Riemannian Embedding Banks for Common Spatial Patterns with EEG-based SPD Neural Networks

Yoon-Je Suh[1]*, **Byung Hyung Kim**[2]*†
[1]School of Electrical Engineering
[2]School of Computing
KAIST, Republic of Korea
{yoonje, bhyung}@kaist.ac.kr

Abstract

Modeling non-linear data as symmetric positive definite (SPD) matrices on Riemannian manifolds has attracted much attention for various classification tasks. In the context of deep learning, SPD matrix-based Riemannian networks have been shown to be a promising solution for classifying electroencephalogram (EEG) signals, capturing the Riemannian geometry within their structured 2D feature representation. However, existing approaches usually learn spatial-temporal structures in an embedding space for all available EEG signals, and their optimization procedures rely on computationally expensive iterations. Furthermore, these approaches often struggle to encode all of the various types of relationships into a single distance metric, resulting in a loss of generality. To address the above limitations, we propose a Riemannian Embedding Banks method, which divides the problem of common spatial patterns learning in an entire embedding space into K-subproblems and builds one model for each subproblem, to be combined with SPD neural networks. By leveraging the concept of the "*separate-to-learn*" technology on a Riemannian manifold, REB divides the data and the embedding space into K non-overlapping subsets and learns K separate distance metrics in a Riemannian geometric space instead of the vector space. Then, the learned K non-overlapping subsets are grouped into neurons in the SPD neural network's embedding layer. Experimental results on public EEG datasets demonstrate the superiority of the proposed approach for learning common spatial patterns of EEG signals despite their non-stationary nature, increasing the convergence speed while maintaining generalization.

Introduction

Covariance-based SPD matrices have been widely used to classify time-series data in various applications (Pennec, Sommer, and Fletcher 2019; Congedo, Barachant, and Bhatia 2017), exploiting the second-order statistics to capture and represent the temporal fluctuations of different data lengths. By taking the advances on the higher-order statistics in SPD matrices, recent EEG signal processing techniques have increasingly carried out practical computations on Riemannian manifolds in feature representation (Barachant et al. 2011; Olias et al. 2019; Rodrigues, Jutten, and Congedo 2018).

However, these successful applications suffer from the computational efficiency of distance or similarity on a Riemannian manifold (Cherian et al. 2013). Since the data points on the SPD manifold are geodesics along the manifold's curvature, computing the geodesics increases computational costs due to the non-existence of closed-form solutions. Besides, applying existing Euclidean-based distance measure directly to SPD matrices often results in undesirable effects. i.e., swelling of the diffusion tensors (Pennec, Fillard, and Ayache 2006). Several approaches have been presented to properly encode the Riemannian geometry of SPD manifolds by typically flattening the SPD manifolds through a tangent space approximation with the matrix logarithm computation (Pennec, Sommer, and Fletcher 2019; Arsigny et al. 2007; Congedo, Barachant, and Bhatia 2017). Then, Euclidean classifiers such as SVM or kNN have been used to learn features represented in the space (Barachant et al. 2011). However, these shallow learning schemes of the matrix logarithm computation for a flatten vector representation have often led to poor performance, leading to sub-optimal solutions on the non-linear manifolds.

In light of the recent success in metric learning algorithms with deep neural networks (DML), several methods have been proposed to interpolate, restore, and classify SPD matrices on a Riemannian manifold (Sra 2012; Liu et al. 2019b). Existing approaches in DML leans a single distance metric for all samples from the given data distribution. The method usually pulls similar samples closer while pushing different samples further away to learn the semantic distance. However, this strategy can not be directly applied to learning EEG data because of its non-linear, non-uniformly distributed, and complex data structure (Kim and Jo 2020; Alarcao and Fonseca 2019). This limitation exhibits inter- and intra-class variability problems for learning high-dimensional neural activities in the brain. In consequence, using DML algorithms to learn a single distance metric between EEG-based SPD matrices becomes a challenging issue because the level of relative similarity in each training pairs or triplet determines how fast the network learns correctly.

To overcome this limitation, we aim to develop a cost-efficient metric learning algorithm on SPD manifolds mo-

[*] These co-first authors contributed equally to this work.

[†] Corresponding Author: *Byung Hyung Kim*.

tivated by the *separate-to-learn* (SL) technique, which partitions the problem of classification into sub-problems and builds one model to solve each sub-problem. The Filter Bank Common Spatial Pattern (FBCSP) (Ang et al. 2008) and the Riemannian potato field (Barthélemy et al. 2019) algorithms are the case for SL-based EEG frequency bank algorithms. They divide frequencies decomposed from EEG signals into K bands and learn common patterns of each band individually. The selected spatial features from the K bands are used to classify motor imagery (MI) related tasks in brain-computer interfaces (BCI). Recently, SL-based algorithms have been integrated into deep neural networks (DNN) to perform various Euclidean-based tasks (Liu et al. 2019a; Sanakoyeu et al. 2019). However, applying the SL strategy into deep Riemannian networks is 1) intractable due to SPD constraints, which also leads 2) the difficulty in optimization.

To solve the above problems, we propose a new SL-based model to embed SPD matrices under a non-linear framework named Riemannian Embedding Banks (REB). The proposed model aims to solve the common spatial pattern problem with respect to the embedding learning problem in DNNs to be combined with Riemannian networks. REB divides the spatial patterns in an entire Riemannian embedding space into K-subproblems and build one model for each sub-problem, focusing on only considering the samples assigned to the corresponding cluster. Without loss of generality in embedding learning, all models share the underlying feature representation. Then the final embedding space is seamlessly composed by concatenating the solutions on each of the non-overlapping sub-spaces.

We use the Riemannian SPD Matrix Network (SPDNet) (Huang and Van Gool 2017) as a baseline SPD neural network. Given SPD matrices as inputs, SPDNet learns the matrix characteristics, preserving the SPD structure across their proposed layers, named BiMap, ReEig, and LogEig to be non-linearly mapped into latent space where the matrix features are transformed to a Euclidean space for further classification. Same as other variants of SPDNet (Liu et al. 2019b; Brooks et al. 2019), REB takes SPD matrices as inputs and uses a sequence of BiMap and ReEig for extracting SPD matrix features, which can be learned jointly through matrix back-propagation with stochastic gradient descent (SGD) (Ionescu, Vantzos, and Sminchisescu 2015). Our approach can be replacement of the fully-connected (FC) layer for learning embeddings in the existing DML approaches, regardless of the loss function used for training.

