This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning to Normalize on the SPD Manifold under Bures-Wasserstein Geometry

Rui Wang[1]*, Shaocheng Jin[1]*, Ziheng Chen[2]†, Xiaoqing Luo[1], Xiao-Jun Wu[1]
[1]School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China
[2]Department of Information Engineering and Computer Science, University of Trento, Trento, Italy
{cs_wr, xqluo, wu_xiaojun}@jiangnan.edu.cn
{6233112021}@stu.jiangnan.edu.cn, ziheng_ch@163.com

Abstract

Covariance matrices have proven highly effective across many scientific fields. Since these matrices lie within the Symmetric Positive Definite (SPD) manifold—a Riemannian space with intrinsic non-Euclidean geometry, the primary challenge in representation learning is to respect this underlying geometric structure. Drawing inspiration from the success of Euclidean deep learning, researchers have developed neural networks on the SPD manifolds for more faithful covariance embedding learning. A notable advancement in this area is the implementation of Riemannian batch normalization (RBN), which has been shown to improve the performance of SPD network models. Nonetheless, the Riemannian metric beneath the existing RBN might fail to effectively deal with the ill-conditioned SPD matrices (ICSM), undermining the effectiveness of RBN. In contrast, the Bures-Wasserstein metric (BWM) demonstrates superior performance for ill-conditioning. In addition, the recently introduced Generalized BWM (GBWM) parameterizes the vanilla BWM via an SPD matrix, allowing for a more nuanced representation of vibrant geometries of the SPD manifold. Therefore, we propose a novel RBN algorithm based on the GBW geometry, incorporating a learnable metric parameter. Moreover, the deformation of GBWM by matrix power is also introduced to further enhance the representational capacity of GBWM-based RBN. Experimental results on different datasets validate the effectiveness of our proposed method. The code is available at https://github.com/jjscc/GBWBN.

1. Introduction

Covariance matrices are prevalent in numerous scientific fields. Due to their capability to capture high-order spatiotemporal fluctuations in data, they have proven to be highly effective in various applications, including Brain-Computer Interfaces (BCI) [3, 32, 46], Magnetic Resonance Imaging (MRI) [10, 19, 47], drone recognition [7, 12, 14], node classification [11, 15, 31, 41], and action recognition [16, 18, 24, 27, 44, 56, 57]. However, the geometry of non-singular covariance matrices doesn't conform to that of a vector space; instead, it forms a Riemannian manifold, known as the SPD manifold. This inherent difference prevents the direct application of Euclidean learning methods to such data. To bridge this gap, several Riemannian metrics have been developed, such as the Log-Euclidean Metric (LEM) [2], Log-Cholesky Metric (LCM) [35], and Affine Invariant Metric (AIM) [47], extending Euclidean computations to the SPD manifold.

Deep Neural Networks (DNNs) [25, 51] have achieved remarkable breakthroughs over traditional shallow learning models in the field of pattern recognition and computer vision, primarily due to their ability to learn deep, nonlinear representations. Building on this success, some researchers have generalized the ideology of DNNs to Riemannian manifolds. Notably, SPDNet [27] stands out as a classical Riemannian neural network, providing a multi-stage, end-to-end nonlinear learning framework specifically designed for SPD matrices. In addition, numerous Euclidean-based model architectures, including Recurrent Neural Network (RNN) [9, 42], Convolutional Neural Network (CNN) [10], and Transformers [59], have been successfully adapted to operate on manifolds. In parallel, researchers have also investigated the fundamental building blocks necessary for Riemannian networks, such as pooling [17, 55], classification [13, 43], and transformation [43, 44].

Traditional Batch Normalization (BatchNorm) has been instrumental in facilitating network training [29], which motivates the development of Riemannian BatchNorm (RBN) over the SPD manifold. Among them, Brooks et al. [7] introduced RBN for SPD networks, utilizing the AIM geometry. Kobler et al. [32] further advanced this methodology by extending control over the Riemannian variance. In parallel, Chakraborty [8] proposed two RBN frameworks and

*Equal contribution.
†Corresponding Author

Dataset	$\kappa > 10^3$	$\kappa > 10^5$
MAMEM-SSVEP-II	113 (22.6%)	11 (2.2%)
HDM05	2086 (100%)	2086 (100%)
NTU RGB+D	56,880 (100%)	56,880 (100%)

Table 1. Statistics (number and percentage) on the condition number ($\kappa = \mathrm{eig}_{\max}/\mathrm{eig}_{\min}$) of the SPD features without the RBN layer on different datasets. The statistics are summarized in the final epoch. This is a teaser table from Tab. 7.

instantiated them on the SPD manifold under LEM and AIM geometries by incorporating first- and second-order statistics. More recently, Chen et al. [14] developed the RBN over Lie groups, specifically showcasing their framework on the SPD manifold under the AIM, LEM, and LCM geometries.

However, the latent metric geometries beneath the existing RBN might be ineffective in dealing with ill-conditioned SPD matrices (ICSM), which are yet prevalent in SPD modeling. Specifically, as covariance matrices are positive semi-definite, the following regularization is widely adopted: $\mathbf{X} \leftarrow \mathbf{X} + \lambda \mathbf{I}_d$, where $\mathbf{I}_d$ is a d-by-d identity matrix and λ denotes a perturbation hyperparameter. Although such a strategy ensures the positive definite property of $\mathbf{X}$, it offers only limited improvement in alleviating ill-conditioning. For any SPD matrix $\mathbf{X}$, the condition number can be characterized as $\kappa(\mathbf{X}) = {}^{\mathrm{eig}_{\max}}/_{\mathrm{eig}_{\min}}$, where $\mathrm{eig}_{\max}$ and $\mathrm{eig}_{\min}$ denotes its maximal and minimal eigenvalues. Therefore, the post-regularization SPD matrix might be ill-conditioned. **Tab. 1 summarizes the ill-conditioning observed across all three used datasets, highlighting its widespread presence**.

Han et al. [21] show that the AIM exhibits a quadratic dependence on SPD matrices, which hinders effective learning over ICSM. In contrast, the BWM [5] has a linear dependence, offering advantages for ill-conditioning. Thereby, we resort to BWM for RBN. Besides, Han et al. [22] further updated BWM into a generalized version, named GBWM, by an SPD parameter. The GBWM is correlated with the AIM, as it is locally AIM [22]. By setting the SPD parameter in GWBM to be learnable, the proposed BWM-based RBN algorithm is further extended into the one based on the GBWM. Compared with the existing methods, our method can not only better deal with ICSM, but also adapts to the evolving latent SPD geometry through the dynamic geometric parameter. Besides, we use the deformation of matrix power [53] to further improve the representational capacity of the suggested GBWM-based RBN. For simplicity, we refer to the proposed method as GBWBN. Tab. 2 summarizes the existing RBN methods compared with ours. The effectiveness of our GBWBN is validated across three different benchmarking datasets involving two signal classification tasks, *i.e.*, skeleton-based action recognition and EEG classification. In summary, our main contributions are as follows:

- **Challenging ICSM, effective countermeasure:** A new RBN algorithm based on the BW geometry—the first work capable of effectively addressing ICSM learning;
- **Learnable normalization, flexible method:** Our BWM-based RBN is extended to a learnable version incorporated with an SPD parameter and a matrix power-based nonlinear deformation of the metric geometry, aiming to generate a flexible and refined normalization space;
- **Experimental validity, functional pluggability:** Extensive empirical evaluations of our GBWBN on three benchmarking datasets under different SPD backbone networks.

Methods	Involved Statistics	Geometries
SPDBN [7]	M	AIM
SPDDSMBN [32]	M+V	AIM
ManifoldNorm [8, Algs. 1-2]	M+V	AIM and LEM
LieBN [14]	M+V	Power-deformed AIM, LEM, and LCM
GBWBN (Ours)	M+V	Power-deformed GBWM

Table 2. Summary of several representative RBN methods on the SPD manifold, where M and V represent the mean and variance.

