Riemannian Multinomial Logistics Regression for SPD Neural Networks

Ziheng Chen[1], Yue Song[1]*, Gaowen Liu[2], Ramana Rao Kompella[2], Xiao-Jun Wu[3] & Nicu Sebe[1]
[1] University of Trento, [2] Cisco Systems, [3] Jiangnan University
ziheng_ch@163.com, yue.song@unitn.it

Abstract

Deep neural networks for learning Symmetric Positive Definite (SPD) matrices are gaining increasing attention in machine learning. Despite the significant progress, most existing SPD networks use traditional Euclidean classifiers on an approximated space rather than intrinsic classifiers that accurately capture the geometry of SPD manifolds. Inspired by Hyperbolic Neural Networks (HNNs), we propose Riemannian Multinomial Logistics Regression (RMLR) for the classification layers in SPD networks. We introduce a unified framework for building Riemannian classifiers under the metrics pulled back from the Euclidean space, and showcase our framework under the parameterized Log-Euclidean Metric (LEM) and Log-Cholesky Metric (LCM). Besides, our framework offers a novel intrinsic explanation for the most popular LogEig classifier in existing SPD networks. The effectiveness of our method is demonstrated in three applications: radar recognition, human action recognition, and electroencephalography (EEG) classification. The code is available at https://github.com/GitZH-Chen/SPDMLR.git.

1. Introduction

Symmetric Positive Definite (SPD) matrices are commonly encountered in a diverse range of scientific fields, such as medical imaging [8, 9], signal processing [1, 5, 6, 28], elasticity [22, 37], question answering [36, 40], graph classification [10], and computer vision [7, 9, 11, 12, 24, 29, 39, 41, 45, 47, 60, 61]. Despite their ubiquitous presence, traditional learning algorithms are ineffective in handling the non-Euclidean geometry of SPD matrices. To address this limitation, several Riemannian metrics have been proposed, including Affine-Invariant Metric (AIM) [44], Log-Euclidean Metric (LEM) [2], and Log-Cholesky Metric (LCM) [35]. With these Riemannian metrics, various machine learning techniques can be generalized into SPD manifolds.

*Corresponding author.

Inspired by the great success of deep learning [25, 27, 32], several deep networks have been developed on SPD manifolds. Despite their promising performance, many approaches still rely on Euclidean spaces for classification, such as tangent spaces [5, 14, 29, 31, 39–41, 55, 56], ambient Euclidean spaces [45, 46, 54], and coordinate systems [8]. However, these strategies distort the intrinsic geometry of the SPD manifold, undermining the effectiveness of SPD neural networks. Notably, there are also some similarity-based classifiers originally designed for shallow learning methods [12, 20, 24]. Although these classifiers can be extended to deep SPD neural networks [57, 58], the calculation of pair-wise distance might undermine the training efficiency. Recently, motivated by HNNs [18], three kinds of SPD Multinomial Logistics Regression (MLR) based on the gyro-structures induced by LEM, LCM and AIM are developed in [42]. However, the proposed SPD MLRs rely on the gyro-structures, limiting their generality. Besides, in [9], the authors also introduce an invariant layer for manifold-valued data mimicking the invariant FC layer in CNNs. However, it is designed for gridded manifold-valued data, which is not the primary data type encountered in many other SPD networks. Following the convention of most SPD networks, we only focus on non-gridded cases.

In fact, SPD MLR can be directly derived under LEM and LCM without the assistance of gyro structures. More generally, LEM and LCM belong to Pullback Euclidean Metrics (PEMs), which are metrics pulled back from the Euclidean space. This paper focuses on PEMs and proposes a unified framework for building SPD Multinomial Logistics Regression (SPD MLR) under PEMs. On the empirical side, we focus on the parameterized Log-Euclidean Metric (LEM) and Log-Cholesky Metric (LCM) [15], which generalize the standard LEM and LCM by the pullback of matrix power. We showcase our SPD MLRs under these parameterized metrics. Besides, our framework encompasses the gyro SPD MLRs induced by the standard LEM and LCM in [42]. More importantly, our framework also provides an intrinsic explanation for the commonly used LogEig classifier on SPD manifolds, which consists of successive matrix logarithm, FC, and softmax layers. Finally, extensive ex-

periments demonstrate that our proposed Riemannian classifiers exhibit consistent performance gains across widely used SPD benchmarks. The main **contributions** are summarized as follows:

(a) We introduce a general framework for building SPD MLRs under PEMs and design specific SPD MLRs under two parameterized metric families.

(b) Our framework offers an intrinsic explanation of the most popular LogEig classifier which stacks matrix logarithm, the FC layer, and softmax.

(c) Extensive experiments on widely used SPD learning benchmarks demonstrate the superiority of our proposed classifiers over the previous baselines.

