

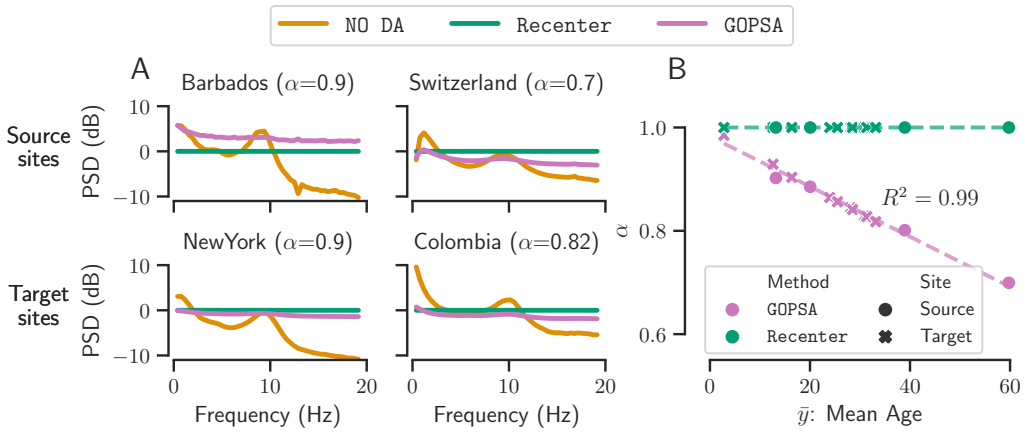

Figure 4: **Model inspection of `GOPSA` versus `No DA` and `Re-center`.** Power Spectral Densities (PSDs) and α values were computed on the source sites Barbados, Chongqing, Germany, and Switzerland. The remaining sites were used as target domains. (**A**) Mean PSDs computed across sensors for `No DA`, `Recenter` and `GOPSA` on two source (Barbados and Switzerland) and two target (New York and Columbia) sites. (**B**) α values versus site's mean age for `Re-center` and `GOPSA`. One point corresponds to one site. The coefficient of determination is reported for the `GOPSA` method.

relationship between alpha and the sites' mean age ($R^2 = 0.99$). This is a direct consequence of the optimization process in `GOPSA`, which thus can be regarded a geodesic intercept in a mixed-effects model. Overall, `GOPSA` effectively re-centered sites with younger participants closer to the identity matrix. Re-centering sites around a common point helped reduce the shift in X, while not placing all sites at the exact same reference point helped manage the shift in y, hence preserving the statistical associations between X and y.

6 Conclusion

We proposed a novel multi-source domain adaptation approach that adapts shifts in X and y simultaneously by learning jointly a domain specific re-centering operator and the regression model. `GOPSA` was specifically developed to handle joint shifts in the data distribution and the outcome distribution, as illustrated by the simulations in Figure 2. `GOPSA` is a test-time method that does not require to retrain a model when a new domain is presented. `GOPSA` achieved state-of-the-art performance on the HarMNqEEG [33] dataset with EEG from 14 recording sites and over 1500 participants. Our benchmarks showed a significant gain in performance for three different metrics in a majority of site combinations compared to baseline methods. `GOPSA` can thus be used by researchers as a decision rule to infer the presence of joint shifts and, hence, serve as a tool for data exploration and model interpretation. While we focused on shallow regression models, the implementation of `GOPSA` using PyTorch readily supports its inclusion in more complex Riemannian deep learning models [26, 56, 11, 40, 31]. This direction seems promising given our observation that GREEN – a simple deep net combining Riemannian computation with a fully connected layer - already possessed some intrinsic robustness to data shifts. This may point at the capacity of the fully-connected layer to provide additional non-linear transformations that can accommodate the data-generating scenario in which continuous log-linear generators are modified in a discrete manner by site factors. More generally, it emphasizes the potential of complex nonlinear methods for domain adaptation, in line with a recent study on the same dataset reporting positive generalization results using a kernel method [28]. Furthermore, although this work specifically addresses age prediction, the methodology is applicable to a broader range of regression analyses. While `GOPSA` necessitates knowledge or estimability of the average $\bar{y}$ per domain, this requirement aligns with that of mixed-effects models [16, 25, 61], which are extensively employed in biomedical statistics. By combining mixed-effects modeling with Riemannian geometry for EEG, `GOPSA` opens up various applications at the interface between machine learning and biostatistics, such as, biomarker exploration in large multicenter clinical trials [46, 52, 53].

Acknowledgment

This work was supported by grants ANR-20-CHIA-0016 and ANR-20-IADJ-0002 to AG while at Inria, ANR-20-THIA-0013 to AM, ANR-22-CE33-0015-01 and ANR-17-CONV-0003 operated by LISN to SC and ANR-22-PESN-0012 to AC under the France 2030 program, all managed by the Agence Nationale de la Recherche (ANR) Competing interests: DE is a full-time employee of F. Hoffmann-La Roche Ltd.

Numerical computation was enabled by the scientific Python ecosystem: Matplotlib [27], Scikit-learn [42], Numpy [21], Scipy [54], PyTorch [41] PyRiemann [3], MNE [19] and SKADA [18].

This work was conducted at Inria, AG is presently employed by Meta Platforms. All the datasets used for this work were accessed and processed on the Inria compute infrastructure.

References

[1] Benyamin Allahgholizadeh Haghi, Spencer Kellis, Sahil Shah, Maitreyi Ashok, Luke Bashford, Daniel Kramer, Brian Lee, Charles Liu, Richard Andersen, and Azita Emami. Deep multi-state dynamic recurrent neural networks operating on wavelet based neural features for robust brain machine interfaces. *Advances in Neural Information Processing Systems*, 32, 2019.

[2] Gopala K. Anumanchipalli, Josh Chartier, and Edward F. Chang. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, 2019.

[3] Alexandre Barachant, Quentin Barthélemy, Jean-Rémi King, Alexandre Gramfort, Sylvain Chevallier, Pedro L. C. Rodrigues, Emanuele Olivetti, Vladislav Goncharenko, Gabriel Wagner vom Berg, Ghiles Reguig, Arthur Lebeurrier, Erik Bjäreholt, Maria Sayu Yamamoto, Pierre Clisson, and Marie-Constance Corsi. pyriemann/pyriemann: v0.3, July 2022.

[4] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Classification of covariance matrices using a riemannian-based kernel for BCI applications. *Neurocomputing*, 112:172–178, 2013.

[5] Alexandre Barachant, Stphane Bonnet, Marco Congedo, and Christian Jutten. Common spatial pattern revisited by riemannian geometry. In *2010 IEEE international workshop on multimedia signal processing*, pages 472–476. IEEE, 2010.

[6] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass Brain–Computer Interface Classification by Riemannian Geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2012.