2. Preliminaries

The set of SPD matrices, denoted as $\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++} := \{\mathbf{X} \in \mathbb{R}^{d\times d}, \mathbf{X} = \mathbf{X}^{\mathrm{T}}, \boldsymbol{v}^{\mathrm{T}}\mathbf{X}\boldsymbol{v} > 0, \forall \boldsymbol{v} \in \mathbb{R}^d \backslash \{0_d\}\}$, forms a smooth manifold, known as the SPD manifold. The tangent space at $\mathbf{X}$ is denoted as $T_{\mathbf{X}}\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++} \cong \boldsymbol{\mathcal{S}}^{\boldsymbol{d}}$, where $\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}$ is the Euclidean space of real symmetric matrices. Given $\mathbf{X} \in \boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$, the Riemannian metric at $\mathbf{X}$ is denoted as: $g_{\mathbf{X}} : T_{\mathbf{X}}\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++} \times T_{\mathbf{X}}\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++} \to \mathbb{R}$, which is often written as an inner product $\langle \cdot, \cdot \rangle_{\mathbf{X}}$. The induced norm of a tangent vector $\mathbf{S} \in T_{\mathbf{X}}\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$ is given by $\|\mathbf{S}\|_{\mathbf{X}} = \sqrt{\langle \mathbf{S}, \mathbf{S} \rangle_{\mathbf{X}}}$. Given $\mathbf{X}_1, \mathbf{X}_2 \in \boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$, $\mathbf{S}_1, \mathbf{S}_2 \in T_{\mathbf{X}_1}\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$, the Riemannian operators under BWM are shown in Tab. 3. Wherein, the $\mathrm{tr}(\mathbf{S})$ denotes the matrix trace of $\mathbf{S}$, $\mathrm{vec}(\mathbf{S}_1)$ and $\mathrm{vec}(\mathbf{S}_2)$ are the vectorizations of $\mathbf{S}_1$ and $\mathbf{S}_2$, $\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_i, \boldsymbol{\delta}_i, \boldsymbol{\delta}_j$ represent the i-th and j-th eigenvalues obtained by SVD of $\mathbf{X}_1$ and $\mathbf{X}_2$ respectively. $\mathbf{X} \oplus \mathbf{X} = \mathbf{X} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{X}$ denotes the Kronecker sum and $\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)$ denotes the Lyapunov operator, *i.e.*, $\mathbf{X}_1\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1) + \mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)\mathbf{X}_1 = \mathbf{S}_1$. The Lyapunov operator can be solved by eigendecomposition [21]:

$$\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1) = \mathbf{V}\left[\frac{\mathbf{S'}_{\mathbf{1}i,j}}{\boldsymbol{\delta}_i + \boldsymbol{\delta}_j}\right]_{i,j}\mathbf{V}^T, \tag{1}$$

where $\mathbf{S'}_1 = \mathbf{V}^T\mathbf{S}_1\mathbf{V}$, $\mathbf{X}_1 = \mathbf{V}\boldsymbol{\Delta}\mathbf{V}^T$ is the eigendecomposition of $\mathbf{X}_1 \in \boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$.

Although $\mathbf{X}_1\mathbf{X}_2$ is not SPD, the following equation of $(\mathbf{X}_1\mathbf{X}_2)^{\frac{1}{2}}$ makes each (inverse) square root can be computed using SVD [38].

$$(\mathbf{X}_1\mathbf{X}_2)^{\frac{1}{2}} = \mathbf{X}_1^{\frac{1}{2}}(\mathbf{X}_1^{\frac{1}{2}}\mathbf{X}_2\mathbf{X}_1^{\frac{1}{2}})^{\frac{1}{2}}\mathbf{X}_1^{-\frac{1}{2}}. \tag{2}$$

The geodesic is not complete on the Riemannian manifold $(\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}g^{\mathrm{BW}})$ [36]. In Tab. 3, the variable t in the geodesic

Riemannian operators under the BWM
Riemannian metric
$g^{\mathrm{BW}}_{\mathbf{X}_1}(\mathbf{S}_1,\mathbf{S}_2) = \frac{1}{2}\mathrm{tr}\left(\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)\mathbf{S}_2\right)$ $= \frac{1}{2}\mathrm{vec}(\mathbf{S}_1)^T(\mathbf{X}_1 \oplus \mathbf{X}_1)^{-1}\mathrm{vec}(\mathbf{S}_2)$
Riemannian distance
$d^2_{\mathrm{BW}}(\mathbf{X}_1,\mathbf{X}_2) = \mathrm{tr}(\mathbf{X}_1)+\mathrm{tr}(\mathbf{X}_2)-2\left(\mathrm{tr}\left(\mathbf{X}_1^{\frac{1}{2}}\mathbf{X}_2\mathbf{X}_1^{\frac{1}{2}}\right)\right)^{\frac{1}{2}}$
Riemannian geodesic
$\gamma_{\mathbf{X}_1,\mathbf{S}_1}(t) = \mathbf{X}_1 + t\mathbf{S}_1 + t^2\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)\mathbf{X}_1\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)$
Riemannian exponential map
$\mathrm{Exp}_{\mathbf{X}_1}\mathbf{S}_1 = \mathbf{X}_1 + \mathbf{S}_1 + \mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)\mathbf{X}_1\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1)$
Riemannian logarithmic map
$\mathrm{Log}_{\mathbf{X}_1}\mathbf{X}_2 = (\mathbf{X}_2\mathbf{X}_1)^{\frac{1}{2}} + (\mathbf{X}_1\mathbf{X}_2)^{\frac{1}{2}} - 2\mathbf{X}_1$
Parallel Transport
$\Gamma_{\mathbf{X}_1\to\mathbf{X}_2}(\mathbf{S}) = \mathbf{U}\left[\sqrt{\frac{\boldsymbol{\delta}_i+\boldsymbol{\delta}_j}{\boldsymbol{\lambda}_i+\boldsymbol{\lambda}_j}}\left[\mathbf{S}'\right]_{i,j}\right]_{i,j}\mathbf{U}^T$

Table 3. Basic operators based on the BWM [5].

depends on as follows: $\mathbf{I}_d + t\mathcal{L}_{\mathbf{X}_1}(\mathbf{S}_1) \in \boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$. For parallel transport, $\mathbf{X}_1, \mathbf{X}_2$ must be commutative matrices [54].

For any two $\mathbf{S}_1, \mathbf{S}_2 \in T_{\mathbf{X}_1}\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$, the AIM on the SPD manifold can be rewritten as:

$$\begin{aligned} g^{\mathrm{AI}}_{\mathbf{X}_1}(\mathbf{S}_1,\mathbf{S}_2) &= \mathrm{tr}(\mathbf{X}_1^{-1}\mathbf{S}_1\mathbf{X}_1^{-1}\mathbf{S}_2) \\ &= \mathrm{vec}(\mathbf{S}_1)^T(\mathbf{X}_1 \otimes \mathbf{X}_1)^{-1}\mathrm{vec}(\mathbf{S}_2), \end{aligned} \tag{3}$$

where $\otimes$ denotes the Kronecker product. Comparing the Riemannian metric based on the BWM in Tab. 3 and Eq. (3), the AIM has a quadratic dependence on $\mathbf{X}_1$ while the BWM has a linear dependence [21], revealing the BWM is more suitable for learning ICSM.

For any two $\mathbf{S}_1, \mathbf{S}_2 \in T_{\mathbf{X}}\boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$, the GBWM can be written as [22]

$$\begin{aligned} g^{\mathrm{GBW}}_{\mathbf{X}}(\mathbf{S}_1,\mathbf{S}_2) &= \frac{1}{2}\mathrm{tr}\left(\mathcal{L}_{\mathbf{X},\mathbf{M}}(\mathbf{S}_1)\,\mathbf{S}_2\right) \\ &= \frac{1}{2}\mathrm{vec}(\mathbf{S}_1)^T(\mathbf{X}\otimes\mathbf{M} + \mathbf{M}\otimes\mathbf{X})^{-1}\mathrm{vec}(\mathbf{S}_2), \end{aligned} \tag{4}$$

where $\mathbf{M} \in \boldsymbol{\mathcal{S}}^{\boldsymbol{d}}_{++}$, $\mathcal{L}_{\mathbf{X},\mathbf{M}}(\mathbf{S})$ is the generalized Lyapunov operator, which is the solution to the matrix equation $\mathbf{X}\mathcal{L}_{\mathbf{X},\mathbf{M}}(\mathbf{S})\mathbf{M} + \mathbf{M}\mathcal{L}_{\mathbf{X},\mathbf{M}}(\mathbf{S})\mathbf{X} = \mathbf{S}, \mathbf{S} \in \boldsymbol{\mathcal{S}}^{\boldsymbol{d}}$. Choosing $\mathbf{M}$ is equivalent to selecting a suitable metric. When $\mathbf{M} = \mathbf{I}$, it reduces to the BWM. When $\mathbf{M} = \mathbf{X}$, it coincides locally with the AIM [22]. Consequently, it connects the BWM and AIM (locally).

3. Proposed method

In this section, we begin by introducing RBN based on the BWM, from which the GBWM-based batch normalization is derived. This foundation enables a smooth extension of RBN to integrate the broader capabilities of GBWM. Due to page limitation, all the proofs are placed in App. D.

3.1. RBN based on the BWM

This subsection begins with an introduction to the BWM-based Riemannian mean. We then generalize the parallel transport, extending it from tangent vectors to data points on the SPD manifold. Following this, we detail the data scaling process. Finally, the aforementioned components are integrated to achieve manifold-valued batch normalization.