Preliminaries

Riemannian Geometry of SPD Matrices

We denote $\mathbf{X}$ as a $N_c \times N_c$ symmetric positive definite (SPD) matrix, where $\mathbf{x} \in \mathbb{R}^{N_c \times N_s}$ indicates the EEG multivariate time-series signals ($\mathbf{x} > 0$) and N_c and N_s are the numbers of channels and samples, respectively. The set of all SPD matrices lies in a differentiable Riemannian manifold $\mathcal{M}$, which is portrayed as a surface with a non-positive curvature. Between any two SPD matrices $\mathbf{X}_1$, $\mathbf{X}_2 \in \mathcal{M}$, a unique curve, called a *geodesic*, is determined by minimizing the length of the curve between the two points. Several metrics have been presented to capture the non-linearity (Pennec, Fillard, and Ayache 2006; Arsigny et al. 2006; Sra 2012; Kulis, Sustik, and Dhillon 2006; Cichocki, Cruces, and Amari 2015). Among them, the affine-invariant Riemannian metric (AIRM) has found great popularity in geometry-aware algorithms for the processing of SPD matrices (Barachant et al. 2011; Rodrigues, Jutten, and Congedo 2018; Yair, Ben-Chen, and Talmon 2019). AIRM is defined as follows:

$$\begin{aligned}\delta_R(\mathbf{X}_1, \mathbf{X}_2) &= \|\mathrm{Log}(\mathbf{X}_1^{-1/2}\mathbf{X}_2\mathbf{X}_1^{-1/2})\|_F \\ &= (\sum_{c=1}^{C} \log^2 \lambda_c)^{1/2},\end{aligned} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\mathrm{Log}(\cdot)$ is the matrix logarithm, and λ_c, $c = 1, \ldots, C$ are the real eigenvalues of $\mathbf{X}^{-1/2}\mathbf{X}_2\mathbf{X}_1^{-1/2}$. In this this study, we refer to the Riemannian distance on the manifold $\mathcal{M}$ as AIRM.

Riemannian SPD Matrix Network (SPDNet)

SPDNet exploits the Riemannian geometry across their proposed layers for more compact and discriminative SPD matrix features, preserving the manifold structure. Let $\mathbf{X}_{s-1}$, $\mathbf{W}_s$, and $\mathbf{X}_s$ be the SPD matrix, transformation matrix, and resulting matrix in the s-th layer, respectively. We summarizes the BiMap and ReEig layers of SPDNet as follows:

- The BiMap layer $f_m^{(s)}$ aims to generate more discriminative and compact SPD matrix features by transforming the inputs into low-dimensional SPD matrices through bilinear mapping:

  $$\mathbf{X}_k = f_m^{(s)}(\mathbf{X}_s; \mathbf{X}_{s-1}) = \mathbf{W}_s\mathbf{X}_{s-1}\mathbf{W}_s^{\intercal}, \tag{2}$$

  where the transformation $\mathbf{W}_s$ should be constrained to a raw full-rank matrix to output $\mathbf{X}_s$ in the form of an SPD matrix.

- The ReEig layer $f_r^{(s)}$ is similar to the ReLU layer (Nair and Hinton 2010). The layer utilizes a non-linear activation to improve the discrimination by rectifying the SPD matrices with their small positive eigenvalues:

  $$\mathbf{X}_s = f_r^{(s)}(\mathbf{X}_{s-1}) = \mathbf{U}_{s-1}\max(\epsilon\mathbf{I}, \mathbf{\Sigma}_{s-1})\mathbf{U}_{s-1}^{\intercal}, \tag{3}$$

  where $\max(\cdot,\cdot)$ is the maximum function, $\mathbf{U}_{s-1}$ and $\mathbf{\Sigma}_{s-1}$ are learned by the eigenvalue decomposition of $\mathbf{X}_{s-1} = \mathbf{U}_{s-1}\mathbf{\Sigma}_{s-1}\mathbf{U}_{s-1}^{\intercal}$, ϵ is a threshold parameter, and $\mathbf{I}$ is the identity matrix.

SPDNet suggests a classification scheme with classical neural network layers, such as a FC layer f_u and a softmax layer f_s. For further classification, the two layers can be inserted after the LogEig layer, which projects the vectorization of the output SPD feature manifold to a Euclidean space.

Deep Metric Learning

Developing efficient DML functions has been a crucial factor in improving the performance of learned features (Roth et al. 2020). Facility Location (Song et al. 2017) learns a cluster quality metric, Histogram Loss (Ustinova and Lempitsky 2016) minimizes the overlap between the distance distribution of positive and negative samples.

Despite their advances, only a small portion is informative and provides a learning signal. Hence, designing a suitable sampling strategy also matters. Although some works (Roth et al. 2020) presented hard and semi-hard negative mining strategies that offer faster convergence by retrieving samples in the high-variance region, this approach often leads to collapsed models. For further details, refer to the below section. A lot of recent research efforts have been devoted to devising new sampling strategies. RankMI (Kemertas et al. 2020) maximizes the mutual information among same-category items and leans low proximity for items from different categories. RLL (Wang et al. 2019) forces the positive pairwise distance smaller than a threshold, which is the diameter of each class's hypersphere. DSML (Yuan et al. 2019) proposed a Signal-to-Noise Ratio (SNR) distance metric that measures the level of anchor features compared to other categorical features. DWL (Wu et al. 2017) presented a new sampling strategy, where samples are distributed uniformly according to their relative distance from neighbors.

However, existing sampling approaches either require running an expensive preprocessing step on the entire dataset for every epoch (Harwood et al. 2017; Liu et al. 2019b) or suffer a lack of global information because a single randomly-drawn mini-batch from an embedding space provides only a local view on the entire dataset.