Main theoretical results: Defs. 3.1 and 3.2 introduce the definitions of the SPD hyperplane and SPD MLR, respectively. The core idea lies in the computation of marginal distance to the hyperplane defined in Eq. (14). As Lem. 3.5 demonstrates, this problem admits a closed-form solution under any PEM. Consequently, we establish a uniform expression of SPD MLR under any PEM in Thm. 3.8. As the parameterized LEM and LCM all belong to PEMs, the associated SPD MLRs can be obtained by Thm. 3.8, the expressions of which are presented in Cor. 4.1. Finally, our framework also offers an intrinsic explanation for the widely used LogEig classifier in Prop. 5.1. Due to page limits, all the proofs are placed in App. C.

2. Preliminaries

This section briefly reviews some basic concepts in Riemannian geometry and SPD manifolds. Please refer to [16, 52] for in-depth discussions.

2.1. Riemannian geometry

We first recap the concept of the pullback metric, which is ubiquitous in differential manifolds.

Definition 2.1 (Pullback Metrics)**.** Suppose $\mathcal{M}, \mathcal{N}$ are smooth manifolds, g is a Riemannian metric on $\mathcal{N}$, and $f : \mathcal{M} \to \mathcal{N}$ is a diffeomorphism. Then f can induce a Riemannian metric on $\mathcal{M}$ defined as

$$(f^* g)_p(V_1, V_2) = g_{f(p)}(f_{*,p}(V_1), f_{*,p}(V_2)), \qquad (1)$$

where $p \in \mathcal{M}$, $f_{*,p}(\cdot)$ is the differential map of f at p, $V_i \in T_p\mathcal{M}$, and f^*g is the pullback metric by f from $\mathcal{N}$.

The exponential & logarithmic maps and parallel transportation are also crucial for Riemannian approaches in machine learning. To bypass the notation burdens caused by their definitions, we review the geometric reinterpretation of these operators [16, 44]. In detail, in a manifold $\mathcal{M}$, geodesics correspond to straight lines in Euclidean space. A tangent vector $\overrightarrow{xy} \in T_x\mathcal{M}$ can be locally identified to a point y on the manifold by geodesic starting at x with an initial velocity of $\overrightarrow{xy}$, *i.e.* $y = \mathrm{Exp}_x(\overrightarrow{xy})$. On the other hand, the logarithmic map is the inverse of the exponential map, generating the initial velocity of the geodesic connecting x and y, *i.e.* $\overrightarrow{xy} = \mathrm{Log}_x(y)$. These two operators generalize the idea of addition and subtraction in the Euclidean space. For the parallel transportation $\Gamma_{x\to y}(V)$, it is a generalization of parallelly moving a vector along a curve in Euclidean space. we summarize the reinterpretation in Tab. 1.

Operations	Euclidean spaces	Riemannian manifolds
Straight line	Straight line	Geodesic
Subtraction	$\overrightarrow{xy} = y - x$	$\overrightarrow{xy} = \log_x(y)$
Addition	$y = x + \overrightarrow{xy}$	$y = \exp_x(\overrightarrow{xy})$
Parallelly moving	$V \to V$	$\Gamma_{x\to y}(V)$

Table 1. Reinterpretation of Riemannian operators.

2.2. The geometry of SPD manifolds

Now, we introduce some necessary preliminaries about SPD manifolds. The set of SPD matrices, denoted as $\mathcal{S}^n_{++}$, forms a smooth manifold known as the SPD manifold [2]. Several successful Riemannian metrics have been established on SPD manifolds, such as LEM [2], AIM [44] and LCM [35]. Recently, LEM and AIM are generalized into two-parameter families of metrics [51] , namely (α, β)-AIM and (α, β)-LEM by the $\mathrm{O}(n)$-invariant inner product on the Euclidean space $\mathcal{S}^n$ of symmetric matrices:

$$\langle V, W\rangle^{(\alpha,\beta)} = \alpha\langle V, W\rangle + \beta\,\mathrm{tr}(V)\,\mathrm{tr}(W), \qquad (2)$$

where $(\alpha, \beta) \in \mathbf{ST} = \{(\alpha, \beta) \in \mathbb{R}^2 \mid \min(\alpha, \alpha + n\beta) > 0\}$, and $V, W \in \mathcal{S}^n$.

In this study, we focus on (α, β)-LEM and LCM. We first make some notations and then summarize all the necessary Riemannian operators in Tab. 2. Given SPD matrices $P, Q \in \mathcal{S}^n_{++}$ along with tangent vectors $V, W \in T_P\mathcal{S}^n_{++}$, we introduce the following notations. Specifically, the Riemannian metric at P is represented as $g_P(\cdot, \cdot)$, while $\mathrm{Log}_P(\cdot)$ denotes the Riemannian logarithm at P. $\Gamma_{P\to Q}$ signifies the parallel transport along the geodesic connecting P and Q. The matrix exponential and logarithmic functions are denoted as $\mathrm{mexp}(\cdot)$ and $\mathrm{mlog}(\cdot)$, respectively. In addition, $\mathrm{Chol}(\cdot)$ denotes the Cholesky decomposition, with $L = \mathrm{Chol}\, P$ and $K = \mathrm{Chol}\, Q$ representing the Cholesky factors of P and Q. The differentials of mlog and Chol^{-1} at P and L are respectively denoted as $\mathrm{mlog}_{*,P}$ and $(\mathrm{Chol})^{-1}_{*,L}$. $\lfloor\cdot\rfloor$ refers to the strictly lower part of a square matrix, and $\mathrm{Dlog}(L)$ denotes a diagonal matrix comprised of the logarithm of the diagonal elements of L.