3.1.1. Riemannian mean and variance

Let $\mathbf{X}_1, \ldots, \mathbf{X}_N$ denote N points on the SPD manifold, $\boldsymbol{\Omega} = [\omega_1, \ldots, \omega_N]$ be a weight vector, satisfying $\omega_i \geq 0$ and $\sum_i \omega_i = 1$. Denoting the weighted Fréchet mean (WFM) as $\mathbb{E}^{\boldsymbol{\Omega}}_N(\{\mathbf{X}_i\}_{i\leq N})$, the Riemannian barycenter, signified as $\boldsymbol{\mathcal{B}}$, we can have the following:

$$\boldsymbol{\mathcal{B}} = \mathbb{E}^{\boldsymbol{\Omega}}_N(\{\mathbf{X}_i\}_{i\leq N}) = \underset{\mathbf{G}\in\boldsymbol{\mathcal{S}}^d_{++}}{\arg\min}\sum_{i=1}^{N}\omega_i d^2_{\mathrm{BW}}(\mathbf{G},\mathbf{X}_i). \tag{5}$$

When $\omega_i = \frac{1}{N}$, Eq. (5) simplifies to the Fréchet mean(FM). The Fréchet variance $\boldsymbol{\nu}^2$ is the value achieved at the minimizer of the Fréchet mean:

$$\boldsymbol{\nu}^2 = \frac{1}{N}\sum_{i=1}^{N} d^2_{\mathrm{BW}}(\boldsymbol{\mathcal{B}},\mathbf{X}_i). \tag{6}$$

In this paper, we use the terms Fréchet mean signifying Riemannian mean, and Fréchet variance signifying Riemannian variance. When $N = 2$, *i.e.,* $\boldsymbol{\Omega} = (1-\omega,\omega), \omega \in [0,1]$, a closed-formed solution of Eq. (5) exists, corresponding to the geodesic connecting $\mathbf{X}_1$ and $\mathbf{X}_2$:

$$\begin{aligned} \mathbb{E}^{\boldsymbol{\Omega}}_2(\mathbf{X}_1,\mathbf{X}_2) = (1-\omega)^2\mathbf{X}_1 + \omega^2\mathbf{X}_2 + \\ \omega(1-\omega)\left((\mathbf{X}_2\mathbf{X}_1)^{\frac{1}{2}} + (\mathbf{X}_1\mathbf{X}_2)^{\frac{1}{2}}\right). \end{aligned} \tag{7}$$

As shown in [5], the minimized solution to Eq. (5) is unique, which is also a solution to a non-linear equation:

$$\mathbf{G} = \sum_{i=1}^{N}\omega_i(\mathbf{G}^{\frac{1}{2}}\mathbf{X}_i\mathbf{G}^{\frac{1}{2}})^{\frac{1}{2}}. \tag{8}$$

When $N > 2$, the solution to Eq. (8) can be obtained by a novel iterative method, called fixed-point iteration [5]:

$$\mathrm{H}(\mathbf{G}) = \mathbf{G}^{-\frac{1}{2}}\left(\sum_{i=1}^{N}\omega_i(\mathbf{G}^{\frac{1}{2}}\mathbf{X}_i\mathbf{G}^{\frac{1}{2}})^{\frac{1}{2}}\right)^2\mathbf{G}^{-\frac{1}{2}}. \tag{9}$$

Its update role is $\mathbf{G}_{t+1} = \mathrm{H}(\mathbf{G}_t)$ and converges to $\boldsymbol{\mathcal{B}}$, *i.e.*, $\lim_{t\to\infty}\mathbf{G}_t = \boldsymbol{\mathcal{B}}$. Considering that the batch barycenter is a noisy estimation of the true barycenter, a relaxed approximation is sufficient [7, 14]. Thereby, we set the number of iterations to one for computational efficiency. Furthermore, although $\mathbf{G}_t$ can be initialized as any arbitrary SPD matrix, it is reasonable to endow it with the arithmetic mean.

3.1.2. Centering data via parallel transport

The conventional batch normalization in Euclidean space typically involves centering and biasing a set of samples using vector subtraction and addition. In contrast, there is no vector structure for a curved Riemannian manifold. Given a batch of data points on the SPD manifold, a viable and effective method to relocate features around the batch mean $\boldsymbol{\mathcal{B}}$ or bias them towards a parameter $\boldsymbol{\mathcal{G}}$ is to employ parallel transport (PT).

Drawing inspiration from [7], we extend the tangent space PT given in Tab. 3 to its manifold counterpart. Let commuting matrices $\mathbf{X}_1, \mathbf{X}_2 \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, for any $\mathbf{P} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, the PT of $\mathbf{P}$ from $\mathbf{X}_1$ to $\mathbf{X}_2$ can be realized via the following manifold map $\phi : \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}} \to \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$:

$$\phi(\mathbf{P}) = \mathrm{Exp}_{\mathbf{X}_2}\left(\Gamma_{\mathbf{X}_1 \to \mathbf{X}_2}\left(\mathrm{Log}_{\mathbf{X}_1}(\mathbf{P})\right)\right). \tag{10}$$

3.1.3. Batch normalization

In deep learning, batch normalization is a widely used technique to speed up convergence and mitigate internal covariate shifts [29]. The standard batch normalization layer applies slightly different transformations to the data during the training and testing. In the training phase, normalization is performed using the batch mean $\boldsymbol{\mathcal{B}}_b$ and variance $\boldsymbol{\nu}_b^2$. Meanwhile, the running statistics—mean $\boldsymbol{\mathcal{B}}_r$ and variance $\boldsymbol{\nu}_r^2$—are iteratively updated, where they are applied for normalization during testing.

Given a batch of samples $\{\mathbf{X}_i\}_{i=1}^N$, the batch normalization based on the BWM is defined as follows:

• Data centralizing on the SPD manifold, *i.e.*, shifting $\{\mathbf{X}_i\}_{i=1}^N$ with Riemannian mean $\boldsymbol{\mathcal{B}}$ to the identity matrix $\mathbf{I}_d$ is given by

$$\bar{\mathbf{X}}_i = \phi_{\boldsymbol{\mathcal{B}} \to \mathbf{I}_d}(\mathbf{X}_i) = \mathrm{Exp}_{\mathbf{I}_d}\left(\Gamma_{\boldsymbol{\mathcal{B}} \to \mathbf{I}_d}\left(\mathrm{Log}_{\boldsymbol{\mathcal{B}}}\left(\mathbf{X}_i\right)\right)\right). \tag{11}$$

• Then, data scaling can be defined as follows:

$$\check{\mathbf{X}}_i = \mathrm{Exp}_{\mathbf{I}_d}\left(\frac{\mathbf{s}}{\sqrt{\boldsymbol{\nu}^2 + \boldsymbol{\epsilon}}}\left(\mathrm{Log}_{\mathbf{I}_d}(\bar{\mathbf{X}}_i)\right)\right), \tag{12}$$

where $\boldsymbol{\nu}^2$ denotes the Fréchet variance, $\mathbf{s} \in \mathbb{R}\backslash\{0\}$ is a scaling factor, and $\boldsymbol{\epsilon}$ is a small positive constant.

• Finally, data biasing is conducted on the SPD manifold, transferring a set of samples from the point $\mathbf{I}_d$ to the target point $\boldsymbol{\mathcal{G}}$, as detailed below:

$$\tilde{\mathbf{X}}_i = \phi_{\mathbf{I}_d \to \boldsymbol{\mathcal{G}}}(\check{\mathbf{X}}_i) = \mathrm{Exp}_{\boldsymbol{\mathcal{G}}}\left(\Gamma_{\mathbf{I}_d \to \boldsymbol{\mathcal{G}}}\left(\mathrm{Log}_{\mathbf{I}_d}(\check{\mathbf{X}}_i)\right)\right). \tag{13}$$

3.1.4. Data scaling

On $\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, several notions of Gaussian density have been proposed [4, 48]. In this work, we adopt the Gaussian distribution on $\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$ with a mean $\boldsymbol{\mathcal{B}} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$ and variance $\boldsymbol{\sigma}^2$ from [48], which is defined as:

$$p(\mathbf{X} \mid \boldsymbol{\mathcal{B}}, \boldsymbol{\sigma}^2) = \frac{1}{\zeta(\boldsymbol{\sigma})} \exp\left(-\frac{d(\mathbf{X}, \boldsymbol{\mathcal{B}})^2}{2\boldsymbol{\sigma}^2}\right), \tag{14}$$

where $\zeta(\boldsymbol{\sigma})$ is the normalization constant.