Our approach is orthogonal to these methods by providing 1) a framework for learning a distance metric independent on the choice of a particular loss function and 2) independent learners assigned to the specific sub-space, including the corresponding portion of the data. Each learner reduces the training complexity in a single model without extra parameters. This is the major difference to the existing ensemble learning methods that train multiple learners inside a single framework in combination with proper loss functions (Opitz et al. 2020; Yuan, Yang, and Zhang 2017).

Riemannian Embedding Banks

Our configuration of SPDNet comprises a block sequence of BiMap and ReEig layers. From the SPD matrix features resulted from the baseline SPDNet, the proposed REB aims to 1) split the entire embedding dimensions into multiple clusters along with the SPD matrix features and learn the characteristics independently in each cluster. 2) Then, the classification tasks are conducted by merging a consequence of individual solutions (Figure 1).

SPD *Separate-to-Learn* Layer

Separating SPD embeddings: We denote $\tilde{\mathbf{X}}_i \in \mathrm{Sym}_d^+$ as the i-th resulting SPD matrix feature extracted from the baseline SPDNet with the input SPD matrix $\mathbf{X}_i \in \mathrm{Sym}_m^+$. Let K be the the number of sub-space. This layer group all features in the embedding space into K clusters. The output of the layer can be defined by a nonlinear function f_p as:

$$C_k = f_p(\tilde{\mathbf{X}}_i), \tag{4}$$

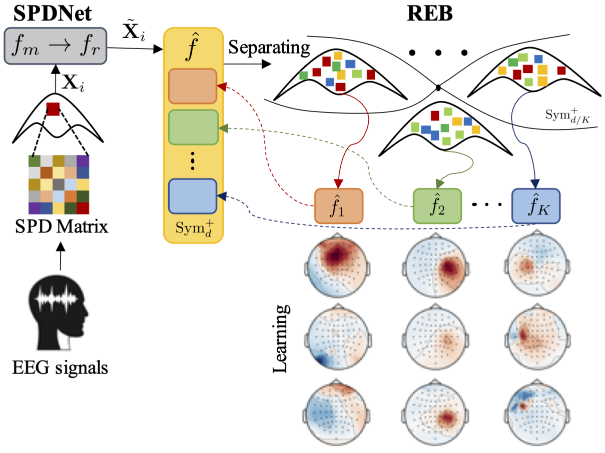


Figure 1: The overview of the proposed REB with SPDNet. SPD matrix features extracted from SPDNet are separated into K clusters.

where $C_k \in \mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ is k-th resulting cluster where $\tilde{\mathbf{X}}_i$ is assigned. Typically, the K-means clustering algorithm has been taken to determine the group of $\mathcal{C}$ (Sanakoyeu et al. 2019). However, the conventional Euclidean-based clustering algorithm can not be applicable because the input features of this layer are SPD matrices. Some unsupervised clustering algorithms have been presented based on Riemannian manifolds (Stanitsas et al. 2017; Zheng, Qiu, and Huang 2018). However, they cannot be utilized because iterative updates of candidate centroids on Riemannian spaces require intensive calculation, leading to sub-optimal problems when applying them to DNNs.

To overcome this problem, we devise an objective function to optimize the function f_p considering class relationships between triplets (anchor, positive, and negative samples) in each cluster. While useful clustering algorithms try to make only homogeneous samples near each other, we aim to partition a set of points into K sets such that the samples are informative for learning both homogeneous and heterogeneous aspects of triplets in metric learning. Assuming all clusters have an equal amount of data,

- We first intend to gather a set of samples in the same cluster are closer to each other than to those in other clusters as follows:

$$\mathcal{L}_{\mathcal{A}}(\tilde{\mathbf{X}}_i; f_p) = \sum_{\tilde{\mathbf{X}}_j \in f_p(\tilde{\mathbf{X}}_i)} \delta_R^2(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j). \tag{5}$$

This function ensures hard negatives within a cluster have a chance of being sampled naturally without explicitly performing a hard negative mining procedure.

- To provide more informative positive samples, they should be dissociated, so that such samples can best present the diversity of the training data. To this end, the target is to maximize the pair-wise distance between two input SPD matrices with the same label,

$$\mathcal{L}_{\mathcal{P}}(\tilde{\mathbf{X}}_i, y_i; f_p) = -\sum_{\tilde{\mathbf{X}}_j \in f_p(\tilde{\mathbf{X}}_i), y_j = y_i} \delta_R^2(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j), \tag{6}$$

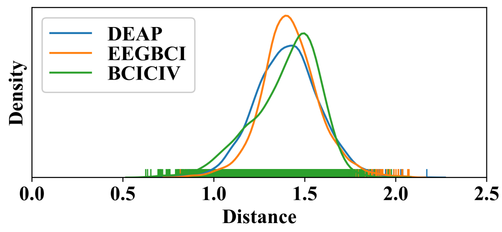


Figure 2: Empirical pairwise-distance distributions for negative pairs on DEAP, EEGBCI, and BCICIV.

where y_i is the class label associated with i-th resulting SPD matrix feature, $\tilde{\mathbf{X}}_i$ and $\tilde{\mathbf{X}}_j$ are positive relationship.

- Sampling informative negative examples have been a critical issue. (Semi-) hard negative mining strategy has contributed to accelerating convergence, providing non-trivial triplets under well-designed batch selection. However, the mining strategy severely leads to collapsed EEG-based models because the non-stationary property of the data influences high intra-class variation, which causes negative effects on sampling triplets. Under this unfavorable condition, hard negative mining yields noisy gradients with low SNR in the high variance area and cannot push two examples apart. Along with the same line, semi-hard negative mining might converge quickly at the beginning, finding a narrow set in between. However, the network will stop making progress since there might be no examples left within the embedding. To overcome this issue, inspired by the study in (Wu et al. 2017), we aim to design that negative samples are spread out according to distance by following the distribution of pairwise distances $\mathcal{D}(\cdot,\cdot)$ asymptotically as follows:

$$q(\mathcal{D}) \propto \mathcal{Q}(\mathcal{D}) = \mathcal{D}^{n-2}[1-\frac{1}{4}\mathcal{D}^2]^{\frac{n-3}{2}}, \qquad (7)$$

where the learned Riemannian embeddings are to constrained to uniformly distributed on the n-dimensional unit sphere $\mathbb{S}^{n-1}$ for large n (Lee 2006; Wu et al. 2017). Our empirical analysis of pairwise distance distributions for negative pairs on the embeddings (Figure 2) justifies our motivation of designing a clustering strategy. The bell-shaped curves give a chance to examples to be sampled according to their distance in a cluster. Thus, we devise a loss function to impose a penalty when negative samples are not uniformly distributed according to their distance with an anchor.