Following the terminology in [13], we define the pullback metrics from Euclidean spaces by diffeomorphisms as the Pullback Euclidean Metrics (PEMs). Chen *et al.* [13]

Name	$g_P(V,W)$	$\mathrm{Log}_P Q$	$\Gamma_{P\to Q}(V)$
(α,β)-LEM	$\langle \mathrm{mlog}_{*,P}(V), \mathrm{mlog}_{*,P}(W)\rangle^{(\alpha,\beta)}$	$(\mathrm{mlog}_{*,P})^{-1}\left[\mathrm{mlog}(Q)-\mathrm{mlog}(P)\right]$	$(\mathrm{mlog}_{*,Q})^{-1}\circ \mathrm{mlog}_{*,P}(V)$
LCM	$\sum_{i>j}\tilde{V}_{ij}\tilde{W}_{ij}+\sum_{j=1}^{n}\tilde{V}_{jj}\tilde{W}_{jj}L_{jj}^{-2}$	$(\mathrm{Chol}^{-1})_{*,L}\left[\lfloor K\rfloor-\lfloor L\rfloor+\mathbb{D}(L)\,\mathrm{Dlog}(\mathbb{D}(L)^{-1}\mathbb{D}(K))\right]$	$(\mathrm{Chol}^{-1})_{*,K}\left[\lfloor\tilde{V}\rfloor+\mathbb{D}(K)\mathbb{D}(L)^{-1}\mathbb{D}(\tilde{V})\right]$

Table 2. Riemannian operators of (α,β)-LEM and LCM on SPD manifolds.

demonstrate that both LEM and LCM are PEMs. We recall an excerpt from Theorem 4.2 of [13], covering the properties of PEMs on SPD manifolds.

Theorem 2.2 (Pullback Euclidean Metrics (PEMs))**.** *Let $S,S_1,S_2\in\mathcal{S}_{++}^n$ and $V_1,V_2\in T_S\mathcal{S}_{++}^n$, $\phi:\mathcal{S}_{++}^n\to\mathcal{S}^n$ is a diffeomorphism. We define the following operations,*

$$S_1\odot_\phi S_2=\phi^{-1}(\phi(S_1)+\phi(S_2)), \tag{3}$$

$$g_S^\phi(V_1,V_2)=\langle\phi_{*,S}(V_1),\phi_{*,S}(V_2)\rangle, \tag{4}$$

*where $\phi_{*,S}:T_S\mathcal{S}_{++}^n\to T_{\phi(S)}\mathcal{S}^n$ is the differential map of ϕ at S, and $\langle\cdot,\cdot\rangle$ is the standard Frobenius inner product. Then, we have the following conclusions: $\{\mathcal{S}_{++}^n,\odot_\phi\}$ is an Abelian Lie group, $\{\mathcal{S}_{++}^n,g^\phi\}$ is a Riemannian manifold, and g^ϕ is a bi-invariant metric, called Pullback Euclidean Metric (**PEM**). The associated geodesic distance is*

$$d^\phi(S_1,S_2)=\|\phi(S_1)-\phi(S_2)\|_{\mathrm{F}}, \tag{5}$$

where $\|\cdot\|_{\mathrm{F}}$ is the norm induced by $\langle\cdot,\cdot\rangle$. The Riemannian operators are as follows

$$\mathrm{Exp}_{S_1}V=\phi^{-1}(\phi(S_1)+\phi_{*,S_1}V), \tag{6}$$

$$\mathrm{Log}_{S_1}S_2=\phi^{-1}_{*,\phi(S_1)}(\phi(S_2)-\phi(S_1)), \tag{7}$$

$$\Gamma_{S_1\to S_2}(V)=\phi^{-1}_{*,\phi(S_2)}\circ\phi_{*,S_1}(V), \tag{8}$$

where $V\in T_{S_1}\mathcal{S}_{++}^n$ is a tangent vector, Exp_{S_1} is the Riemannian exponential at S_1, and ϕ_*^{-1} are the differential maps ϕ^{-1}.

3. SPD MLRs on SPD manifolds

This section first reformulates the Euclidean MLR. Then, we deal with SPD MLR under arbitrary PEM on SPD manifolds.

3.1. Reformulation of Euclidean MLR

The Euclidean MLR was first reformulated in [33] from the perspective of distances to margin hyperplanes. Hyperbolic MLR was designed based on this reformulation [18]. In [42], the authors further proposed three gyro SPD MLRs based on the gyro-structures induced by AIM, LEM, and LCM. We now briefly review the reformulation of Euclidean MLR.