Proposition 3.1. *Given N SPD matrices $\{\mathbf{X}_i\}_{i=1}^N$, $\mathbf{s} \in \mathbb{R}\backslash\{0\}$, defining $\boldsymbol{\psi}_{\mathbf{s}}(\mathbf{X}_i) = \mathrm{Exp}_{\mathbf{I}_d}\left(\mathbf{s}\mathrm{Log}_{\mathbf{I}_d}(\mathbf{X}_i)\right) = \left(\mathbf{s}(\mathbf{X}^{\frac{1}{2}} - \mathbf{I}_d) + \mathbf{I}_d\right)^2$, we control the dispersion from $\mathbf{I}_d$:*

$$\sum_{i=1}^N \omega_i d_{\mathrm{BW}}\left(\boldsymbol{\psi}_{\mathbf{s}}(\mathbf{X}_i), \mathbf{I}_d\right) = \mathbf{s} \sum_{i=1}^N \omega_i d_{\mathrm{BW}}(\mathbf{X}_i, \mathbf{I}_d), \tag{15}$$

where $\{\omega_i\}_{i=1}^N$ are weights satisfying $\omega_i \geq 0$, $\sum_i \omega_i = 1$.

Proof. The proof is presented in App. D.1. □

Therefore, data scaling can be achieved by scaling data on the tangent space at the Riemannian mean.

3.2. RBN based on the GBWM

We generalize the BWM-based RBN and introduce a learnable version that implements RBN under the GBWM.

3.2.1. Power-deformed GBWM

Previous works have extended the AIM, LEM, LCM, and BWM to power-deformed metrics [14, 52, 53] using the diffeomorphism matrix power transformation $\phi_\theta(\mathbf{X}) = \mathbf{X}^\theta, \forall \mathbf{X} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$. Intuitively, the matrix power parameter θ allows for interpolation between the original metric ($\theta = 1$) and a LEM-like metric ($\theta \to 0$) [53]. Inspired by the flexibility of the power function, we suggest the power-deformed GBWM, denoted as (θ)-GBWM. For any $\mathbf{X} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$ and $\mathbf{S}_1, \mathbf{S}_2 \in T_{\mathbf{X}}\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, it can be formulated as:

$$g_{\mathbf{X}}^{(\theta)-\mathrm{GBW}}(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{\theta^2} g_{\mathbf{X}^\theta}^{\mathrm{GBW}}\left(\left(\phi_\theta\right)_{*,\mathbf{X}}(\mathbf{S}_1), \left(\phi_\theta\right)_{*,\mathbf{X}}(\mathbf{S}_2)\right), \tag{16}$$

where ϕ_θ signifies the matrix power, and $(\phi_\theta)_{*,\mathbf{X}}(\cdot)$ denotes the differential map of ϕ_θ at $\mathbf{X}$.

Proposition 3.2 (Deformation)**.** *When $\theta \to 0$, $\forall\, \mathbf{X} \in \mathcal{S}_{++}^d$ and $\forall\, \mathbf{S} \in T_{\mathbf{X}}\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, the following can be derived:*

$$g_{\mathbf{X}}^{(\theta)-\mathrm{GBW}}(\mathbf{S}, \mathbf{S}) \overset{\theta \to 0}{\longrightarrow} \frac{1}{2}\left\langle \log_{*,\mathbf{X}}\left(\mathcal{L}_{\mathbf{M}}(\mathbf{S})\right), \log_{*,\mathbf{X}}(\mathbf{S})\right\rangle, \tag{17}$$

where the right side can be viewed as a variant of LEM.

Proof. The proof is presented in App. D.2. □

The method studied in [53] extended AIM into power-deformed metric, denoted as (θ)-AIM. We find that (θ)-GBWM is locally power-deformed (θ)-AIM, which is detailed in the following proposition.

Proposition 3.3 (Locally Deformed AIM)**.** *For any $\mathbf{X} \in \mathcal{S}_{++}^d$ and $\mathbf{S}_1, \mathbf{S}_2 \in T_{\mathbf{X}}\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, we have the following:*

$$g_{\mathbf{X}}^{(\theta)-\mathrm{GBW}}(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{4} g_{\mathbf{X}}^{(\theta)-\mathrm{AI}}(\mathbf{S}_1, \mathbf{S}_2). \tag{18}$$

Algorithm 1: Batch normalization on the SPD manifold under deformed GBWM

Input : batches of N SPD matrices $\{\mathbf{X}_i\}_{i=1}^N$, small positive constant $\boldsymbol{\epsilon}$, running mean $\boldsymbol{\mathcal{B}}_r = \mathbf{I}_d$, running variance $\boldsymbol{\nu}_r^2 = 1$, power parameter $\theta \in \mathbb{R}\backslash\{0\}$, momentum ω, a learnable parameter $\mathbf{M} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, bias $\boldsymbol{\mathcal{G}} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, and scaling parameter $\mathbf{s} \in \mathbb{R} \setminus \{0\}$.

Output : normalized batch $\{\hat{\mathbf{X}}_i\}_{i=1}^N$.

$\hat{\mathbf{X}}_i = \mathbf{M}^{-\frac{1}{2}}\mathbf{X}_i^{\theta}\mathbf{M}^{-\frac{1}{2}}$, $\hat{\boldsymbol{\mathcal{G}}}_i = \mathbf{M}^{-\frac{1}{2}}\boldsymbol{\mathcal{G}}^{\theta}\mathbf{M}^{-\frac{1}{2}}$

if *training* **then**

  $\boldsymbol{\mathcal{B}}_b \leftarrow \mathbb{E}_N(\{\hat{\mathbf{X}}_i\}_{i=1}^N), \boldsymbol{\nu}_b^2 \leftarrow \frac{1}{N}\sum_{i=1}^N \frac{1}{\theta^2} d_{\mathrm{BW}}^2(\boldsymbol{\mathcal{B}}_b, \hat{\mathbf{X}}_i)$;

  $\boldsymbol{\mathcal{B}}_r \leftarrow \mathbb{E}_2^{\Omega}(\boldsymbol{\mathcal{B}}_r, \boldsymbol{\mathcal{B}}_b), \boldsymbol{\nu}_r^2 \leftarrow (1-\omega)\boldsymbol{\nu}_r^2 + \omega\boldsymbol{\nu}_b^2$;

end

if *training* **then** $\boldsymbol{\mathcal{B}} \leftarrow \boldsymbol{\mathcal{B}}_b, \boldsymbol{\nu}^2 \leftarrow \boldsymbol{\nu}_b^2$

else $\boldsymbol{\mathcal{B}} \leftarrow \boldsymbol{\mathcal{B}}_r, \boldsymbol{\nu}^2 \leftarrow \boldsymbol{\nu}_r^2$

for $i \leq N$ **do**

  $\bar{\mathbf{X}}_i \leftarrow \Gamma_{\boldsymbol{\mathcal{B}}\to\mathbf{I}_d}(\hat{\mathbf{X}}_i)$;

  $\check{\mathbf{X}}_i \leftarrow \mathrm{Exp}_{\mathbf{I}_d}\left(\frac{\mathbf{s}}{\sqrt{\boldsymbol{\nu}^2+\boldsymbol{\epsilon}}}\left(\mathrm{Log}_{\mathbf{I}_d}(\bar{\mathbf{X}}_i)\right)\right)$;

  $\tilde{\mathbf{X}}_i \leftarrow \Gamma_{\mathbf{I}_d\to\hat{\boldsymbol{\mathcal{G}}}}(\check{\mathbf{X}}_i)$.

end

$\acute{\mathbf{X}}_i = (\mathbf{M}^{\frac{1}{2}}\tilde{\mathbf{X}}_i\mathbf{M}^{\frac{1}{2}})^{\frac{1}{\theta}}$

Proof. The proof is presented in App. D.3. □

Props. 3.2 and 3.3 reveal that the power-deformed GBWM is a locally deformed AIM and LEM-like metric ($\theta \to 0$). Therefore, the power-deformed GBWM-based RBN may achieve better results than the GBWM-based one. This is further demonstrated in the following experiments.