[7] Philipp Bomatter, Joseph Paillard, Pilar Garces, Jörg Hipp, and Denis-Alexander Engemann. Machine learning of brain-specific biomarkers from EEG. *Ebiomedicine*, 106, 2024.

[8] Clément Bonet, Benoît Malézieux, Alain Rakotomamonjy, Lucas Drumetz, Thomas Moreau, Matthieu Kowalski, and Nicolas Courty. Sliced-Wasserstein on Symmetric Positive Definite Matrices for M/EEG Signals. In *Proceedings of the 40th International Conference on Machine Learning*, pages 2777–2805. PMLR, July 2023.

[9] Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

[10] György Buzsáki and Kenji Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264–278, 2014.

[11] Igor Carrara, Bruno Aristimunha, Marie-Constance Corsi, Raphael Y de Camargo, Sylvain Chevallier, and Théodore Papadopoulo. Geometric neural network based on phase space for BCI decoding. *arXiv preprint arXiv:2403.05645*, 2024.

[12] Jérôme Dockès, Gaël Varoquaux, and Jean-Baptiste Poline. Preventing dataset shift from breaking machine-learning biomarkers. *GigaScience*, 10(9):giab055, 2021.

[13] Denis A. Engemann, Apolline Mellot, Richard Höchenberger, Hubert Banville, David Sabbagh, Lukas Gemein, Tonio Ball, and Alexandre Gramfort. A reusable benchmark of brain-age prediction from M/EEG resting-state signals. *NeuroImage*, 262:119521, November 2022.

[14] Denis A Engemann, Federico Raimondo, Jean-Rémi King, Benjamin Rohaut, Gilles Louppe, Frédéric Faugeras, Jitka Annen, Helena Cassol, Olivia Gosseries, Diego Fernandez-Slezak, et al.

 https://doi.org/10.52202/079017-1033

Robust EEG-based cross-site and cross-protocol classification of states of consciousness. *Brain*, 141(11):3179–3192, 2018.

[15] Dylan Forenzo, Hao Zhu, Jenn Shanahan, Jaehyun Lim, and Bin He. Continuous tracking using deep learning-based decoding for noninvasive brain-computer interface. *PNAS nexus*, 3(4):pgae145, 2024.

[16] Andrew Gelman. Multilevel (hierarchical) modeling: what it can and cannot do. *Technometrics*, 48(3):432–435, 2006.

[17] Lukas AW Gemein, Robin T Schirrmeister, Patryk Chrabąszcz, Daniel Wilson, Joschka Boedecker, Andreas Schulze-Bonhage, Frank Hutter, and Tonio Ball. Machine-learning-based diagnostics of EEG pathology. *NeuroImage*, 220:117021, 2020.

[18] Théo Gnassounou, Oleksii Kachaiev, Rémi Flamary, Antoine Collas, Yanis Lalou, Antoine de Mathelin, Alexandre Gramfort, Ruben Bueno, Florent Michel, Apolline Mellot, Virginie Loison, Ambroise Odonnat, and Thomas Moreau. Skada : Scikit adaptation, 7 2024.

[19] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7(267):1–13, 2013.

[20] Haris Hakeem, Wei Feng, Zhibin Chen, Jiun Choong, Martin J Brodie, Si-Lei Fong, Kheng-Seang Lim, Junhong Wu, Xuefeng Wang, Nicholas Lawn, et al. Development and validation of a deep learning model for predicting treatment response in patients with newly diagnosed epilepsy. *JAMA neurology*, 79(10):986–996, 2022.

[21] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T.E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[22] Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. Domain adaptation for time series under feature and label shifts. In *International Conference on Machine Learning*, pages 12746–12774. PMLR, 2023.

[23] Elisabeth R M Heremans, Huy Phan, Pascal Borzée, Bertien Buyse, Dries Testelmans, and Maarten De Vos. From unsupervised to semi-supervised adversarial domain adaptation in electroencephalography-based sleep staging. *Journal of Neural Engineering*, 19(3):036044, jun 2022.

[24] JL Hernández, P Valdés, R Biscay, T Virues, S Szava, J Bosch, A Riquenes, and I Clark. A global scale factor in brain topography. *International journal of neuroscience*, 76(3-4):267–278, 1994.

[25] Joop Hox. Multilevel modeling: When and why. In *Classification, data analysis, and data highways: proceedings of the 21st Annual Conference of the Gesellschaft für Klassifikation eV, University of Potsdam, March 12–14, 1997*, pages 147–154. Springer, 1998.

[26] Zhiwu Huang and Luc Van Gool. A riemannian network for SPD matrix learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[27] J.D. Hunter. Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[28] Cecilia Jarne, Ben Griffin, and Diego Vidaurre. Predicting subject traits from brain spectral signatures: an application to brain ageing. *bioRxiv*, pages 2023–11, 2023.

[29] Haiteng Jiang, Peiyin Chen, Zhaohong Sun, Chengqian Liang, Rui Xue, Liansheng Zhao, Qiang Wang, Xiaojing Li, Wei Deng, Zhongke Gao, et al. Assisting schizophrenia diagnosis using clinical electroencephalography and interpretable graph neural networks: a real-world and cross-site study. *Neuropsychopharmacology*, 48(13):1920–1930, 2023.

[30] Hyunwoo J. Kim, Nagesh Adluru, Heemanshu Suri, Baba C. Vemuri, Sterling C. Johnson, and Vikas Singh. Riemannian nonlinear mixed effects models: Analyzing longitudinal deformations in neuroimaging. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5777–5786, 2017.

[31] Reinmar Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe. SPD domain-specific batch normalization to crack interpretable unsupervised domain adaptation in EEG. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 6219–6235. Curran Associates, Inc., 2022.

[32] Dianqi Li, Yizhe Zhang, Zhe Gan, Yu Cheng, Chris Brockett, Bill Dolan, and Ming-Ting Sun. Domain adaptive text style transfer. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3304–3313, Hong Kong, China, November 2019. Association for Computational Linguistics.

[33] Min Li, Ying Wang, Carlos Lopez-Naranjo, Shiang Hu, Ronaldo César García Reyes, Deirel Paz-Linares, Ariosky Areces-Gonzalez, Aini Ismafairus Abd Hamid, Alan C Evans, Alexander N Savostyanov, et al. Harmonized-Multinational qEEG norms (HarMNqEEG). *NeuroImage*, 256:119190, 2022.

[34] Yitong Li, Michael Murias, Samantha Major, Geraldine Dawson, and David Carlson. On target shift in adversarial domain adaptation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 616–625. PMLR, 2019.