Given C classes, MLR in $\mathbb{R}^n$ computes the following softmax probabilities:

$$\forall k\in\{1,\dots,C\},p(y=k\mid x)\propto\exp\left(\left(\langle a_k,x\rangle-b_k\right)\right), \tag{9}$$

where $b_k\in\mathbb{R}$, and $x,a_k\in\mathbb{R}^n$. As shown in [33, Sec. 5] and [18, Sec. 3.1], Eq. (9) can be reformulated as

$$\begin{aligned}&p(y=k\mid x)\propto\\&\exp(\mathrm{sign}(\langle a_k,x-p_k\rangle)\|a_k\|d(x,H_{a_k,p_k})),\end{aligned} \tag{10}$$

where $\langle a_k,p_k\rangle=b_k$, and H_{a_k,p_k} is referred to a hyperplane, defined as

$$H_{a_k,p_k}=\{x\in\mathbb{R}^n:\langle a_k,x-p_k\rangle=0\}, \tag{11}$$

Recalling Tab. 1, $\mathrm{Log}_p x$ is the natural generalization of the directional vector $\vec{px}=x-p$ starting at p and ending at x, while the Riemannian metric at p corresponds to the inner product. Therefore, the MLR in Eq. (10) and hyperplane in Eq. (11) can be readily generalized into the SPD manifold $\{\mathcal{S}_{++}^n,g\}$.

Definition 3.1 (SPD hyperplanes)**.** Given $P\in\mathcal{S}_{++}^n,A\in T_P\mathcal{S}_{++}^n\backslash\{\mathbf{0}\}$, we define the SPD hyperplane as

$$\begin{aligned}\tilde{H}_{A,P}=\{S\in\mathcal{S}_{++}^n:\,&g_P(\mathrm{Log}_P S,A)\\=\,&\langle\mathrm{Log}_P S,A\rangle_P=0\},\end{aligned} \tag{12}$$

where P and A are referred to as shift and normal matrices, respectively.

Definition 3.2 (SPD MLR)**.** SPD MLR is defined as

$$\begin{aligned}&p(y=k\mid S)\\&\propto\exp(\mathrm{sign}(\langle A_k,\mathrm{Log}_{P_k}(S)\rangle_{P_k})\|A_k\|_{P_k}d(S,\tilde{H}_{A_k,P_k})),\end{aligned} \tag{13}$$

where $P_k\in\mathcal{S}_{++}^n$, $A_k\in T_{P_k}\mathcal{S}_{++}^n\backslash\{\mathbf{0}\}$, $\langle\cdot,\cdot\rangle_{P_k}=g_{P_k}$, and $\|\cdot\|_{P_k}$ is the norm on $T_{P_k}\mathcal{S}_{++}^n$ induced by g at P_k, and $\tilde{H}_{A_k,P_k}$ is a margin hyperplane in $\mathcal{S}_{++}^n$ as defined in Eq. (12). $d(S,\tilde{H}_{A_k,P_k})$ denotes the margin distance between S and SPD hyperplane $\tilde{H}_{A_k,P_k}$, which is formulated as:

$$d(S,\tilde{H}_{A_k,P_k}))=\inf_{Q\in\tilde{H}_{A_k,P_k}}d(S,Q), \tag{14}$$

where $d(S,Q)$ is the geodesic distance induced by g.

In geometry, the hyperplane in Eq. (11) is actually a regular submanifold of the trivial manifold $\mathbb{R}^n$. As for our definition of SPD hyperplanes, we have a similar result.

Proposition 3.3 (Submanifolds)**.** *The SPD hyperplane (as defined in Eq. (12)) under any geometrically complete Riemannian metric g is a regular submanifold of SPD manifolds.*

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

Prop. 3.3 rationalizes our Def. 3.1, as both the SPD hyperplane and Euclidean hyperplane are submanifolds. Nevertheless, we still follow the nomenclature of [18, 33] and call $\tilde{H}_{A,P}$ SPD hyperplane.

Remark 3.4 (Difference with the gyro SPD MLR). Although the gyro SPD MLR introduced in [42] and our method both extend the Euclidean MLR into SPD manifolds, there exist two main differences:

1. **The mathematical techniques employed are different.** [42] adopted gyro structures to reformulate Eqs. (10) and (11). However, their gyro structures are induced by the Riemannian metrics. Also, the gyro inner product and gyro norm [42, Def. 2.15] are defined by the inner product and norm in the tangent space at the identity matrix, *i.e.* $T_I\mathcal{S}^n_{++}$. In contrast, our approach directly applies Riemannian geometry to reformulate Euclidean MLR.
2. **The margin distance in Eq. (14) are calculated differently.** The margin distance in gyro SPD MLR shares the same expression as our Eq. (14), except that the distance in the right-hand side is gyro distance, which is defined by the distance on $T_I\mathcal{S}^n_{++}$. To bypass the optimization problem in Eq. (14), Xuan Son Nguyen and Shuo Yang [42] introduced the *pseudo-gyrodistance*. In contrast, we directly use the geodesic distance, which is the most natural descriptor for characterizing the distance on manifolds.