$$\mathcal{L}_{\mathcal{N}}(\tilde{\mathbf{X}}_i; f_p) = \sum_{\tilde{\mathbf{X}}_m \in f_p(\tilde{\mathbf{X}}_i), y_m \neq y_i} \log(\min(10^{-4}, \mathcal{Q}(\mathcal{D}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_m)))), \qquad (8)$$

where $\mathbf{X}_i$ and $\mathbf{X}_m$ are negative relationship. $\mathcal{D}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_m)$ = $\parallel$ Log($\tilde{\mathbf{X}}_i$) − Log($\tilde{\mathbf{X}}_m$) $\parallel_F$, where Log($\cdot$) is the matrix and $\parallel \cdot \parallel_F$ is the Frobenius norm of the matrix logarithm. The function offers each cluster can contain a wide range of negative examples, and thus steadily produce informative examples while controlling the variance.

In sum, optimizing the function f_p can be achieved to minimize the following the loss function:

$$\mathcal{L}_k^{\mathcal{C}}(\tilde{\mathbf{X}}_i; f_p) = \mathcal{L}_{\mathcal{A}}(\tilde{\mathbf{X}}_i; f_p) + \mathcal{L}_{\mathcal{P}}(\tilde{\mathbf{X}}_i; f_p) + \mathcal{L}_{\mathcal{N}}(\tilde{\mathbf{X}}_i; f_p), \qquad (9)$$

where $\mathcal{L}_k^{\mathcal{C}}(\tilde{\mathbf{X}}_i; f_p)$ determines the clustering quality of C_k. Given the K number of clusters, we also decompose the function $\hat{f}(\cdot; \theta_{\hat{f}}) : \mathrm{Sym}_m^+ \rightarrow \mathrm{Sym}_d^+$ into K functions $\{\hat{f}_1, \hat{f}_2, \ldots \hat{f}_K\}$, where each $\hat{f}_k$ maps the input into the d/K-dimensional sub-space of the original d-dimensional embedding space: $\hat{f}_k(\cdot; \theta_{\hat{f}_k}) : \mathrm{Sym}_m^+ \rightarrow \mathrm{Sym}_{d/K}^+$.

Learning separated SPD embeddings: All of the separated learners associated with their clusters are trained independently. That is, only one of the learners is updated in each training iteration. We uniformly sample a cluster $C_k, 1 \leq k \leq K$ and draw a random mini-batch $\mathcal{B}$ from it ($\mathcal{B} \subset C_k$). Then, a learner $\hat{f}_k(\cdot; \theta_{\hat{f}_k})$ aims to minimize the following function:

$$\mathcal{L}_k^{\hat{f}_k}(\tilde{\mathbf{X}}_i; \hat{f}_k) = \sum_{\tilde{\mathbf{X}}_i \in \mathcal{B}} [f_d(\tilde{\mathbf{X}}_i; \hat{f}_k)], \qquad (10)$$

where the function $f_d(\tilde{\mathbf{X}}_i; \hat{f}_k)$ discriminates inter- and intra-class features assigned to a learner $\hat{f}_k$ in metric learning. There can be several alternative metric learning algorithms (Roth et al. 2020). In order to realize the layer, we selected two deep metric learning structures, including Margin loss (Wu et al. 2017) and Triplet loss (Song et al. 2017). For instance, the function $f_d(\tilde{\mathbf{X}}_i; \hat{f}_k)$ with Triplet loss can be defined as

$$f_d^{\text{triplet}}(\tilde{\mathbf{X}}_i; \hat{f}_m) = [\delta_R^2(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j) - \delta_R^2(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_m) + \alpha]_+, \qquad (11)$$

where $\tilde{\mathbf{X}}_j, \tilde{\mathbf{X}}_m \in \mathcal{B}$. $\tilde{\mathbf{X}}_j$ is a positive neighbor and $\tilde{\mathbf{X}}_m$ is a negative neighbor with $\tilde{\mathbf{X}}_i$. $\delta_R(\cdot,\cdot)$ is the Riemannian distance measured by the AIRM in (1) and α is the margin. We note that each backward pass for (10) updates the shared parameters θ_ϕ of the baseline SPDNet associated with the only parameters of $\theta_{\hat{f}_k}$. Later, for the final classification, the full embedding is configured simply by concatenating the K sub-embeddings produced by the individual learners.

SPD Classification Layer

The layer aims to classify SPD matrix features as a consequence of individual learners' solution. This process is done by concatenating the embeddings associated with the K learners as follows:

$$\hat{f} = [f_d(\tilde{\mathbf{X}}_i; \hat{f}_1), f_d(\tilde{\mathbf{X}}_i; \hat{f}_2), \cdots, f_d(\tilde{\mathbf{X}}_i; \hat{f}_K)]. \qquad (12)$$

Hence, the output of REB can be learned by minimizing the following loss function for each cluster:

$$\mathcal{L}_k = \lambda_1 \sum_{\tilde{\mathbf{X}}_i \in C_k} \mathcal{L}_k^{\mathcal{C}}(\tilde{\mathbf{X}}_i; f_p) + \lambda_2 \sum_{\tilde{\mathbf{X}}_i \in C_k} \mathcal{L}_k^{\hat{f}_k}(\tilde{\mathbf{X}}_i; \hat{f}_k). \qquad (13)$$