[35] Carlos Lopez Naranjo, Fuleah Abdul Razzaq, Min Li, Ying Wang, Jorge F Bosch-Bayard, Martin A Lindquist, Anisleidy Gonzalez Mitjans, Ronaldo Garcia, Arielle G Rabinowitz, Simon G Anderson, et al. EEG functional connectivity as a riemannian mediator: An application to malnutrition and cognition. *Human Brain Mapping*, 45(7):e26698, 2024.

[36] Apolline Mellot, Antoine Collas, Pedro LC Rodrigues, Denis Engemann, and Alexandre Gramfort. Harmonizing and aligning M/EEG datasets with covariance-based techniques to enhance predictive regression modeling. *Imaging Neuroscience*, 1:1–23, 2023.

[37] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Advances in neural information processing systems*, 12, 1999.

[38] Chuong H Nguyen, George K Karavas, and Panagiotis Artemiadis. Inferring imagined speech using EEG signals: a new approach using riemannian manifold features. *Journal of neural engineering*, 15(1):016002, 2017.

[39] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[40] Joseph Paillard, Joerg F Hipp, and Denis A Engemann. GREEN: a lightweight architecture using learnable wavelets and riemannian geometry for biomarker exploration. *bioRxiv*, pages 2024–05, 2024.

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. Curran Associates, Inc., 2019.

[42] 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*, 12:2825–2830, 2011.

[43] Peizhen Peng, Liping Xie, Kanjian Zhang, Jinxia Zhang, Lu Yang, and Haikun Wei. Domain adaptation for epileptic EEG classification using adversarial learning and riemannian manifold. *Biomedical Signal Processing and Control*, 75:103555, 2022.

[44] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian Framework for Tensor Computing. *International Journal of Computer Vision*, 66(1):41–66, January 2006.

[45] Pedro Luiz Coelho Rodrigues, Christian Jutten, and Marco Congedo. Riemannian Procrustes Analysis: Transfer Learning for Brain–Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 66(8):2390–2401, August 2019.

[46] Andrea O Rossetti, Kaspar Schindler, Raoul Sutter, Stephan Rüegg, Frédéric Zubler, Jan Novy, Mauro Oddo, Loane Warpelin-Decrausaz, and Vincent Alvarez. Continuous vs routine electroencephalogram in critically ill adults with altered consciousness and no recent seizure: a multicenter randomized clinical trial. *JAMA neurology*, 77(10):1225–1232, 2020.

 https://doi.org/10.52202/079017-1033

[47] David Sabbagh, Pierre Ablin, Gaël Varoquaux, Alexandre Gramfort, and Denis A Engemann. Manifold-regression to predict from MEG/EEG brain signals without source modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

[48] David Sabbagh, Pierre Ablin, Gaël Varoquaux, Alexandre Gramfort, and Denis A Engemann. Predictive regression modeling with MEG/EEG: from source power to signals and cognitive states. *NeuroImage*, 222:116893, 2020.

[49] Jean-Baptiste Schiratti, Stéphanie Allassonniere, Olivier Colliot, and Stanley Durrleman. Learning spatiotemporal trajectories from manifold-valued longitudinal data. *Advances in neural information processing systems*, 28, 2015.

[50] Jean-Baptiste Schiratti, Stéphanie Allassonnière, Olivier Colliot, and Stanley Durrleman. A bayesian mixed-effects model to learn trajectories of changes from repeated manifold-valued observations. *Journal of Machine Learning Research*, 18(133):1–33, 2017.

[51] Lene Theil Skovgaard. A Riemannian Geometry of the Multivariate Normal Model. *Scandinavian Journal of Statistics*, 11(4):211–223, 1984. Publisher: [Board of the Foundation of the Scandinavian Journal of Statistics, Wiley].

[52] Paola Vassallo, Jan Novy, Frédéric Zubler, Kaspar Schindler, Vincent Alvarez, Stephan Rüegg, and Andrea O Rossetti. EEG spindles integrity in critical care adults. analysis of a randomized trial. *Acta Neurologica Scandinavica*, 144(6):655–662, 2021.

[53] Paul M Vespa, DaiWai M Olson, Sayona John, Kyle S Hobbs, Kapil Gururangan, Kun Nie, Masoom J Desai, Matthew Markert, Josef Parvizi, Thomas P Bleck, et al. Evaluating the clinical impact of rapid response electroencephalography: the DECIDE multicenter prospective observational clinical study. *Critical care medicine*, 48(9):1249–1257, 2020.

[54] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, J.K. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, I. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[55] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[56] Daniel Wilson, Robin Tibor Schirrmeister, Lukas Alexander Wilhelm Gemein, and Tonio Ball. Deep riemannian networks for EEG decoding. *arXiv preprint arXiv:2212.10426*, 2022.

[57] Jonathan R Wolpaw, Dennis J McFarland, Gregory W Neat, and Catherine A Forneris. An EEG-based brain-computer interface for cursor control. *Electroencephalography and clinical neurophysiology*, 78(3):252–259, 1991.

[58] Wei Wu, Yu Zhang, Jing Jiang, Molly V Lucas, Gregory A Fonzo, Camarin E Rolle, Crystal Cooper, Cherise Chin-Fatt, Noralie Krepel, Carena A Cornelssen, et al. An electroencephalographic signature predicts antidepressant response in major depression. *Nature biotechnology*, 38(4):439–447, 2020.

[59] Or Yair, Felix Dietrich, Ronen Talmon, and Ioannis G. Kevrekidis. Domain Adaptation with Optimal Transport on the Manifold of SPD matrices, July 2020. arXiv:1906.00616 [cs, stat].

[60] Yuzhe Yang, Yuan Yuan, Guo Zhang, Hao Wang, Ying-Cong Chen, Yingcheng Liu, Christopher G Tarolli, Daniel Crepeau, Jan Bukartyk, Mithri R Junna, et al. Artificial intelligence-enabled detection and assessment of parkinson's disease using nocturnal breathing signals. *Nature medicine*, 28(10):2207–2215, 2022.

[61] Tal Yarkoni. The generalizability crisis. *Behavioral and Brain Sciences*, 45:e1, 2022.

[62] Paolo Zanini, Marco Congedo, Christian Jutten, Salem Said, and Yannick Berthoumieu. Transfer Learning: A Riemannian Geometry Framework With Applications to Brain–Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 65(5):1107–1116, May 2018.

[63] Hongyi Zhang and Suvrit Sra. First-order Methods for Geodesically Convex Optimization. In *Conference on Learning Theory*, pages 1617–1638. PMLR, June 2016. ISSN: 1938-7228.