3.2.2. RBN under power-deformed GBWM

The computation of RBN under the power-deformed GBWM is similar to that under BWM, as shown below:

$$\text{Centering: } \bar{\mathbf{X}}_i = \widetilde{\mathrm{Exp}}_{\mathbf{I}_d}\left(\widetilde{\Gamma}_{\boldsymbol{\mathcal{B}}\to\mathbf{I}_d}\left(\widetilde{\mathrm{Log}}_{\boldsymbol{\mathcal{B}}}(\mathbf{X}_i)\right)\right), \quad (19)$$

$$\text{Scaling: } \check{\mathbf{X}}_i = \widetilde{\mathrm{Exp}}_{\mathbf{I}_d}\left(\frac{\mathbf{s}}{\sqrt{\boldsymbol{\nu}^2+\boldsymbol{\epsilon}}}\left(\widetilde{\mathrm{Log}}_{\mathbf{I}_d}(\bar{\mathbf{X}}_i)\right)\right), \quad (20)$$

$$\text{Biasing: } \tilde{\mathbf{X}}_i = \widetilde{\mathrm{Exp}}_{\boldsymbol{\mathcal{G}}}\left(\widetilde{\Gamma}_{\mathbf{I}_d\to\boldsymbol{\mathcal{G}}}\left(\widetilde{\mathrm{Log}}_{\mathbf{I}_d}(\check{\mathbf{X}}_i)\right)\right), \quad (21)$$

where $\widetilde{\mathrm{Exp}}, \widetilde{\mathrm{Log}}$ and $\widetilde{\Gamma}$ represent the Riemannian exponential, logarithm, and parallel transportation defined under the power-deformed GBWM, respectively. To streamline the process, we find that the GBWBN can be derived directly from the RBN under the standard BWM. As shown in [20], for any $\mathbf{X} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$ and $\mathbf{M} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, the map $\varphi : (\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}, g^{\mathrm{BW}}) \to (\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}, g^{\mathrm{GBW}})$, denoted as $\varphi(\mathbf{X}) = \mathbf{M}^{\frac{1}{2}}\mathbf{X}\mathbf{M}^{\frac{1}{2}}$, is a Riemannian isometry. Therefore, a diffeomorphism, $f = \boldsymbol{\phi}_\theta \circ \varphi$, can be defined, which is a Riemannian isometry from BWM to $\theta^2 g$ with g as (θ)-GBWM. With the BWM-based RBN defined below:

$$\mathrm{BWBN}(\mathbf{X}_i, \boldsymbol{\mathcal{G}}, \omega, \boldsymbol{\epsilon}, \mathbf{s}), \forall \mathbf{X}_i \in \{\mathbf{X}_{i\cdots N} \in \boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}\}, \quad (22)$$

the following theorem can be derived.

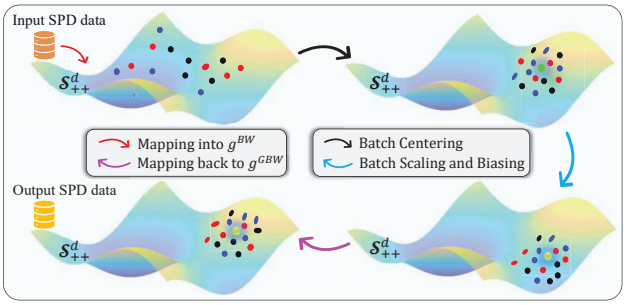


Figure 1. An overview of the computation process for GBWBN.

Theorem 3.4. *Given an SPD manifold $\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}$, a diffeomorphism $f = \boldsymbol{\phi}_\theta \circ \varphi : (\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}, g^{BW}) \to (\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}, g^{\theta\text{-}GBW})$, and a batch of SPD matrices $\{\mathbf{X}_i\}_{i=1}^N$, GBWBN can be achieved via the following steps:*

- ***Mapping data into*** g^{BW}:

$$\hat{\mathbf{X}}_i = f^{-1}(\mathbf{X}_i), \quad \hat{\boldsymbol{\mathcal{G}}} = f^{-1}(\boldsymbol{\mathcal{G}}). \quad (23)$$

- ***Computing BWBN in*** $(\boldsymbol{\mathcal{S}}_{++}^{\boldsymbol{d}}, g^{BW})$:

$$\tilde{\mathbf{X}}_i = BWBN(\hat{\mathbf{X}}_i, \hat{\boldsymbol{\mathcal{G}}}, \omega, \boldsymbol{\epsilon}, \mathbf{s}). \quad (24)$$

- ***Mapping the normalized data back to*** $g^{\theta\text{-}GBW}$:

$$\acute{\mathbf{X}}_i = f(\tilde{\mathbf{X}}_i). \quad (25)$$

Proof. The proof is presented in App. D.4. □

Fig. 1 is an overview of the proposed GBWBN algorithm. As shown in [14], RBN only differs in variance under $\frac{1}{\theta^2}g$ and g. When calculating the variance in the BWBN, we need to divide the variance by θ^2. Now, we give the framework of (θ)-GBWBN in Algorithm 1. Please refer to App. B for detailed information about training the GBWBN module.

4. Experiment

To validate the performance of the proposed GBWBN, we integrate it into two different SPD networks and test it on three distinct signal classification tasks: skeleton-based human action recognition using the HDM05 [39] and NTU RGB+D [50] datasets, and EEG signal classification employing the MAMEM-SSVEP-II dataset [45]. In the experiments, we denote the network structure as $\{d_0, ..., d_r, ..., d_{\mathbb{L}}\}$, where $\mathbb{L}$ represents the number of BiMap layers in each SPD backbone network, and $d_{r-1} \times d_r$ signifies the size of the transformation matrix in the r-th BiMap layer. If the GBWBN layer does not reach saturation under the standard metric ($\theta = 1$), we report the results for the GBWBN under the deformed metric. Detailed descriptions of the data preprocessing and implementation are provided in App. A.

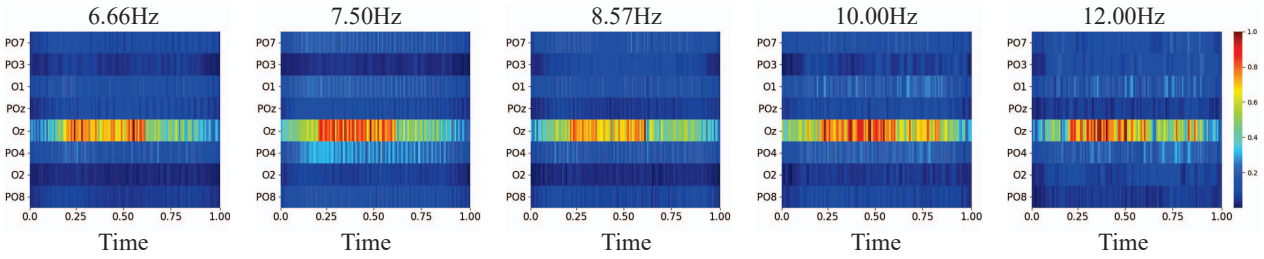


Figure 2. The heatmaps of the absolute gradient responses (computed as in [46]) for the S11 model across five frequency categories on the MAMEM-SSVEP-II dataset. In each heatmap, the x-axis represents time, while the y-axis signifies the EEG channels.

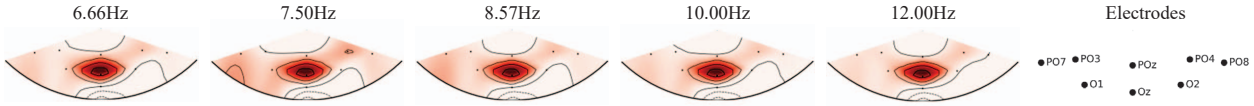


Figure 3. The spatial topomaps of the mean absolute gradient responses (computed as in [46]) across time for the S11 model on the MAMEM-SSVEP-II dataset. The brain region marked in dark red corresponds to channel Oz, which shows strong gradient activation across the visual cortex for all stimulation frequencies.

4.1. Integration with SPDNet

This subsection evaluates GBWBN under the widely adopted SPDNet backbone [27], choosing the MAMEM-SSVEP-II dataset as an example. This dataset [45] has 11 subjects and 5 sessions. Following the protocol of Pan et al. [46], the first three sessions from each subject are used for training, with session 4 for validation and session 5 for testing. Moreover, our SPDNet-GBWBN is evaluated on this dataset under an architecture of {15, 12}, a learning rate of $2.5e^{-3}$, and a batch size of 30.

For the evaluation, in addition to the aforementioned RBN methods, several Euclidean deep learning-based EEG models are also selected for comprehensive comparison, including EEGNet [34], ShallowConvNet [49], SCCNet [58], EEG-TCNet [28], FBCNet [37], TCNet-Fusion [40], and MBEEGSE [1]. The average classification results are reported in Tab. 4. It can be seen that **our GBWBN layer improves the vanilla SPDNet by 3.17%**. Besides, the accuracy of SPDNet-GBWBN is 2.71%, 4.87%, 3.39%, and 5.32% higher than that of SPDNet-BN (m), SPDNet-BN (m+v), SPDNet-Manifoldnorm(AIM), and SPDNet-LieBN(AIM), respectively. These results show the effectiveness of our GBWBN in parsing complex EEG signals.