Algorithm 1 Riemannian Embedding Banks

```
Input: Training SPD matrices $\{\mathbf{X}, y\}$, initialized layer parameters $\theta_\phi$ and weights $\theta_{\hat{f}}$, and the learning rate $l_r$ for the losses.
Output: The parameters $\theta_\phi, \theta_{\hat{f}}$
$\tilde{\mathbf{X}} \leftarrow$ SPDNet output ($\mathbf{X} \rightarrow f_b \rightarrow f_r \rightarrow \tilde{\mathbf{X}}$)
$epoch \leftarrow 0$
while not converge do
    $\mathcal{C} = \{C_1, \cdots, C_K\} \leftarrow$ Separate $\tilde{\mathbf{X}}$ in $\hat{f}$ into $K$ groups
    $\hat{f} = \{\hat{f}_1, \cdots, \hat{f}_K\} \leftarrow$ Assign a learner $\hat{f}_k$ to $C_k$
    Compute the loss $\mathcal{L}_k^{\mathcal{C}}$ of each cluster $C_k$ in (9)
    repeat
        $\mathcal{B} \leftarrow$ Draw a mini-batch from $C_k$
        Compute $\mathcal{L}_k^{\hat{f}_k}$ of each learner $\hat{f}_k$ with $\mathcal{B}$ in (10)
        Compute $\mathcal{L}_k$ in (13)
        Compute backprogagation error $\frac{\partial \mathcal{L}_k}{\partial C_k}$ in (14)
        Update weights $\theta_\phi, \theta_{\hat{f}_k}$ with $l_r$ and $\mathcal{L}_k$
    until Epoch completed
    $epoch \leftarrow epoch + 1$
end while
$\hat{f} \leftarrow$ Concatenate all $\hat{f}_k$
$\theta_\phi, \theta_{\hat{f}} \leftarrow$ Fine-tune with $\tilde{\mathbf{X}}, \theta_\phi, \theta_{\hat{f}}, \hat{f}$
return $\theta_\phi, \theta_{\hat{f}}$
```

Learning with REB

The proposed model integrated with SPDNet can be written as a series of non-linear function compositions. For training each layer, we exploit the matrix back-propagation (Ionescu, Vantzos, and Sminchisescu 2015) with stochastic gradient descent for computing the gradients in ReEig and LogEig layers where the eigenvalue decomposition of SPD matrices is involved. For updating the weights in BiMap layers, we update them on Stiefel manifolds. For more details of computing the gradients of the involved data in the three layers, readers are referred to (Huang and Van Gool 2017).

For the gradients of the proposed REB, the updating schemes are achieved by the following the chain rule:

$$\frac{\partial \mathcal{L}_k}{\partial C_k} = \left(\frac{\partial \mathcal{L}_k^{\mathcal{C}}}{\partial f_p} + \frac{\partial \mathcal{L}_k^{\hat{f}_k}}{\partial f_d} \cdot \frac{\partial f_d}{\partial f_p}\right) \cdot \frac{\partial f_p}{\partial \tilde{\mathbf{X}}_i}. \qquad (14)$$

While we have different distance metrics, all methods measure the Riemannian distance using the AIRM, in which the gradient is computed by (Harandi, Salzmann, and Hartley 2018). The proposed REB is summarized in Algorithm 1.

Experiments

Comparison of other State-of-the-art Methods

We compared our approach to a series of state-of-the-art methods: Facility Location, Histogram Loss, RLL, RankMI, BoMS, DSML, and DWL. All methods are described in the above section. To verify the efficiency of SPD neural networks, we also compared with two shallow Riemannian methods: MDM (Barachant et al. 2011) and FgMDM (Barachant et al. 2013). For the methods, we use authors' published source codes and tune the parameters according to the original works. If necessary, we empirically set the best parameters with the highest accuracy based on the original study. For instance, we set $\lambda_1 = 1.3$ and $\lambda_2 = 0.7$ for evaluating BoMS.

Methods		Dataset			
		BCICIV	EEGBCI	DEAP-4	DEAP-9
MDM		24.4	55.8	38.0	21.2
FgMDM		30.6	55.3	48.7	33.1
Facility Location		46.9	65.1	55.3	37.4
Histogram Loss		48.0	63.3	54.8	38.5
RLL		49.5	70.3	55.7	40.1
RankMI		46.3	71.6	54.8	40.5
BoMS		46.8	73.8	53.6	39.8
Triplet	Semihard	32.5	51.5	42.5	37.2
	Random	31.5	53.7	44.6	38.1
	DSML	44.4	74.2	53.6	41.8
	DWL	47.3	**75.1**	52.5	42.5
	REB	46.2	73.5	55.4	48.7
Margin	Random	32.2	56.7	47.5	39.9
	DSML	45.9	73.8	55.6	42.5
	DWL	48.8	74.4	56.4	42.9
	REB	**51.2**	73.2	**59.2**	**50.4**

Table 1: Comparison of mAP results against the state-of-the-art methods on BCICIV, EEGBCI, and DEAP.

Datasets

We evaluated REB on different tasks during EEG classification: emotion recognition, and motor imagery tasks using three EEG datasets. The two motor imagery datasets were imported from an open-source repository[1] (Jayaram and Barachant 2018).

- BCICIV (Tangermann et al. 2012): The BCI competition IV Database-Dataset IIa (BCICIV) contains 22-channeled EEG signals at 250 Hz gathered from nine subjects. They were asked to imagine four different motor imagery tasks during 6 s. Each participant conducted 6 runs of 48 trials from 2 sessions.
- EEGBCI (Schalk et al. 2004): The EEG Motor Movement/Imagery Dataset using the BCI2000 system (EEGBCI) contains 64 channeled EEG signals at 160 Hz over 1500 1- and 2-min recordings from 109 participants. We fetched motor imagery data for classifying two classes from left and right hands, hands and feet-related tasks.
- DEAP (Koelstra et al. 2011): The Database for Emotion Analysis Using Physiological Signals (DEAP) is a large-scale EEG-based emotion dataset, which contains 32-channel EEG signals recorded from 32 participants; each participant watched 40 1-minute-long excerpts of music videos excerpts, annotating continuous valence and arousal ratings on scales from 1 to 9. We grouped pairs of