A Appendix

A.1 Matrix operations

Given $\boldsymbol{\Sigma} \in \mathbb{S}_d^{++}$ and its Singular Value Decomposition (SVD) $\boldsymbol{\Sigma} = \boldsymbol{U}\,\mathrm{diag}(\lambda_1, \ldots, \lambda_d)\boldsymbol{U}^\top$, the matrix logarithm of $\boldsymbol{\Sigma}$ is

$$\log(\boldsymbol{\Sigma}) = \boldsymbol{U}\,\mathrm{diag}(\log(\lambda_1), \ldots, \log(\lambda_d))\boldsymbol{U}^\top. \tag{16}$$

$\boldsymbol{\Sigma}$ to the power $\alpha \in \mathbb{R}$ is

$$\boldsymbol{\Sigma}^\alpha = \boldsymbol{U}\,\mathrm{diag}(\lambda_1^\alpha, \ldots, \lambda_d^\alpha)\boldsymbol{U}^\top. \tag{17}$$

A.2 Proof of Lemma 2.1

First, we recall that the geodesic associated with the affine invariant metric from $\boldsymbol{\Sigma}$ to $\boldsymbol{\Sigma}'$ is

$$\boldsymbol{\Sigma} \sharp_\alpha \boldsymbol{\Sigma}' \triangleq \boldsymbol{\Sigma}^{1/2}\left(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-1/2}\right)^\alpha \boldsymbol{\Sigma}^{1/2} \quad \text{for } \alpha \in [0,1]. \tag{18}$$

Hence, $\boldsymbol{\Sigma}\sharp_\alpha \boldsymbol{I}_d = \boldsymbol{\Sigma}^{1-\alpha}$. From [59], the parallel transport of $\boldsymbol{\Sigma}'$ from $\boldsymbol{\Sigma}_1$ to $\boldsymbol{\Sigma}_2$ is

$$\boldsymbol{E}\boldsymbol{\Sigma}'\boldsymbol{E}^\top \quad \text{with} \quad \boldsymbol{E} \triangleq \boldsymbol{\Sigma}_1^{1/2}\left(\boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1/2}\right)^{1/2}\boldsymbol{\Sigma}_1^{-1/2}. \tag{19}$$

Hence, the parallel transport of $\boldsymbol{\Sigma}'$ from $\boldsymbol{\Sigma}$ to $\boldsymbol{\Sigma}\sharp_\alpha \boldsymbol{I}_d$ is $\boldsymbol{E}\boldsymbol{\Sigma}'\boldsymbol{E}^\top$ with

$$\begin{aligned}\boldsymbol{E} &\triangleq \boldsymbol{\Sigma}^{1/2}\left(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}^{1-\alpha}\boldsymbol{\Sigma}^{-1/2}\right)^{1/2}\boldsymbol{\Sigma}^{-1/2} \\ &= \boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{-\alpha/2}\boldsymbol{\Sigma}^{-1/2} = \boldsymbol{\Sigma}^{-\alpha/2}\end{aligned} \tag{20}$$

which concludes the proof.

A.3 Cross-spectrum computation and preprocessing

Bartlett estimator From [33], the features provided in the HarMNqEEG dataset have been computed using the Bartlett's estimator by averaging more than 20 consecutive and non-overlapping segments. Thus, data consist of cross-spectral matrices with a frequency range of $f_{\text{min}} = 1.17\,\text{Hz}$ to $f_{\text{max}} = 19.14\,\text{Hz}$, sampled at a resolution of $\Delta\omega = 0.39\,\text{Hz}$. These cross-spectral matrices are denoted $\mathbf{S}_{k,i}(\omega) \in \mathcal{H}_d^{++}$ where k is the site, i the participant and $\omega \in \{f_{\text{min}}, f_{\text{min}} + \Delta\omega, \ldots, f_{\text{max}}\}$.

Common average reference (CAR) The cross-spectrum matrices $\mathbf{S}_i(\omega)$ were re-referenced from their original montages with a CAR:

$$\widetilde{\mathbf{S}}_{k,i}(\omega) \triangleq \mathbf{H}\mathbf{S}_{k,i}(\omega)\mathbf{H}^\top \tag{21}$$

where $\mathbf{H} \triangleq \boldsymbol{I}_d - \mathbf{1}_d\mathbf{1}_d^\top/d$.

Global Scale Factor (GSF) Co-spectrum matrices were re-scaled with an individual scalar $\widehat{\zeta}_{k,i}$ that is calculated as the geometric mean of their power spectrum across sensors and frequencies:

$$\widehat{\zeta}_{k,i} \triangleq \exp\left(\frac{1}{N_\omega d}\sum_{\ell=0}^{N_\omega-1}\sum_{c=1}^{d}\log\left(\left(\widehat{\mathbf{S}}_{k,i}(f_{\text{min}} + \ell\Delta\omega)\right)_{c,c}\right)\right) \tag{22}$$

where $N_\omega \triangleq \frac{f_{\text{max}} - f_{\text{min}}}{\Delta\omega} + 1$. The GSF correction is then applied to the co-spectrum (the real part of the cross-spectrum) for all frequencies ω:

$$\boldsymbol{\Sigma}_{k,i}(\omega) \triangleq \Re\left(\widetilde{\mathbf{S}}_{k,i}(\omega)\right)/\widehat{\zeta}_{k,i}\,. \tag{23}$$

The $\boldsymbol{\Sigma}_{k,i}(\omega) \in \mathbb{S}_d^{++}$ are the features used the Section 4.

https://doi.org/10.52202/079017-1033

A.4 Baselines

The four baseline methods used in this work are detailed in the following. For every methods we have access to K labeled source domains, each including N_k covariance matrices and their corresponding variables of interest $(\boldsymbol{\Sigma}_{k,i}, y_{k,i})_{i=1}^{N_k}$. The method `DO Dummy` and the mixed-effects model baseline `DO Intercept` both access the mean value $\bar{y}_{\mathcal{T}}$ of the target domain variable to predict. We remind that as the dataset used in the empirical benchmarks is constituted of several frequency bands, each method is applied to each frequency band independently and then computed vectors are concatenated.

Domain-aware dummy model (`DO Dummy`) The `DO Dummy` simply returns the mean value $\bar{y}_{\mathcal{T}}$ for every predictions of a given target domain.