To reveal the characteristics learned from EEG signals, Fig. 2 visualizes the gradient responses of the S11 model in the channel and temporal domains. The dark red pixels in the heatmaps highlight the robust gradient responses derived from the 8-channel SSVEP EEG data. Besides, the brain topomaps illustrated in Fig. 3 highlight regions in the visual cortex, particularly Oz, with potent gradient activations marked in dark red. Notably, Oz shows the most pronounced gradient responses, especially between 0.25 and 0.60 seconds, underscoring its pivotal role in processing visual information. These findings dovetail with previous studies on the relationship between SSVEP and Oz in EEG research [23, 26]. The position of Oz, right at the heart of the primary visual cortex, is known for its high potential amplitudes and superior signal-to-noise ratio, further supporting these observations. To intuitively demonstrate the proposed power-deformed GBWM-based RBN method is capable of extracting more pivotal geometric features than the AIM-based ones, we also give the visualization results generated by the AIM-based RBN method. The detailed comparison is available at App. E.

Models	Acc. (%)
EEGNet [34]	53.72 ± 7.23
ShallowConvNet [49]	56.93 ± 6.97
SCCNet [58]	62.11 ± 7.70
EEG-TCNet [28]	55.45 ± 7.66
FBCNet [37]	53.09 ± 5.67
TCNet-Fusion [40]	45.00 ± 6.45
MBEEGSE [1]	56.45 ± 7.27
SPDNet [27]	62.30 ± 3.12
SPDNet-BN(m) [7]	62.76 ± 3.01
SPDNet-BN(m+v) [33]	60.60 ± 3.57
SPDNet-Manifoldnorm [8]	62.08 ± 3.56
SPDNet-LieBN [14]	60.15 ± 3.42
SPDNet-GBWBN-($\theta = 1$)	**65.47 ± 3.33**

Table 4. Accuracy comparison on the MAMEM-SSVEP-II dataset.

4.2. Integration with RResNet

We further test the availability of the proposed GBWBN on the skeleton-based action recognition task, choosing the

Models	HDM05		NTU RGB+D	
	Acc. (%)	Time	Acc. (%)	Time
RResNet [31]	61.09 ± 0.60	**3.62**	52.54 ± 0.59	**359.25**
RResNet-BN (m) [7]	63.31 ± 0.61	4.49	53.86 ± 1.19	478.62
RResNet-BN (m+v) [33]	64.51 ± 1.00	6.65	53.94 ± 1.28	546.15
RResNet-Manifoldnorm [8]	63.07 ± 0.80	6.24	53.50 ± 0.46	531.48
RResNet-LieBN [14]	66.43 ± 0.92	5.52	54.74 ± 0.75	523.19
RResNet-GBWBN-($\theta = 1$)	62.01 ± 1.23	8.21	59.45 ± 0.38	563.47
RResNet-GBWBN-($\theta = 0.5$)	$\mathbf{69.10 \pm 0.83}$	8.21	$\mathbf{59.72 \pm 0.31}$	563.47

Table 5. Comparison of different methods on the HDM05 and NTU RGB+D datasets.

λ	SPDNet-BN [7]	SPDNet-GBWBN
$1e^{-7}$	61.28 ± 3.89	$\mathbf{65.17 \pm 3.87}$
$1e^{-6}$	61.40 ± 3.86	$\mathbf{65.32 \pm 3.62}$
$1e^{-5}$	62.76 ± 3.01	$\mathbf{65.47 \pm 3.33}$
$1e^{-4}$	63.56 ± 3.44	$\mathbf{65.44 \pm 3.27}$
$1e^{-3}$	62.42 ± 3.21	$\mathbf{65.65 \pm 3.13}$
$1e^{-2}$	61.89 ± 3.44	$\mathbf{64.56 \pm 3.34}$

Table 6. Accuracy (%) on the MAMEM-SSVEP-II dataset.

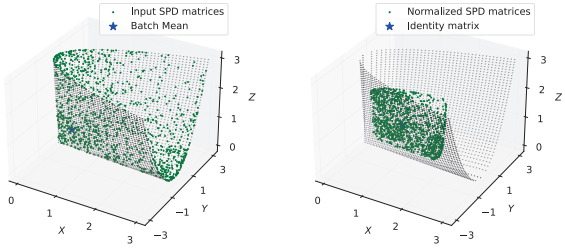


Figure 4. Illustration of ill-conditioned 2×2 SPD matrices (left) and the output (right) of the GBWBN layer. The black dots are semi-positive matrices ($\kappa = \infty$), denoting the SPD boundary, while the interior of the cone is the SPD manifold.

HDM05 and NTU RGB+D datasets as two representative benchmarks. For these experiments, the recently developed Riemannian Residual Network (RResNet) [31] is selected as the SPD backbone. The HDM05 dataset [39] contains 2,337 action sequences across 130 distinct classes. To ensure a fair comparison, we conduct experiments using pre-processed covariance features from [7], yielding a reduced dataset comprising 2,086 sequences over 117 classes. The NTU RGB+D [50] is a large-scale action recognition dataset consisting of 56,880 video clips belonging to 60 different tasks, where we use the flattened 3D joint coordinates as feature vectors, in line with [31]. In the experiments, we evaluate the designed RResNet-GBWBN under the network structures of {93, 30} and {75, 30} for the HDM05 and NTU RGB+D datasets, respectively. We use half of the samples for training and the rest for testing. Besides, the learning rate, number of training epochs, and batch size are set to $2.5e^{-3}$, 200, and 30 on the HDM05 dataset, while those on the NTU RGB+D dataset are configured as $1e^{-2}$, 100, and 256, respectively.

The 10-fold experimental results, along with the average training time (s/epoch) for different methods, are presented in Tab. 5. We can observe that **RResNet-GBWBN with proper power deformation respectively improves the vanilla backbone by 8.01% and 7.18%** on the HDM05 and NTU RGB+D datasets, further demonstrating the efficacy of the proposed GBWBN module (placed post-BiMap layer) in geometric data analysis. It also can be noted that our GBWBN can generally benefit from power deformation, underscoring the significance of metric parameterization. While the proposed GBWBN module incurs slightly more training time than other RBN methods (involving more SVD operations), it consistently delivers superior performance across both datasets.

4.3. Ablations

Statistics of ICSM: As discussed in Sec. 1, given the input covariance matrix $\mathbf{X}$ is not necessarily positive definite, a widely used regularization trick is applied: $\mathbf{X} \leftarrow \mathbf{X} + \lambda \mathbf{I}$. Wherein, λ is a small perturbation and $\mathbf{I}$ is the identity matrix. This means that the smallest eigenvalues of $\mathbf{X}$ is likely to be λ, contributing marginally to alleviate the issue of matrix ill-conditioning. Therefore, we compare the designed SPDNet-GBWBN with SPDNet-BN [7] on the MAMEM-SSVEP-II dataset across various values of λ. As shown in Tab. 6, the accuracy of SPDNet-GBWBN is superior to that of SPDNet-BN in all cases. To further support our claim, we present the condition number ($\kappa = \mathrm{eig}_{\max}/\mathrm{eig}_{\min}$) and the corresponding proportion on the hidden SPD features without, before, and after the GBWBN layer under different λ and training epochs in Tab. 7. **It is evident that ill-conditioning is prevalent across all the datasets, which explains the superiority of our GBWBN over previous RBN methods, as BWM is advantageous in addressing ill-conditioning** [22].

We further investigate the impact of BWM on ICSM to gain a better understanding of the effectiveness of the proposed GBWBN. As shown in Fig. 4, ill-conditioned SPD matrices are concentrated near the boundary of the SPD manifold, restricting the network's expressibility. In contrast, with the designed RBN layer, the matrices are shifted toward the spacious interior of the SPD manifold, thereby enhancing the network's representation power. Besides, as illustrated in Fig. 5, the AIM-based operators could overly stretch eigenvalues, especially under ill-conditioned scenario, leading to computational instability. In contrast, the BWM is relatively more stable, enabling a more effective ICSM learning.