[1] The repository is called *Mother of all BCI Benchmark*

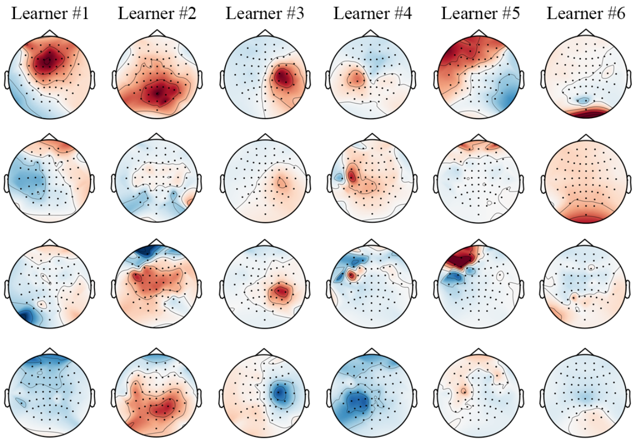


Figure 3: Four common spatial patterns discovered by the six ($K = 6$) learners and their corresponding sub-spaces on the dataset EEGBCI.

continuous labels on the valence and arousal into k discrete states, denoted as DEAP-k. For instance, DEAP-9 comprises nine combinations of negative $(1-3)$, neutral $(4-6)$, and positive $(7-9)$ valence ratings and low $(1-3)$, mid $(4-6)$, and high $(7-9)$ arousal ratings.

Experimental Setup

EEG Preprocessing All EEG signals for each channel are first band-pass filtered with a bandwidth of $4-47$Hz for DEAP and $7-30$Hz for BCICIV and EEGBCI. Then, we standardize the continuous EEG data by computing the electrode-wise exponential moving means and variances. In consequence, each EEG signal is represented by a channel × channel SPD matrix, which is estimated by a second-order statistics with a shrinkage estimator (Chen et al. 2010).

Training, Validation, and Testing Datasets EEG segments were extracted with a sliding time with a width 10 s for DEAP and 2 s from -0.5 to 3.5 s for BCICIV and EEGBCI after cues. Except for the BCICIV dataset, we conducted a 5-fold cross validation, which splits the full datasets into fifths for testing. From the remaining data (four-fifths of the total data), we used one-fifth of the remaining data for validation and four-fifths for training. Because BCICIV already separates the training and testing sessions, the training session was split into fifths for validation. The remaining data (four-fifths) were used as a training set. Note that the training, validation, and testing data were subject-independent. We report mean average precision (mAP) results from all participants.

Network Configuration and Parameter Settings For a fair comparison, all methods applied a batch size, learning rate, weight decay, and momentum of 32, 10^{-2}, 10^{-3}, and 0.9, respectively, as the training parameters. The initial weights were set to random semi-orthogonal matrices, and the rectification threshold ϵ was set to 10^{-4}. The separation procedure was initialized randomly with a same amount of data in every C_k. Early-stopping during validation with a fixed patience size was adopted to prevent an overfitting in learning the deep features. All methods including REB have a simplified configuration, which is a block pair of BiMap and ReEig layers ($f_m \rightarrow f_r$). The sizes of the transformation matrices are set to 32×26, 22×18, and 64×56 for DEAP, BCICIV, and EEGBCI, respectively.

Comparative Results

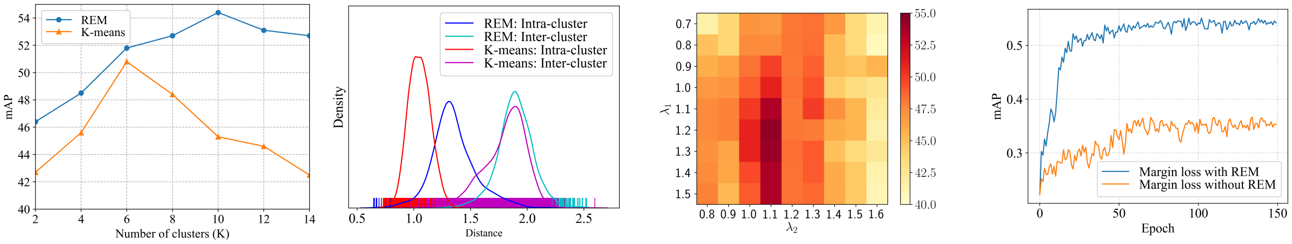
Table 1 reports the comparative performance on the three public datasets. The results from ours with Triplet and Margin loss outperform existing the state-of-the-art methods. This confirms that our approach is universal and can be applied to a variety of metric learning loss functions. Intuitively, all methods combined with the proposed REB have average 14.2% accuracy improvements against the methods under the entire embeddings with random sampling. The triplet loss functions on the datasets had great improvement when it comes with REB. This result supports the significance of mining positive and negative samples within an embedding space. The level of relative similarity in each training triplet determines to generate discriminative features for classifying non-stationary EEG data. The poor performance of the metric learning methods under the entire embedding space implies that the loss may waste a gradient update on SPD matrices far from the decision boundary. DWL yielded the second-best performance in most cases. This observation partially supports the efficacy of sampling negative examples uniformly based on their distances. However, the model had similar performance with other state-of-the-art methods when experimented on the DEAP dataset. The method could not prevail over the inter- and intra-subject variability problem in light of the imbalanced and problematic data distribution.

The results on DEAP-4 and DEAP-9 demonstrate the superiority of REB on unbalanced datasets. Most methods had significant difficulty in learning spatial patterns commonly used to represent SPD features of each class. When the imbalance of data distribution between classes was increased (DEAP-4 vs. DEAP-9), the performance of the state-of-the-art methods was decreased by about 14.86%. On the other hand, the proposed model had small decrements in performance about 7.75%. This result implies that our *separate-to-learn* strategy provides more efficient structures to retrieve informative samples by dividing the entire embedding into multiple groups. The multiple isolated K-assignments prevents the model from being collapsed by low SNR.

Qualitative Results

In addition to the quantitative results, we also demonstrate our model's efficacy in Figure 3, which shows the four common spatial patterns discovered by true positive results from the proposed REB with Margin loss on EEGBCI. We used the CSP algorithm and visualized the patterns using an open-source software[2]. This result indicates that every learner has its own abstract "spatial specialization" on different brain lobes. For instance, the learner 3 and 4 focus on discovering latent spatial factors over the left-right centro-parietal lobes.

[2] https://mne.tools



(a) The number of clusters

(b) Inter- and intra- cluster dist.