No re-center / No domain adaptation (`No DA`) The covariance matrices are used as input of the regression pipeline without any re-centering. First, the geometric mean of the source matrices is computed:

$$\overline{\boldsymbol{\Sigma}}_{\mathcal{S}} \triangleq \underset{\boldsymbol{\Sigma}\in\mathbb{S}_d^{++}}{\arg\min} \sum_{k=1}^{K}\sum_{i=1}^{N_k} \delta_R(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_{k,i})^2. \tag{24}$$

Then, source feature vectors are extracted with:

$$\phi(\boldsymbol{\Sigma}_{k,i}, \overline{\boldsymbol{\Sigma}}_{\mathcal{S}}) = \operatorname{uvec}\left(\log\left(\overline{\boldsymbol{\Sigma}}_{\mathcal{S}}^{-1/2}\boldsymbol{\Sigma}_{k,i}\overline{\boldsymbol{\Sigma}}_{\mathcal{S}}^{-1/2}\right)\right) \in \mathbb{R}^{d(d+1)/2}. \tag{25}$$

Finally, the target feature vectors are extracted from the target data $(\boldsymbol{\Sigma}_{\mathcal{T},i})_{i=1}^{N_{\mathcal{T}}}$ with:

$$\phi(\boldsymbol{\Sigma}_{\mathcal{T},i}, \overline{\boldsymbol{\Sigma}}_{\mathcal{S}}) = \operatorname{uvec}\left(\log\left(\overline{\boldsymbol{\Sigma}}_{\mathcal{S}}^{-1/2}\boldsymbol{\Sigma}_{\mathcal{T},i}\overline{\boldsymbol{\Sigma}}_{\mathcal{S}}^{-1/2}\right)\right) \in \mathbb{R}^{d(d+1)/2}. \tag{26}$$

Re-center to a common reference point (`Re-center`) Domains are re-centered to a common reference point, here we decided to use the identity. First, the geometric mean of each domain is computed:

$$\overline{\boldsymbol{\Sigma}}_{k} \triangleq \underset{\boldsymbol{\Sigma}\in\mathbb{S}_d^{++}}{\arg\min} \sum_{i=1}^{N_k} \delta_R(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_{k,i})^2 \ . \tag{27}$$

Then, feature vectors are extracted using the specific geometric mean of each domain:

$$\phi(\boldsymbol{\Sigma}_{k,i}, \overline{\boldsymbol{\Sigma}}_{k}) = \operatorname{uvec}\left(\log\left(\overline{\boldsymbol{\Sigma}}_{k}^{-1/2}\boldsymbol{\Sigma}_{k,i}\overline{\boldsymbol{\Sigma}}_{k}^{-1/2}\right)\right) \in \mathbb{R}^{d(d+1)/2} \ . \tag{28}$$

Covariance matrices of the target domain $(\boldsymbol{\Sigma}_{\mathcal{T},i})_{i=1}^{N_{\mathcal{T}}}$ are also re-centered to the identity using their geometric mean :

$$\overline{\boldsymbol{\Sigma}}_{\mathcal{T}} \triangleq \underset{\boldsymbol{\Sigma}\in\mathbb{S}_d^{++}}{\arg\min} \sum_{i=1}^{M} \delta_R(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_{\mathcal{T},i})^2 \tag{29}$$

$$\phi(\boldsymbol{\Sigma}_{\mathcal{T},i}, \overline{\boldsymbol{\Sigma}}_{\mathcal{T}}) = \operatorname{uvec}\left(\log\left(\overline{\boldsymbol{\Sigma}}_{\mathcal{T}}^{-1/2}\boldsymbol{\Sigma}_{i}\overline{\boldsymbol{\Sigma}}_{\mathcal{T}}^{-1/2}\right)\right) \in \mathbb{R}^{d(d+1)/2}. \tag{30}$$

For both `No DA` and `Re-center`, the regression task was performed using a Ridge regression, which included an intercept:

$$\boldsymbol{\beta}^{\star}_{\mathcal{S}}, \beta^{\star}_{0,\mathcal{S}} = \underset{\substack{\boldsymbol{\beta}\in\mathbb{R}^{d(d+1)/2}\\ \beta_0\in\mathbb{R}}}{\arg\min} \sum_{k=1}^{K}\sum_{i=1}^{N_k} \left(y_{k,i} - \boldsymbol{\beta}^{\top}\boldsymbol{z}_{k,i} - \beta_0\right)^2 + \lambda\left\|\boldsymbol{\beta}\right\|_2^2 \tag{31}$$

where $\boldsymbol{z}_{k,i}$ is computed with (25) or (28). The predicted values were computed as:

$$\widehat{y}_{\mathcal{T},i} = \left(\boldsymbol{\beta}^{\star}_{\mathcal{S}}\right)^{\top}\boldsymbol{z}_{\mathcal{T},i} + \beta^{\star}_{0,\mathcal{S}} \tag{32}$$

where $\boldsymbol{z}_{\mathcal{T},i}$ is computed with (26) or (30).

Domain-aware intercept (`DO Intercept`) In addition to the K labeled source domains, we assume to have access to the mean of the variable to predict of the target domain $\bar{y}_{\mathcal{T}}$. At train-time, we fit a Ridge regression with a specific intercept for each domain

$$\boldsymbol{\beta}^{\star}_{\mathcal{S}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{d(d+1)/2}}{\arg\min} \sum_{k=1}^{K}\sum_{i=1}^{N_k} \left(y_{k,i} - \boldsymbol{\beta}^{\top}\phi(\boldsymbol{\Sigma}_{k,i}, \overline{\boldsymbol{\Sigma}}_{\mathcal{S}}) - \bar{y}_k\right)^2 + \lambda\left\|\boldsymbol{\beta}\right\|_2^2 \ . \tag{33}$$

Then, at test-time, we fit a new intercept $\beta_{0,\mathcal{T}}$ using the target features:

$$\phi\left(\boldsymbol{\Sigma}_{\mathcal{T},i}, \overline{\boldsymbol{\Sigma}}_{\mathcal{S}}\right) = \operatorname{uvec}\left(\log\left(\overline{\boldsymbol{\Sigma}}_{\mathcal{S}}^{-1/2} \boldsymbol{\Sigma}_{\mathcal{T},i} \overline{\boldsymbol{\Sigma}}_{\mathcal{S}}^{-1/2}\right)\right) \in \mathbb{R}^{d(d+1)/2}. \tag{34}$$

The fitted intercept is

$$\beta_{0,\mathcal{T}} = \bar{y}_{\mathcal{T}} - \frac{1}{N_{\mathcal{T}}} \sum_{i=1}^{N_{\mathcal{T}}} (\boldsymbol{\beta}_{\mathcal{S}}^{\star})^{\top} \phi\left(\boldsymbol{\Sigma}_{\mathcal{T},i}, \overline{\boldsymbol{\Sigma}}_{\mathcal{S}}\right) \tag{35}$$

and the predictions are

$$\widehat{y}_{\mathcal{T},i} = (\boldsymbol{\beta}_{\mathcal{S}}^{\star})^{\top} \phi\left(\boldsymbol{\Sigma}_{\mathcal{T},i}, \overline{\boldsymbol{\Sigma}}_{\mathcal{S}}\right) + \beta_{0,\mathcal{T}} \, . \tag{36}$$