3.2. SPD MLRs under PEMs

Recalling that for our SPD MLR in Def. 3.2, under most Riemannian metrics on SPD manifolds, all the involved operators in Eq. (13) have close form expressions, except the margin distance in Eq. (14). Therefore, the only difficulty lies in the calculation of the margin distance. This subsection follows the notations in Thm. 2.2 and proposes a general expression for SPD MLRs under PEMs.

We chose PEMs as our starting metrics mainly because of its extensive inclusion and easy computation. Several Riemannian metrics, including LEM, LCM, and their variants [50, 51], all belong to PEMs. Besides, due to the fast and simple calculation of PEMs, the margin distance under a PEM has a closed-form expression, while other metrics like AIM would be complicated to obtain the distances to hyperplanes.

We start by calculating the margin distance in Eq. (14) under a given PEM.

Lemma 3.5. *Given a PEM g, the margin distance defined in Eq. (14) has a closed-form solution:*

$$d(S, \tilde{H}_{A_k,P_k})) = d(\phi(S), H_{\phi_{*,P_k}(A_k),\phi(P_k)}), \tag{15}$$

$$= \frac{|\langle \phi(S) - \phi(P_k), \phi_{*,P_k}(A_k)\rangle|}{\|A_k\|_{P_k}}, \tag{16}$$

where $|\cdot|$ is the absolute value.

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

Putting Eq. (16) into Eq. (13), we obtain our SPD MLR under a given PEM:

$$p(y = k \mid S) \propto \exp(\langle A_k, \mathrm{Log}_{P_k}(S)\rangle_{P_k}), \tag{17}$$

$$= \exp(\langle \phi(S) - \phi(P_k), \phi_{*,P_k}(A_k)\rangle), \tag{18}$$

where $S, P_k \in \mathcal{S}^n_{++}$ and $A_k \in T_{P_k}\mathcal{S}^n_{++}\backslash\{\mathbf{0}\}$. When P_k is fixed, $A_k \in T_{P_k}\mathcal{S}^n_{++}$ indeed lies in a Euclidean space. However, P_k would vary during training, making A_k non-Euclidean. To remedy this issue, we propose two solutions. The first one is the parallel transportation from a fixed tangent space, writing $A_k = \Gamma_{Q\to P_k}(\tilde{A}_k)$ with $\tilde{A}_k \in T_Q\mathcal{S}^n_{++}$ as a Euclidean parameter. This is the solution also adopted by HNNs [18], where the tangent point is the zero vector. Alternatively, one can also rely on the differential of a Lie group translation, which is widely used in differential manifolds [52, § 20]. Since the Lie groups associated with PEMs are abelian, we only consider the left translation. We have the following two lemmas to show the relation between the parallel transport and the differential of left translation.

Lemma 3.6. *Given a PEM, any parallel transportation is equivalent to the differential map of a left translation and vice versa.*

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

Lemma 3.7. *Given two fixed SPD matrices $Q_1, Q_2 \in \mathcal{S}^n_{++}$, we have the following equivalence for parallel transportations under a PEM,*

$$\begin{aligned} &\forall \tilde{A}_{1,k} \in T_{Q_1}\mathcal{S}^n_{++}, \exists! \tilde{A}_{2,k} \in T_{Q_2}\mathcal{S}^n_{++}, \\ &s.t. \Gamma_{Q_1\to P_k}(\tilde{A}_{1,k}) = \Gamma_{Q_2\to P_k}(\tilde{A}_{2,k}). \end{aligned} \tag{19}$$

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

Lem. 3.6 indicates that under PEMs, the above two solutions are equivalent, while Lem. 3.7 implies that anchor points can be arbitrarily chosen. Therefore, without loss of generality, we generate A_k from the tangent space at the identity matrix I by parallel transportation, *i.e.* $A_k = \Gamma_{I\to P_k}(\tilde{A}_k)$ with $\tilde{A}_k \in T_I\mathcal{S}^n_{++} \cong \mathcal{S}^n$. Together with Eq. (8), Eq. (18) can be further simplified.

Theorem 3.8 (SPD MLR under a PEM)**.** *Under any PEM, SPD MLR and SPD hyperplane is*

$$p(y=k \mid S) \propto \exp(\langle \phi(S) - \phi(P_k), \phi_{*,I}(\tilde{A}_k) \rangle), \quad (20)$$

$$\tilde{H}_{\tilde{A}_k, P_k} = \{S \in \mathcal{S}^n_{++} : \langle \phi(S) - \phi(P_k), \phi_{*,I}(\tilde{A}_k) \rangle = 0\}, \quad (21)$$

where $\tilde{A}_k \in T_I \mathcal{S}^n_{++}/\{0\} \cong \mathcal{S}^n/\{0\}$ *is a symmetric matrix, and* $P_k \in \mathcal{S}^n_{++}$ *is an SPD matrix.*

Proof. The proof is presented in App. C.5. □

4. SPD MLRs under deformed LEM and LCM

In this section, we first review the deformed LEM and LCM, and then we showcase our SPD MLR in Thm. 3.8 under these deformed metrics.