Ablation on the SPD dimension: We conduct experiments on the NTU RGB+D dataset to investigate the impact of SPD dimension on the performance of the GBWBN. As shown in Tab. 8, removing the BiMap layer (which increases

Dataset	λ	Epochs	$\kappa > 10^3$			$\kappa > 10^4$			$\kappa > 10^5$		
			w/o	before	after	w/o	before	after	w/o	before	after
MAMEM-SSVEP-II	$1e^{-7}$	1	399 (79.8%)	108 (21.6%)	0 (0.0%)	195 (39.0%)	5 (1.0%)	0 (0.0%)	74 (14.8%)	0 (0.0%)	0 (0.0%)
		100	428 (85.6%)	17 (3.4%)	0 (0.0%)	247 (49.4%)	0 (0.0%)	0 (0.0%)	115 (23%)	0 (0.0%)	0 (0.0%)
		200 (Final)	458 (91.6%)	13 (2.6%)	0 (0.0%)	274 (100.0%)	0 (0.0%)	0 (0.0%)	113 (54.8%)	0 (0.0%)	0 (0.0%)
	$1e^{-6}$	1	375 (75.0%)	108 (21.6%)	0 (0.0%)	191 (38.2%)	6 (1.2%)	0 (0.0%)	64 (12.8%)	0 (0.0%)	0 (0.0%)
		100	384 (76.8%)	7 (1.4%)	0 (0.0%)	201 (40.2%)	0 (0.0%)	0 (0.0%)	67 (13.4%)	0 (0.0%)	0 (0.0%)
		200 (Final)	370 (74.0%)	40 (8.0%)	0 (0.0%)	202 (40.4%)	0 (0.0%)	0 (0.0%)	71 (14.2%)	0 (0.0%)	0 (0.0%)
	$1e^{-5}$	1	379 (75.8%)	119 (23.8%)	0 (0.0%)	163 (32.6%)	8 (1.6%)	0 (0.0%)	24 (4.8%)	0 (0.0%)	0 (0.0%)
		100	302 (60.4%)	15 (3.0%)	0 (0.0%)	129 (25.8%)	0 (0.0%)	0 (0.0%)	21 (4.2%)	0 (0.0%)	0 (0.0%)
		200 (Final)	113 (22.6%)	32 (6.4%)	0 (0.0%)	11 (2.2%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	$1e^{-4}$	1	194 (38.8%)	134 (26.8%)	0 (0.0%)	99 (19.8%)	12 (2.4%)	0 (0.0%)	4 (0.8%)	0 (0.0%)	0 (0.0%)
		100	196 (39.2%)	30 (6.0%)	0 (0.0%)	43 (8.6%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
		200 (Final)	205 (41.0%)	24 (4.8%)	0 (0.0%)	56 (11.2%)	0 (0.0%)	0 (0.0%)	1 (0.2%)	0 (0.0%)	0 (0.0%)
HDM05	$1e^{-6}$	1	2086 (100.0%)	2086 (100.0%)	0 (0.0%)	2086 (100.0%)	2086 (100.0%)	0 (0.0%)	2086 (100.0%)	2086 (100.0%)	0 (0.0%)
		100	2086 (100.0%)	2086 (100.0%)	0 (0.0%)	2086 (100.0%)	2086 (100.0%)	0 (0.0%)	2086 (100.0%)	2086 (100.0%)	0 (0.0%)
		200 (Final)	2086 (100.0%)	2086 (100.0%)	0 (0.0%)	2086 (100.0%)	2086 (100.0%)	0 (0.0%)	2086 (100.0%)	2086 (100.0%)	0 (0.0%)
NTU RGB+D	$1e^{-6}$	1	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)
		50	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)
		100 (Final)	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)	56,880 (100.0%)	56,880 (100.0%)	0 (0.0%)

Table 7. Statistics (quantity and proportion) on the condition number ($\kappa = \mathrm{eig}_{\max}/\mathrm{eig}_{\min}$) of the SPD features without ('w/o'), before, and after the GBWBN layer on different datasets across various values of λ and training epochs.

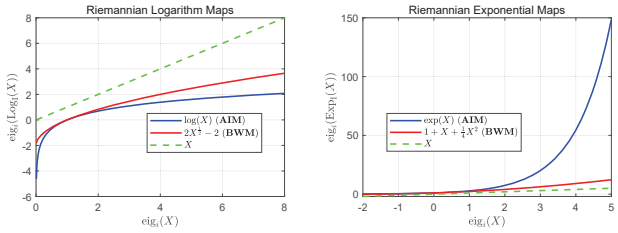


Figure 5. Numerical comparison of the Log and Exp maps under AIM and BWM, where we select the identity matrix as the base point for simplicity. Here, x-axis denotes an eigenvalue and y-axis is the corresponding output eigenvalue.

Methods	Acc. (%)
RResNet-GBWBN	**59.45 ± 0.38**
RResNet-GBWBN-'w/o'BiMap	57.32 ± 0.43
RResNet-BN [7]	53.86 ± 1.19
RResNet-BN-'w/o'BiMap	52.24 ± 1.33

Table 8. Accuracy comparison on the NTU RGB+D dataset.

Datasets	RResNet	RResNet-BWBN	RResNet-GBWBN
HDM05	61.09 ± 0.60	61.94 ± 0.62	**62.01 ± 1.23**
NTU RGB+D	52.54 ± 0.59	57.40 ± 0.52	**59.45 ± 0.38**

Table 9. Accuracy (%) comparison under different RResNets.

the SPD dimension) results in a slight decrease in the learning ability of both RResNet-GBWBN and ResNet-BN. Nevertheless, the accuracy of RResNet-GBWBN-'w/o'BiMap is still higher than that of RResNet-BN-'w/o'BiMap.

Ablation on the GBWBN: As shown in Sec. 3, GBWBN represents a principled generalization of the BWM-based RBN (simplified as BWBN). Therefore, we make experiments on the HDM05 and NTU RGB+D datasets to explore the effectiveness of the generalized BWM in RBN design.

θ	0.25	0.5	0.75	1
HDM05	65.28 ± 1.03	**69.10 ± 0.83**	62.89 ± 0.74	62.01 ± 1.23
NTU RGB+D	58.23 ± 0.28	59.72 ± 0.31	**59.82 ± 0.27**	59.45 ± 0.38

Table 10. Accuracy (%) comparison under various values of θ.

The results listed in Tab. 9 show that both GBWBN and BWBN can intensify the classification ability of the vanilla SPD backbone. Notably, the performance of BWBN is inferior to that of GBWBN. This further supports the notion that the learnable extension applied to BWBN allows our RBN algorithm to flexibly capture batch-specific metric geometry, thereby improving its representational capacity.

Ablation on the power deformation: In this paper, we propose the power-deformed GBWM, which interpolates between GBWM ($\theta = 1$) and LEM-like metric ($\theta \to 0$). Thereby, we study the impact of such deformation on the performance of RBN under different θ, selecting the HDM05 and NTU RGB+D datasets as two examples. The results given in Tab. 10 show that the nonlinear activation of the metric geometry can enhance the model accuracy.

5. Conclusion

We introduce a novel RBN algorithm grounded in the GBW geometry in this article. Compared to other metrics, the GBWM offers distinct advantages in computing ICSM, which we believe provides an edge for the proposed GBWBN over existing methods. In addition, the power deformation is employed to further refine and enhance the latent GBWM geometry. Extensive experiments and ablation studies validate the superiority of our approach over previous RBN methods.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62306127, 62020106012, 62332008), the Natural Science Foundation of Jiangsu Province (BK20231040), the Fundamental Research Funds for the Central Universities (JUSRP124015), and the National Key R&D Program of China (2023YFF1105102, 2023YFF1105105).

References

[1] Ghadir Ali Altuwaijri, Ghulam Muhammad, Hamdi Altaheri, and Mansour Alsulaiman. A multi-branch convolutional neural network with squeeze-and-excitation attention blocks for EEG-based motor imagery signals classification. *Diagnostics*, 2022. 6

[2] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.*, 2007. 1

[3] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing*, 2013. 1

[4] Frédéric Barbaresco. Jean-louis koszul and the elementary structures of information geometry. *Geometric Structures of Information*, 2019. 4

[5] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the bures–wasserstein distance between positive definite matrices. *Expo. Math.*, 2019. 2, 3

[6] Silvere Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Trans. Autom. Control*, 2013. 1

[7] Daniel Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for SPD neural networks. In *NeurIPS*, 2019. 1, 2, 3, 4, 6, 7, 8

[8] Rudrasis Chakraborty. Manifoldnorm: Extending normalizations on Riemannian manifolds. *arXiv preprint arXiv:2003.13869*, 2020. 1, 2, 6, 7

[9] Rudrasis Chakraborty, Chun-Hao Yang, Xingjian Zhen, Monami Banerjee, Derek Archer, David Vaillancourt, Vikas Singh, and Baba Vemuri. A statistical recurrent model on the manifold of symmetric positive definite matrices. In *NeurIPS*, 2018. 1

[10] Rudrasis Chakraborty, Jose Bouza, Jonathan Manton, and Baba C Vemuri. Manifoldnet: A deep neural network for manifold-valued data with applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[11] Ziheng Chen, Tianyang Xu, Xiao-Jun Wu, Rui Wang, and Josef Kittler. Hybrid Riemannian graph-embedding metric learning for image set classification. *IEEE Trans. Big Data*, 2021. 1

[12] Ziheng Chen, Tianyang Xu, Xiao-Jun Wu, Rui Wang, Zhiwu Huang, and Josef Kittler. Riemannian local mechanism for SPD neural networks. In *AAAI*, 2023. 1