https://doi.org/10.52202/079017-1033

A.5 HarMNqEEG dataset

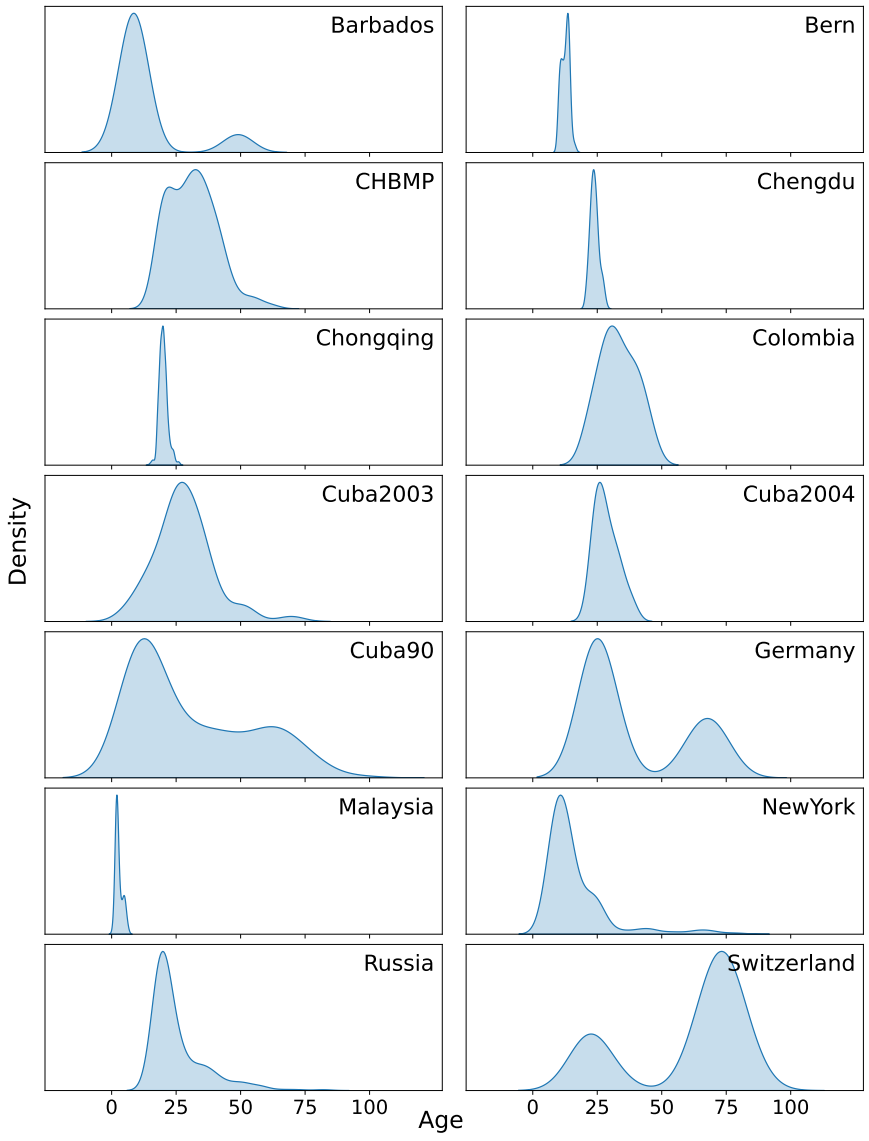

Figure 5: **Age distribution of the 14 sites of the HarMNqEEG dataset [33].** The distributions are represented with a kernel density estimate. The y-scales are not shared for visualization purpose.

A.6 Table of performance scores.

Table 1: **Performance scores for different combinations of source sites.** The remaining sites were used as target domains.

Spearman's $\rho \uparrow$

Sites source	DO Dummy	No DA	GREEN	Re-center	DO Intercept	GOPSA
Ba,Cho,G,S	0.51 ± 0.04	0.63 ± 0.02	0.69 ± 0.03	0.52 ± 0.02	0.75 ± 0.02	$\mathbf{0.78 \pm 0.02}$
Be,Chb,S	0.58 ± 0.02	0.73 ± 0.01	$\mathbf{0.75 \pm 0.02}$	0.43 ± 0.02	0.68 ± 0.02	0.72 ± 0.02
Ba,Co,G	0.62 ± 0.02	0.64 ± 0.02	0.72 ± 0.02	0.42 ± 0.02	0.71 ± 0.02	$\mathbf{0.74 \pm 0.02}$
Cu03,M,R,S	0.62 ± 0.03	0.63 ± 0.01	0.70 ± 0.05	0.46 ± 0.02	$\mathbf{0.76 \pm 0.02}$	0.75 ± 0.04
Ba,Be,Cho, Co,Cu90,G,R	0.77 ± 0.02	0.79 ± 0.01	0.82 ± 0.01	0.44 ± 0.03	0.85 ± 0.01	$\mathbf{0.87 \pm 0.01}$
Mean	0.62 ± 0.03	0.68 ± 0.01	0.74 ± 0.03	0.45 ± 0.02	0.75 ± 0.02	$\mathbf{0.77 \pm 0.02}$

R^2 score $\uparrow$

Sites source	DO Dummy	No DA	GREEN	Re-center	DO Intercept	GOPSA
Ba,Cho,G,S	0.21 ± 0.02	0.06 ± 0.06	0.26 ± 0.33	-0.32 ± 0.10	0.57 ± 0.03	$\mathbf{0.58 \pm 0.05}$
Be,Chb,S	0.25 ± 0.02	-0.07 ± 0.08	0.39 ± 0.30	-1.36 ± 0.13	0.43 ± 0.03	$\mathbf{0.49 \pm 0.03}$
Ba,Co,G	0.48 ± 0.03	0.26 ± 0.03	0.47 ± 0.09	0.10 ± 0.03	0.60 ± 0.03	$\mathbf{0.64 \pm 0.03}$
Cu03,M,R,S	0.26 ± 0.02	0.26 ± 0.04	0.48 ± 0.13	-0.30 ± 0.07	0.51 ± 0.02	$\mathbf{0.51 \pm 0.09}$
Ba,Be,Cho, Co,Cu90,G,R	0.60 ± 0.03	0.54 ± 0.02	0.62 ± 0.10	0.14 ± 0.02	$\mathbf{0.76 \pm 0.02}$	0.75 ± 0.02
Mean	0.36 ± 0.03	0.21 ± 0.05	0.44 ± 0.19	-0.35 ± 0.07	0.57 ± 0.02	$\mathbf{0.59 \pm 0.04}$