Inspired by the deforming utility of the matrix power function [49, 50], Chen *et al*. [15] define (θ, α, β)-LEM and (θ)-LCM as the pullback metric of (α, β)-LEM and LCM by matrix power function $(\cdot)^\theta$ and scaled by $\frac{1}{\theta^2}$ $(\theta \neq 0)$. As shown in [15, Props. 5.1], (θ, α, β)-LEM is equal to (α, β)-LEM and (θ)-LCM interpolates between the standard LCM $(\theta = 1)$ and an LEM-like metric $(\theta \to 0)$.

Besides, both (α, β)-LEM and (θ)-LCM are PEMs [15]. Therefore, the SPD MLRs under these two families of metrics can be directly obtained by Thm. 3.8.

Corollary 4.1 (SPD MLRs under the deformed LEM and LCM)**.** *The SPD MLRs under* (α, β)*-LEM is*

$$p(y=k \mid S) \propto \exp\left[\langle \mathrm{mlog}(S) - \mathrm{mlog}(P_k), \tilde{A}_k \rangle^{(\alpha,\beta)}\right], \quad (22)$$

where $\tilde{A}_k \in T_I \mathcal{S}^n_{++} \cong \mathcal{S}^n$ *and* $P_k \in \mathcal{S}^n_{++}$*. The SPD MLRs under* (θ)*-LCM is*

$$p(y=k \mid S) \propto \exp\left[\frac{1}{\theta}\langle X, Y \rangle\right], \quad (23)$$

with X *and* Y *defined as*

$$X = \lfloor \tilde{K} \rfloor - \lfloor \tilde{L}_k \rfloor + \left[\mathrm{Dlog}(\mathbb{D}(\tilde{K})) - \mathrm{Dlog}(\mathbb{D}(\tilde{L}_k))\right], \quad (24)$$

$$Y = \lfloor \tilde{A}_k \rfloor + \frac{1}{2}\mathbb{D}(\tilde{A}_k), \quad (25)$$

where $\tilde{K} = \mathrm{Chol}(S^\theta)$*,* $\tilde{L}_k = \mathrm{Chol}(P_k^\theta)$*, and* $\mathbb{D}(\tilde{A}_k)$ *denotes a diagonal matrix with diagonal elements of* $\tilde{A}_k$*.*

Proof. The proof is presented in App. C.6. □

$\mathcal{S}^2_{++}$ can be visualized as open cone in $\mathbb{R}^3$ by the condition that $\forall P = \begin{pmatrix} x & y \\ y & z \end{pmatrix} \in \mathcal{S}^2$ is positive definite iff $x, z > 0 \wedge xz > y^2$. Fig. 1 illustrates SPD hyperplanes induced by (α, β)-LEM and (θ)-LCM.

Remark 4.2. Our paper incorporates the results w.r.t. LEM and LCM presented in [42]. For (α, β)-LEM, when $(\alpha, \beta) = (1, 0)$, (α, β)-LEM becomes the standard LEM. Our margin distance to the hyperplane in Lem. 3.5 becomes the pseudo-gyrodistance under LEM [42, Thm. 2.23]. For (θ)-LCM, when $\theta = 1$, (θ)-LCM becomes the standard LCM. Our Lem. 3.5 becomes the pseudo-gyrodistance induced by LCM [42, Thm. 2.24]. However, our framework does not require gyro structures and directly obtains margin distance and SPD MLR based on the Riemannian metric.

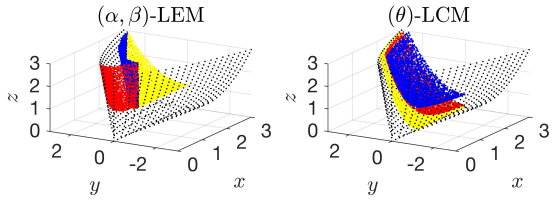

Figure 1. Conceptual illustration of SPD hyperplanes induced by (α, β)-LEM and (θ)-LCM. In each subfigure, the black dots are symmetric positive semi-definite (SPSD) matrices, denoting the boundary of $\mathcal{S}^2_{++}$, while the blue, red, and yellow dots denote three SPD hyperplanes.

5. Rethinking the existing LogEig classifier

Many of the existing SPD neural networks [5, 14, 29, 39, 43, 55, 56] rely on a Euclidean MLR in the codomain of matrix logarithm, *i.e*. matrix logarithm followed by an FC layer and a softmax layer. For simplicity, we call this classifier as LogEig MLR. The existing explanation of LogEig MLR is approximating manifolds by tangent space. However, our framework can offer a novel intrinsic explanation for this widely used MLR.