[13] Ziheng Chen, Yue Song, Gaowen Liu, Ramana Rao Kompella, Xiao-Jun Wu, and Nicu Sebe. Riemannian multiclass logistics regression for SPD neural networks. In *CVPR*, 2024. 1

[14] Ziheng Chen, Yue Song, Yunmei Liu, and Nicu Sebe. A Lie group approach to Riemannian normalization for SPD neural networks. In *ICLR*, 2024. 1, 2, 3, 4, 5, 6, 7

[15] Ziheng Chen, Yue Song, Rui Wang, Xiaojun Wu, and Nicu Sebe. RMLR: Extending multinomial logistic regression into general geometries. In *NeurIPS*, 2024. 1

[16] Ziheng Chen, Yue Song, Tianyang Xu, Zhiwu Huang, Xiao-Jun Wu, and Nicu Sebe. Adaptive Log-Euclidean metrics for SPD matrix learning. *IEEE Trans. Image Process.*, 2024. 1

[17] Ziheng Chen, Yue Song, Xiaojun Wu, Gaowen Liu, and Nicu Sebe. Understanding matrix function normalizations in covariance pooling through the lens of Riemannian geometry. In *ICLR*, 2025. 1

[18] Ziheng Chen, Yue Song, Xiaojun Wu, and Nicu Sebe. Gyrogroup batch normalization. In *ICLR*, 2025. 1

[19] Mengyu Dai, Zhengwu Zhang, and Anuj Srivastava. Analyzing dynamical brain functional connectivity as trajectories on space of covariance matrices. *IEEE Trans. Med. Imaging*, 2019. 1

[20] Andi Han, Bamdev Mishra, Pratik Jawanpuria, and Junbin Gao. Generalized bures-wasserstein geometry for positive definite matrices. *arXiv preprint arXiv:2110.10464*, 2021. 5

[21] Andi Han, Bamdev Mishra, Pratik Kumar Jawanpuria, and Junbin Gao. On Riemannian optimization over positive definite matrices with the bures-wasserstein geometry. In *NeurIPS*, 2021. 2, 3

[22] Andi Han, Bamdev Mishra, Pratik Jawanpuria, and Junbin Gao. Learning with symmetric positive definite matrices via generalized bures-wasserstein geometry. In *International Conference on Geometric Science of Information*, 2023. 2, 3, 7

[23] Chengcheng Han, Guanghua Xu, Jun Xie, Chaoyang Chen, and Sicong Zhang. Highly interactive brain–computer interface based on flicker-free steady-state motion visual evoked potential. *Sci Rep*, 2018. 6, 3

[24] Mehrtash Harandi, Mathieu Salzmann, and Richard Hartley. Joint dimensionality reduction and metric learning: A geometric take. In *ICML*, 2017. 1

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[26] Christoph S Herrmann. Human EEG responses to 1–100 hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Exp. Brain Res.*, 2001. 6, 3

[27] Zhiwu Huang and Luc Van Gool. A Riemannian network for spd matrix learning. In *AAAI*, 2017. 1, 6

[28] Thorir Mar Ingolfsson, Michael Hersche, Xiaying Wang, Nobuaki Kobayashi, Lukas Cavigelli, and Luca Benini. EEG-TCNet: An accurate temporal convolutional network for embedded motor-imagery brain–machine interfaces. *IEEE International Conference on Systems, Man, and Cybernetics*, 2020. 6

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 1, 4

[30] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Training deep networks with structured layers by matrix backpropagation. *arXiv preprint arXiv:1509.07838*, 2015. 1

[31] Isay Katsman, Eric Chen, Sidhanth Holalkere, Anna Asch, Aaron Lou, Ser Nam Lim, and Christopher M De Sa. Riemannian residual neural networks. In *NeurIPS*, 2024. 1, 7

[32] Reinmar Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe. SPD domain-specific batch normalization to crack interpretable unsupervised domain adaptation in EEG. In *NeurIPS*, 2022. 1, 2

[33] Reinmar J Kobler, Jun-ichiro Hirayama, and Motoaki Kawanabe. Controlling the fréchet variance improves batch normalization on the symmetric positive definite manifold. In *ICASSP*, 2022. 6, 7

[34] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *J. Neural Eng.*, 2018. 6, 1

[35] Zhenhua Lin. Riemannian geometry of symmetric positive definite matrices via cholesky decomposition. *SIAM J. Matrix Anal. Appl.*, 2019. 1

[36] Luigi Malagò, Luigi Montrucchio, and Giovanni Pistone. Wasserstein Riemannian geometry of gaussian densities. *Information Geometry*, 2018. 2

[37] Ravikiran Mane, Effie Chew, Karen Chua, Kai Keng Ang, Neethu Robinson, A Prasad Vinod, Seong-Whan Lee, and Cuntai Guan. FBCNet: A multi-view convolutional neural network for brain-computer interface. *arXiv preprint arXiv:2104.01233*, 2021. 6

[38] Hà Quang Minh. Alpha procrustes metrics between positive definite operators: a unifying formulation for the bures-wasserstein and log-euclidean/log-hilbert-schmidt metrics. *Linear Alg. Appl.*, 2022. 2

[39] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Mocap database hdm05. *Institut für Informatik II, Universität Bonn*, 2007. 5, 7, 1

[40] Yazeed K Musallam, Nasser I AlFassam, Ghulam Muhammad, Syed Umar Amin, Mansour Alsulaiman, Wadood Abdul, Hamdi Altaheri, Mohamed A Bencherif, and Mohammed Algabri. Electroencephalography-based motor imagery classification using temporal convolutional network fusion. *Biomed. Signal Process. Control*, 2021. 6

[41] Xuan Son Nguyen. Geomnet: A neural network based on Riemannian geometries of SPD matrix space and Cholesky space for 3d skeleton-based interaction recognition. In *ICCV*, 2021. 1

[42] Xuan Son Nguyen. The gyro-structure of some matrix manifolds. In *NeurIPS*, 2022. 1

[43] Xuan Son Nguyen and Shuo Yang. Building neural networks on matrix manifolds: A gyrovector space approach. In *ICML*, 2023. 1

[44] Xuan Son Nguyen, Shuo Yang, and Aymeric Histace. Matrix manifold neural networks++. In *ICLR*, 2024. 1

[45] Spiros Nikolopoulos. Mamem EEG ssvep dataset ii (256 channels, 11 subjects, 5 frequencies presented simultaneously). 2021. 5, 6, 1

[46] Yue-Ting Pan, Jing-Lun Chou, and Chun-Shu Wei. MAtt: A manifold attention network for EEG decoding. In *NeurIPS*, 2022. 1, 6

[47] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian framework for tensor computing. *Int. J. Comput. Vis.*, 2006. 1

[48] Salem Said, Lionel Bombrun, Yannick Berthoumieu, and Jonathan H Manton. Riemannian gaussian distributions on the space of symmetric positive definite matrices. *IEEE Trans. Inf. Theory*, 2017. 4

[49] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum. Brain Mapp.*, 2017. 6

[50] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, 2016. 5, 7, 1

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[52] Yann Thanwerdas and Xavier Pennec. Is affine-invariance well defined on SPD matrices? a principled continuum of metrics. In *International Conference on Geometric Science of Information*, 2019. 4

[53] Yann Thanwerdas and Xavier Pennec. The geometry of mixed-Euclidean metrics on symmetric positive definite matrices. *Differ. Geom. Appl.*, 2022. 2, 4

[54] Yann Thanwerdas and Xavier Pennec. O (n)-invariant Riemannian metrics on SPD matrices. *Linear Alg. Appl.*, 2023. 3

[55] Rui Wang, Xiao-Jun Wu, and Josef Kittler. Symnet: A simple symmetric positive definite manifold deep learning method for image set classification. *IEEE Trans. Neural Netw. Learn. Syst.*, 2022. 1

[56] Rui Wang, Xiao-Jun Wu, Ziheng Chen, Cong Hu, and Josef Kittler. SPD manifold deep metric learning for image set classification. *IEEE Trans. Neural Netw. Learn. Syst.*, 2024. 1

[57] Rui Wang, Xiao-Jun Wu, Tianyang Xu, Cong Hu, and Josef Kittler. Deep metric learning on the SPD manifold for image set classification. *IEEE Trans. Circuits Syst. Video Technol.*, 2024. 1

[58] Chun-Shu Wei, Toshiaki Koike-Akino, and Ye Wang. Spatial component-wise convolutional network (SCCNet) for motor-imagery EEG classification. *IEEE/EMBS Conference on Neural Engineering*, 2019. 6, 1

[59] Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *KDD*, 2024. 1