MAE $\downarrow$

Sites source	DO Dummy	No DA	GREEN	Re-center	DO Intercept	GOPSA
Ba,Cho,G,S	9.25 ± 0.16	12.00 ± 0.21	9.08 ± 1.98	14.69 ± 0.24	7.69 ± 0.19	$\mathbf{7.61 \pm 0.25}$
Be,Chb,S	9.48 ± 0.14	11.83 ± 0.37	8.67 ± 2.30	22.48 ± 0.24	9.28 ± 0.20	$\mathbf{8.61 \pm 0.20}$
Ba,Co,G	9.42 ± 0.14	13.83 ± 0.46	9.44 ± 0.77	15.25 ± 0.45	8.77 ± 0.15	$\mathbf{8.50 \pm 0.20}$
Cu03,M,R,S	9.64 ± 0.17	10.98 ± 0.22	$\mathbf{8.53 \pm 1.12}$	16.50 ± 0.25	8.94 ± 0.18	8.75 ± 0.72
Ba,Be,Cho, Co,Cu90,G,R	10.37 ± 0.23	11.40 ± 0.31	9.53 ± 1.10	15.92 ± 0.45	8.53 ± 0.23	$\mathbf{8.40 \pm 0.24}$
Mean	9.63 ± 0.17	12.01 ± 0.31	9.05 ± 1.45	16.97 ± 0.33	8.64 ± 0.19	$\mathbf{8.37 \pm 0.32}$

https://doi.org/10.52202/079017-1033

A.7 Figure 3 without normalization

Without `Re-center` and `Re-scale`:

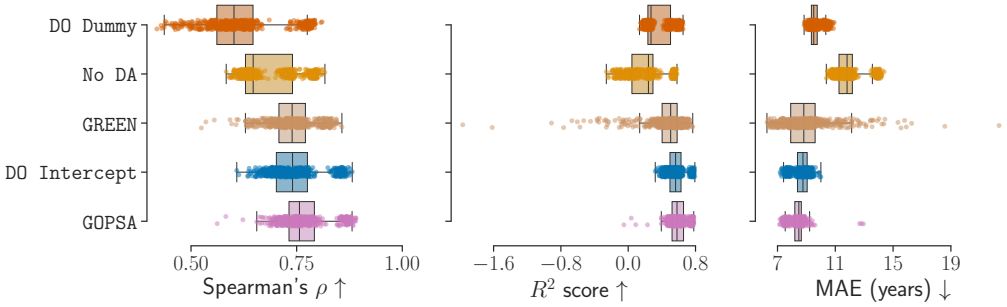

Figure 6: **Performance of four methods on several source-target combinations for three metrics:** Spearman's $\rho \uparrow$ (left), R^2 score $\uparrow$ (middle) and Mean Absolute Error $\downarrow$ (right). `Re-center` was removed from the plot to better visualize the other methods. A box represents the concatenated results across all site combinations. One point corresponds to one split of one site combination.

With `Re-center` and `Re-scale`:

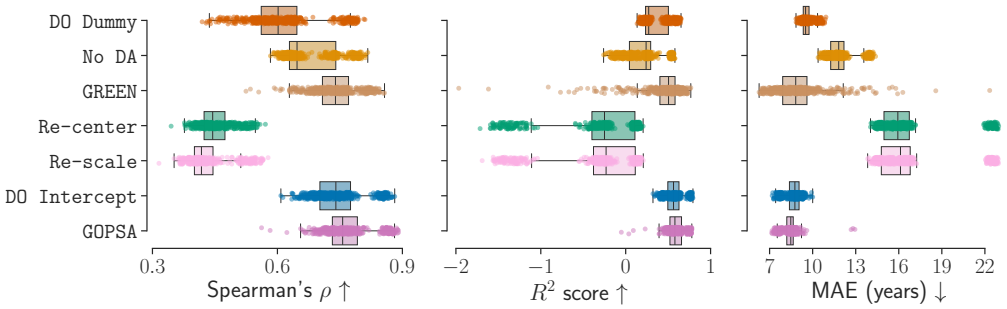

Figure 7: **Performance of all methods on several source-target combinations for three metrics:** Spearman's $\rho \uparrow$ (left), R^2 score $\uparrow$ (middle) and Mean Absolute Error $\downarrow$ (right). A box represents the concatenated results across all site combinations. One point corresponds to one split of one site combination.

A.8 Boxplots of each source-target sites for the three metrics

The following figures represent the performance scores that are displayed in subsection A.6.

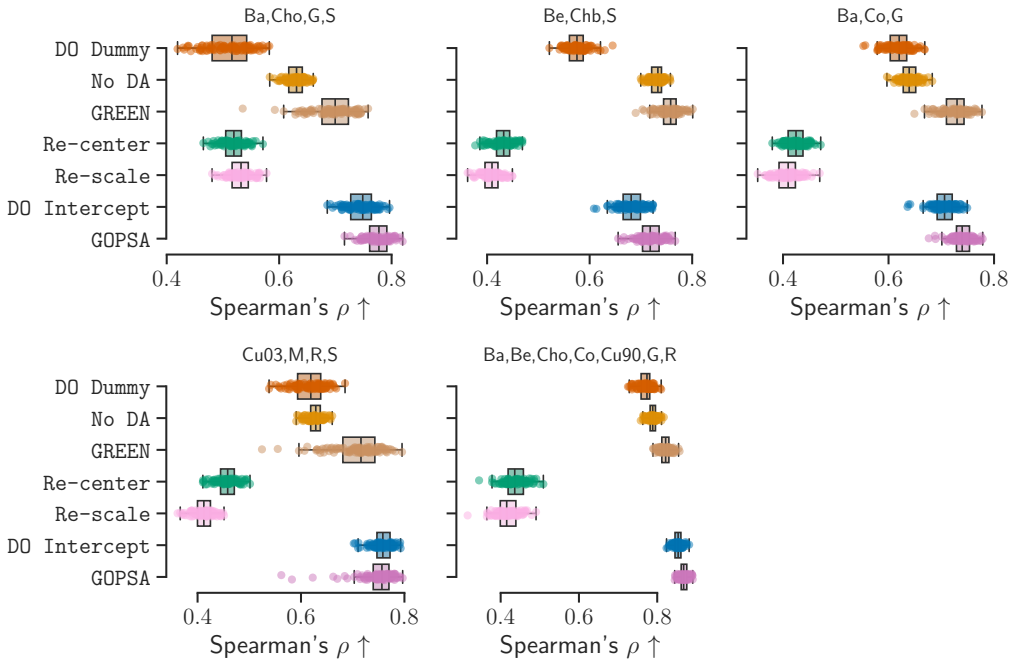

Figure 8: **Spearman's $\rho \uparrow$ for every site combination.** One panel corresponds to the results of one site combination. One point corresponds to one split.