When $(\alpha, \beta) = (1, 0)$ for (α, β)-LEM, the SPD MLR in Eq. (22) is very similar to the LogEig MLR. However, due to the nonlinearity of $\mathrm{mlog}(\cdot)$ and the non-Euclideanness of SPD parameter P_k, SPD MLR cannot be hastily viewed as equivalent to LogEig MLR. Nevertheless, under special circumstances, Eq. (22) is indeed equivalent to a LogEig MLR.

Proposition 5.1. *Endowing SPD manifolds with the standard LEM, optimizing SPD parameter* P_k *in Eq. (22) by LEM-based RSGD and Euclidean parameter* A_k *by Euclidean SGD, the LEM-based SPD MLR is equivalent to a LogEig MLR with parameters in FC layer optimized by Euclidean SGD.*

Proof. The proof is presented in App. C.7. □

Prop. 5.1 implies that optimized by LEM-based RSGD, the LEM-based SPD MLR is equivalent to the Euclidean MLR in the codomain of matrix logarithm. Nevertheless,

a substantial body of prior works underscores the theoretical and empirical superiority of the AIM-based optimization over its LEM-based counterpart [23, 48]. Therefore, we adopt the AIM-based optimizer in this paper to update the involved SPD parameters.

6. Experiments

In this section, we implement the proposed two families of SPD MLRs to SPD neural networks. Note that our SPD MLRs are architecture-agnostic and can be applied to any existing SPD neural network. This paper focuses on two network architectures, SPDNet [29] and TSMNet+SPDDSMBN [31]. SPDNet is the most classic SPD neural network. Following previous works [5, 29], we evaluate our SPD MLRs under this architecture for radar recognition on the Radar dataset [5] and human action recognition on the HDM05 [38]. TSMNet+SPDDSMBN is the SOTA Riemannian approach to EEG classification, which is the improved version of SPDNetBN [5] for transfer learning on EEG tasks. We evaluate our SPD MLRs under this baseline for EEG classification on the Hinss2021 dataset [26].

6.1. Baseline models

SPDNet [29] mimics the conventional densely connected feedforward network, consisting of three basic building blocks

$$\text{BiMap: } S^k = W^k S^{k-1} W^k, \tag{26}$$

$$\text{ReEig: } S^k = U^{k-1} \max(\Sigma^{k-1}, \epsilon I_n) U^{k-1\top}, \tag{27}$$

$$\text{LogEig: } S^k = \phi_{mln}(S^{k-1}), \tag{28}$$

where $S^{k-1} = U^{k-1}\Sigma^{k-1}U^{k-1\top}$ is the eigendecomposition, and W^k is semi-orthogonal. The BiMap (Bilinear Mapping) is the counterpart of linear mapping in Euclidean networks. The ReEig (Eigenvalue Rectification) mimics the ReLu-like nonlinear activation functions by eigen-rectification. The LogEig layer projects SPD-valued data into the tangent space for further classification.

The architecture of TSMNet+SPDDSMBN [31] can be explained as

$$\begin{aligned} &f_{TC} \rightarrow f_{SC} \rightarrow f_{BiMap} \rightarrow f_{ReEig} \\ &\quad \rightarrow f_{SPDDSMBN} \rightarrow f_{LogEig}, \end{aligned} \tag{29}$$

where f_{TC} and f_{SC} denote temporal and spatial convolution, and $f_{SPDDSMBN}$ denotes SPD domain-specific momentum batch normalization, which is an SPD batch normalization layer for domain adaptation. For simplicity, we abbreviate TSMNet+SPDDSMBN as SPDDSMBN.

6.2. Datasets

Datasets and preprocessing: **Radar** dataset [5] contains 3,000 synthetic radar signals. Following the protocol in [5], each signal is split into windows of length 20, resulting in 3,000 covariance matrices of the size 20×20 equally distributed in 3 classes. **HDM05** dataset [38] consists of 2,273 skeleton-based motion capture sequences executed by different actors. Each frame can be represented as a 93×93 covariance matrix. In line with [5], we remove some under-represented clips and trim the dataset down to 2086 instances scattered throughout 117 classes. **Hinss2021** dataset [26] is a recently released competition dataset containing EEG signals for mental workload estimation. The dataset is employed for two tasks, namely inter-session and inter-subject classification, which are treated as domain adaptation problems. Recently, geometry-aware methods [31, 59] have demonstrated promising performance in EEG classification, due to the invariance to linear mixing of latent sources and interpretability of SPD modeling [31]. We follow the Python implementation[1] of Kobler et al. [31] for data preprocessing. In detail, the python package MOABB [30] and MNE [21] are used to preprocess the datasets. The applied steps include resampling the EEG signals to 250/256 Hz, applying temporal filters to extract oscillatory EEG activity in the 4 to 36 Hz range, extracting short segments ($\leq$ 3s) associated with a class label, and finally obtaining 40×40 SPD covariance matrices.