(c) The hyper-parameters

(d) mAP curves

Figure 4: Effect of the *Separate-to-Learn* strategy on DEAP-9. (a) mAP results on the different number of clusters. (b) Inter- and intra- cluster distance of different samples. (c) mAP results on different hyper-parameter λ_1 and λ_2. (d) mAP convergence curves on different epochs.

This observation partially supports existing neuroscientific studies that revealed spatial patterns commonly involved in MI tasks (Wolpaw and Wolpaw 2012).

Discussion

We conduct an ablation study to demonstrate the proposed method's effectiveness and evaluate the different components of our contribution. We compare the mAP results with the existing K-means method (Sanakoyeu et al. 2019) on the DEAP-9 validation dataset with Margin loss during 150 epochs.

Effect of the *Separate-to-Learn* Strategy

The effect of the number of K: The correct number of trainers can improve the performance of the EEG classification. As shown in Figure 4a, we observed that as many isolated search areas the model has, the model efficiently learns discriminative features, enhancing the classification performance. Unlike the K-means method's performance has been degraded rapidly after the best setting, our REB had small decrements less than 2%. This observation resonates with the benefits of our clustering selection strategy in the number of trainers.

The effect of the isolation: The *separate-to-learn* strategy yielded better performance than the sampling methods with a unified trainer in Table 1. This observation resonates with the fact that retrieving mini-batches from the individual clusters yields more informative training samples than mini-batches from the entire dataset. Furthermore, Figure 4b shows the efficacy of our isolating approach on the inter- and intra- cluster distance between negative samples compared with the K-means method. Whereas the K-means method focuses on gathering only hard negative samples distributed in a narrow space, our approach contributes to scatter the negative samples more uniformly. This offers a wide range of negative samples in each cluster, and thus steadily produce informative examples while keeping high SNR in SPD matrices. Hence, this isolation enables individuals to learn discriminative features in their way, reducing the complexity of the non-linear learning task.

The effect of the hyper-parameters: The λ variables determine the balance of the cluster and the structures in discriminative feature learning. As shown in Figure 4c, increasing the parameter λ_1 during training would improve the classification performance with the value of λ_2. The performance remains largely stable across a wide range of the parameter λ_1, reducing its fluctuation caused by increasing the parameter λ_2. This observation implies that informative and sound samples, constrained efficiently by mutually non-intervened groups, boost the distinction in metric learning.

Runtime Complexity Separating the full d-dimensional embedding space into d/K-dimensional sub-spaces and assigning the embeddings into K independent trainers can reduce the time required for a single forward and backward propagation. As shown in Figure 4d, the proposed model exhibits a steeper curve compared with others. For the same number of iterations, not only REB takes less overall time, but it also reaches better accuracy much faster[3]. We note that the clustering procedure depends on the number of K, samples, and iterations, but this can be negligible compared to the time required for a full reciprocal step of all signals in datasets as in (Sanakoyeu et al. 2019).

Conclusion

We proposed a *separate-to-learn* method, which partitions and optimizes EEG signals in K clusters and assigns them to individual trainers. The independent learning in multiple trainers is then completed by combining the partial solutions into the final entire embedding. REB can be easily combined with SPD-based neural networks, replacing any last linear embedding layers independent of the loss function's choice. The experimental results on public datasets demonstrated the superiority of REB for discovering common spatial patterns of EEG signals despite their non-stationary nature, increasing the convergence speed while maintaining generalization.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1C1C2012437).

[3] Quad-Core i7 with 64GB memory.

References

Alarcao, S. M.; and Fonseca, M. J. 2019. Emotions Recognition Using EEG Signals: A Survey. *IEEE Transactions on Affective Computing* 10(3): 374–393.

Ang, K. K.; Chin, Z. Y.; Zhang, H.; and Guan, C. 2008. Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2390–2397.

Arsigny, V.; Fillard, P.; Pennec, X.; and Ayache, N. 2006. Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 56(2): 411–421.

Arsigny, V.; Fillard, P.; Pennec, X.; and Ayache, N. 2007. Geometric Means in A Novel Vector Space Structure on Symmetric Positive-Definite Matrices. *SIAM Journal on Matrix Analysis and Applications* 29(1): 328–347.

Barachant, A.; Bonnet, S.; Congedo, M.; and Jutten, C. 2011. Multiclass Brain-Computer Interface Classification by Riemannian Geometry. *IEEE Transactions on Biomedical Engineering* 59(4): 920–928.

Barachant, A.; Bonnet, S.; Congedo, M.; and Jutten, C. 2013. Classification of Covariance Matrices Using a Riemannian-based Kernel for BCI Applications. *Neurocomputing* 112: 172–178.

Barthélemy, Q.; Mayaud, L.; Ojeda, D.; and Congedo, M. 2019. The Riemannian Potato Field: A Tool for Online Signal Quality Index of EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27(2): 244–255.

Brooks, D.; Schwander, O.; Barbaresco, F.; Schneider, J.-Y.; and Cord, M. 2019. Riemannian Batch Normalization for SPD Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 15463–15474.

Chen, Y.; Wiesel, A.; Eldar, Y. C.; and Hero, A. O. 2010. Shrinkage Algorithms for MMSE Covariance Estimation. *IEEE Transactions on Signal Processing* 58(10): 5016–5029.

Cherian, A.; Sra, S.; Banerjee, A.; and Papanikolopoulos, N. 2013. Jensen-Bregman LogDet Divergence with Application to Efficient Similarity Search for Covariance Matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(9): 2161–2174.

Cichocki, A.; Cruces, S.; and Amari, S.-i. 2015. Log-Determinant Divergences Revisited: Alpha-Beta and Gamma Log-Det Divergences. *Entropy* 17(5): 2988–3034.

Congedo, M.; Barachant, A.; and Bhatia, R. 2017. Riemannian Geometry for EEG-based Brain-computer Interfaces; A Primer and a Review. *Brain-Computer Interfaces* 4(3): 155–174.

Harandi, M.; Salzmann, M.; and Hartley, R. 2018. Dimensionality Reduction on SPD manifolds: The Emergence of Geometry-aware Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(1): 48–62.