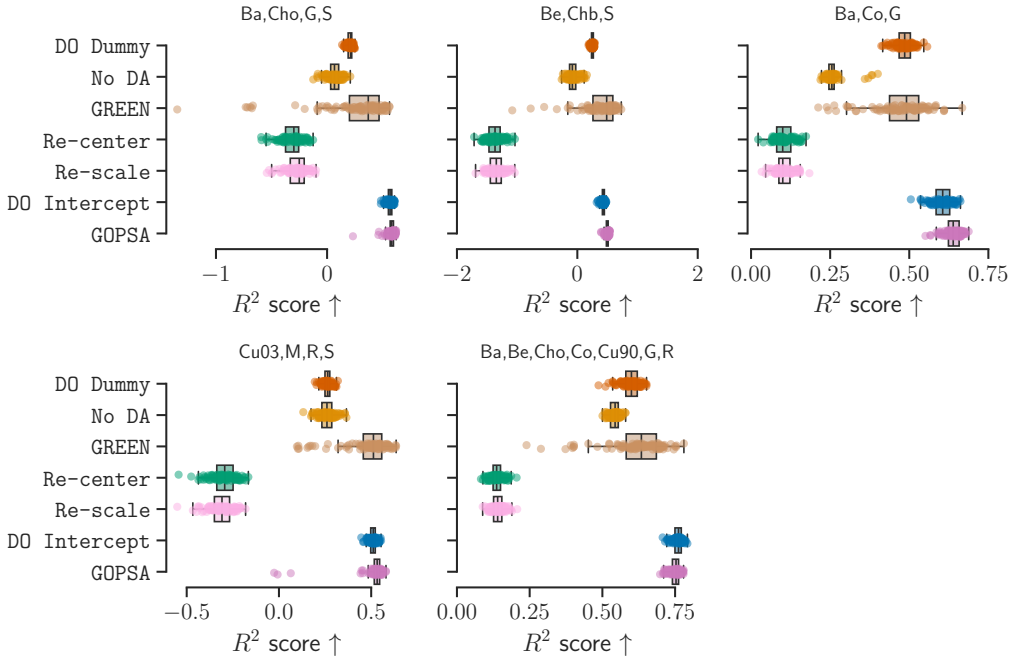

Figure 9: **R^2 score $\uparrow$ for every site combination.** One panel corresponds to the results of one site combination. One point corresponds to one split.

https://doi.org/10.52202/079017-1033

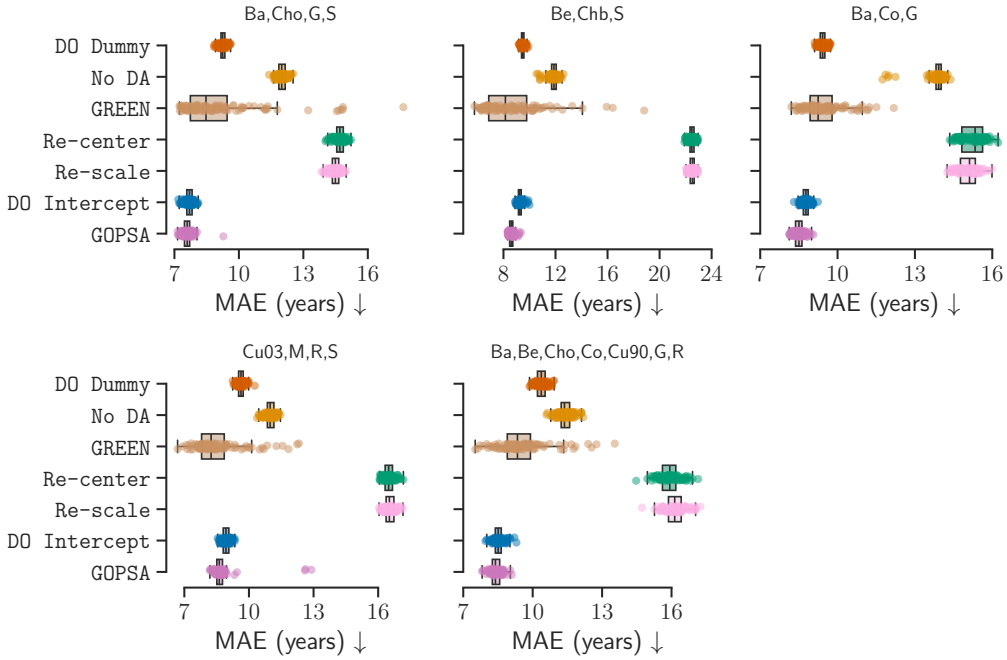

Figure 10: **Mean Absolute Error ↓ for every site combination.** One panel corresponds to the results of one site combination. One point correspond to one split.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim we propose a new test-time multi-source domain adaption method. We also claim this method is state of the art on the HarMNqEEG dataset. These two claims are asserted in Section 3 and in the **Results** paragraph of Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A limitation of the proposed method is to require the mean outcome value $\bar{y}_{\mathcal{T}}$ of the target set. This limitation is stated in the **Contributions** paragraph of Section 1 and discussed in Section 6.

   Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The proposed method uses the parallel transport from a given SPD matrix to the identity. The formula is presented in Lemma 2.1, and the proof is provided in Appendix A.2.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The empirical benchmarks (Section 4) use the HarMNqEEG dataset which is available in open access. All the preprocessing steps, as well as the baselines, are extensively detailed in Appendix A.3 and Appendix A.4, respectively. The technical details (such as gradient computation and optimization algorithm) to implement the proposed method are provided in Section 3.

https://doi.org/10.52202/079017-1033

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The dataset HarMNqEEG [33] is in open access. We provide the code to reproduce the experiments from the raw data.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 3 and Section 4 provide all the necessary details to reproduce the results: optimizer, splitting strategy, hyperparameter selection, and model evaluation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars and box plots computed from different splits of the data were reported. p-values following [37] between the best baseline and the proposed method were computed and presented in Figure 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided in Section 4 the computational resources used for running all the benchmarks.

https://doi.org/10.52202/079017-1033

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors acknowledged having read the NeurIPS Code of Ethics, and the paper conforms to it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The proposed approach opens up various applications at the interface between machine learning and biostatistics, such as biomarker exploration in large multicenter clinical trials. References are provided in Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Anonymized data are already available in open access, and the proposed benchmark is limited to biomarker regression.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The dataset presented by [33] is an open-source dataset accessible on `https://synapse.org`, `https://10.0.28.135/syn26712693`. We cited the work beyond the data reference and gave credit to scientific ideas and concepts based on [33].

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new assets are provided in the paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

https://doi.org/10.52202/079017-1033

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification:

    The human research data used in this study was curated by [33] and conducted by academic researchers. We do not reproduce their data acquisition protocol here and refer to the original work [33]. We argue that this is appropriate as our work propose a new statistical method and does not present novel biomedical results.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification:

    The data acquisition was approved by institutional review boards as stated in [33]: *"The studies involving human participants were reviewed and approved by the Ethics Committees of all involved institutions. In all cases, the participants and/or their legal guardians/next of kin provided written informed consent to participate in this study. All data were de-identified, and participants gave permission for their data to be shared as part of the informed consent process."*.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.