Harwood, B.; Kumar BG, V.; Carneiro, G.; Reid, I.; and Drummond, T. 2017. Smart Mining for Deep Metric Learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2821–2829.

Huang, Z.; and Van Gool, L. 2017. A Riemannian Network for SPD Matrix Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2036–2042.

Ionescu, C.; Vantzos, O.; and Sminchisescu, C. 2015. Matrix Backpropagation for Deep Networks with Structured Layers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2965–2973.

Jayaram, V.; and Barachant, A. 2018. MOABB: Trustworthy Algorithm Benchmarking for BCIs. *Journal of Neural Engineering* 15(6): 066011.

Kemertas, M.; Pishdad, L.; Derpanis, K. G.; and Fazly, A. 2020. RankMI: A Mutual Information Maximizing Ranking Loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 14362–14371.

Kim, B. H.; and Jo, S. 2020. Deep Physiological Affect Network for the Recognition of Human Emotions. *IEEE Transactions on Affective Computing* 11(2): 230–243.

Koelstra, S.; Muhl, C.; Soleymani, M.; Lee, J.-S.; Yazdani, A.; Ebrahimi, T.; Pun, T.; Nijholt, A.; and Patras, I. 2011. Deap: A Database for Emotion Analysis; Using Physiological Signals. *IEEE Transactions on Affective Computing* 3(1): 18–31.

Kulis, B.; Sustik, M.; and Dhillon, I. 2006. Learning Low-rank Kernel Matrices. In *Proceedings of the International Conference on Machine Learning (ICML)*, 505–512.

Lee, J. M. 2006. *Riemannian Manifolds: An Introduction to Curvature*, volume 176. Springer Science & Business Media.

Liu, H.; Cao, Z.; Long, M.; Wang, J.; and Yang, Q. 2019a. Separate to Adapt: Open Set Domain Adaptation via Progressive Separation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2927–2936.

Liu, H.; Li, J.; Wu, Y.; and Ji, R. 2019b. Learning Neural Bag-of-Matrix-Summarization with Riemannian Network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 8746–8753.

Nair, V.; and Hinton, G. E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 807–814.

Olias, J.; Martín-Clemente, R.; Sarmiento-Vega, M. A.; and Cruces, S. 2019. EEG Signal Processing in MI-BCI Applications with Improved Covariance Matrix Estimators. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27(5): 895–904.

Opitz, M.; Waltner, G.; Possegger, H.; and Bischof, H. 2020. Deep Metric Learning with BIER: Boosting Independent Embeddings Robustly. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42(2): 276–290.

Pennec, X.; Fillard, P.; and Ayache, N. 2006. A Riemannian Framework for Tensor Computing. *International Journal of Computer Vision* 66(1): 41–66.

Pennec, X.; Sommer, S.; and Fletcher, T. 2019. *Riemannian Geometric Statistics in Medical Image Analysis*. Academic Press.

Rodrigues, P. L. C.; Jutten, C.; and Congedo, M. 2018. Riemannian Procrustes Analysis: Transfer Learning for Brain–Computer Interfaces. *IEEE Transactions on Biomedical Engineering* 66(8): 2390–2401.

Roth, K.; Milbich, T.; Sinha, S.; Gupta, P.; Ommer, B.; and Cohen, J. P. 2020. Revisiting Training Strategies and Generalization Performance in Deep Metric Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 8242–8252.

Sanakoyeu, A.; Tschernezki, V.; Buchler, U.; and Ommer, B. 2019. Divide and Conquer the Embedding Space for Metric Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 471–480.

Schalk, G.; McFarland, D. J.; Hinterberger, T.; Birbaumer, N.; and Wolpaw, J. R. 2004. BCI2000: A General-purpose Brain-Computer Interface (BCI) System. *IEEE Transactions on Biomedical Engineering* 51(6): 1034–1043.

Song, H. O.; Jegelka, S.; Rathod, V.; and Murphy, K. 2017. Deep Metric Learning via Facility Location. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5382–5390.

Sra, S. 2012. A New Metric on The Manifold of Kernel Matrices with Application to Matrix Geometric Means. In *Advances in Neural Information Processing Systems (NeurIPS)*, 144–152.

Stanitsas, P.; Cherian, A.; Morellas, V.; and Papanikolopoulos, N. 2017. Clustering Positive Definite Matrices by Learning Information Divergences. In *Proceedings of the International Conference on Computer Vision (ICCV) Workshops*, 1304–1312.

Tangermann, M.; Müller, K.-R.; Aertsen, A.; Birbaumer, N.; Braun, C.; Brunner, C.; Leeb, R.; Mehring, C.; Miller, K. J.; Mueller-Putz, G.; et al. 2012. Review of the BCI competition IV. *Frontiers in Neuroscience* 6: 55.

Ustinova, E.; and Lempitsky, V. 2016. Learning Deep Embeddings with Histogram Loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 4170–4178.

Wang, X.; Hua, Y.; Kodirov, E.; Hu, G.; Garnier, R.; and Robertson, N. M. 2019. Ranked List Loss for Deep Metric Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5207–5216.

Wolpaw, J.; and Wolpaw, E. W. 2012. *Brain-Computer Interfaces: Principles and Practice*. OUP USA.

Wu, C.-Y.; Manmatha, R.; Smola, A. J.; and Krahenbuhl, P. 2017. Sampling Matters in Deep Embedding Learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2840–2848.

Yair, O.; Ben-Chen, M.; and Talmon, R. 2019. Parallel Transport on the Cone Manifold of SPD matrices for Domain Adaptation. *IEEE Transactions on Signal Processing* 67(7): 1797–1811.

Yuan, T.; Deng, W.; Tang, J.; Tang, Y.; and Chen, B. 2019. Signal-to-Noise Ratio: A Robust Distance Metric for Deep Metric Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4815–4824.

Yuan, Y.; Yang, K.; and Zhang, C. 2017. Hard-aware Deeply Cascaded Embedding. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 814–823.

Zheng, L.; Qiu, G.; and Huang, J. 2018. Riemannian Competitive Learning for Symmetric Positive Definite Matrices Clustering. *Neurocomputing* 295: 153–164.