6.3. Implementation details

Network settings: The original classification in SPDNet and TSMNet is conducted by the LogEig MLR (matrix logarithm+FC+softmax). We substitute their LogEig classifiers with our intrinsic SPD MLRs to ensure a fair comparison. We use the standard cross-entropy loss as the training objective and optimize the parameters with the Riemannian AMSGrad optimizer [3]. The network architectures are represented as $[d_0, d_1, \ldots, d_L]$, where the dimension of the parameter in the i-th BiMap layer is $d_i \times d_{i-1}$. For the Radar and HDM05 datasets, we adopt a learning rate of $1e^{-2}$, a batch size of 30, and a maximum training epoch of 200. For the Hinss2021 dataset, in line with [31], we apply a learning rate of $1e^{-3}$ with a weight decay of $1e{-}4$, a batch size of 50, and a training epoch of 50. For better comparison, we also implement the AIM-based gyro SPD MLR [42] to SPDNet and TSMNet, which is named SPDNet+Gyro-AIM or TSMNet+Gyro-AIM. All experiments use an Intel Core i9-7960X CPU with 32GB RAM and an NVIDIA GeForce RTX 2080 Ti GPU.

Evaluation methods: In line with the previous work [29, 31], we use accuracy as the scoring metric for the Radar and HDM05 datasets, and balanced accuracy (*i.e.* the average recall across classes) for the Hinss2021 dataset. Ten-fold experiments on the Radar and HDM05 datasets are carried out with randomized initialization and split, while on the Hinss2021 dataset, models are fit and evaluated with a

[1] https://github.com/rkobler/TSMNet

Backbone	Classifier	[20,16,8]	[20,16,14,12,10,8]
SPDNet	LogEig MLR	92.88±1.05	93.47±0.45
	Gyro-AIM	94.53±0.95	94.32±0.94
	(1,0)-LEM	93.55±1.21	94.60±0.70
	(1,1)-LEM	**95.64±0.83**	**95.87±0.58**
	(1)-LCM	93.49±1.25	93.93±0.98
	(0.5)-LCM	**94.59±0.82**	**95.16±0.67**

Table 3. Results of SPDNet with different classifiers on the Radar dataset.

Backbone	Classifier	[93,30]	[93,70,30]	[93,70,50,30]
SPDNet	LogEig MLR	57.42±1.31	60.69±0.66	60.76±0.80
	Gyro-AIM	58.07±0.64	60.72±0.62	61.14±0.94
	(1,0)-LEM	57.02±0.75	61.34±0.62	60.78±0.86
	(1)-LCM	62.04±1.05	62.11±2.11	62.89±2.09
	(0.5)-LCM	**65.66±0.73**	**65.79±0.63**	**65.71±0.75**

Table 4. Results of SPDNet with different classifiers on the HDM05 dataset.

randomized leave 5% of the sessions (inter-session) or subjects (inter-subject) out cross-validation scheme.

Hyper-parameters: We implement the SPD MLRs induced by both the standard metrics and parameterized metrics ((α,β)-LEM and (θ)-LCM). Therefore, in our SPD MLRs, we have one or two hyper-parameters, *i.e.* θ in (θ)-LCM and (α,β) in (α,β)-LEM, where θ controls deformation and (α,β) are associated with $\mathrm{O}(n)$-invariance. Recalling Eq. (2), α is a scaling factor, while β measures the relative significance of traces. As scaling is less important [49], we set $\alpha = 1$. We select the value of β from the candidate set $\{1, {}^1\!/_n, {}^1\!/_{n^2}, 0, -{}^1\!/_n + \epsilon, -{}^1\!/_{n^2}\}$, where n is the dimension of input SPD matrices in SPD MLRs[2]. These chosen values for β allow for amplifying, neutralizing, or suppressing the trace components, depending on the characteristics of the datasets. For the deformation factor θ, we roughly select its values around the deformation boundary. Specifically, for (θ)-LCM, θ is selected from the set $\{0.5, 1, 1.5\}$.

6.4. Experimental results

For each family of SPD MLRs, we report two representative baselines: the standard SPD MLR induced from the standard metric ($\theta = 1, \alpha = 1, \beta = 0$), and the one induced from the parameterized metric with selected hyper-parameters. If the standard SPD MLR is already saturated, we only report the results of the standard ones. In Tabs. 3 to 5, we denote (α,β)-LEM ((θ)-LCM) as the baseline model endowed with the SPD MLR induced by (α,β)-LEM ((θ)-LCM).

[2]The purpose of including a small positive constant $\epsilon \in \mathbb{R}_+$ is to ensure $\mathrm{O}(n)$-invariance, *i.e.* $(\alpha,\beta) \in \mathbf{ST}$.

Radar: In line with [5], we evaluated our classifiers on the Radar dataset under two network architectures: [20, 16, 8] for the 2-layer configuration and [20, 16, 14, 12, 10, 8] for the 5-layer configuration. The 10-fold results (mean±std) are presented in Tab. 3. Generally speaking, our SPD MLRs achieve superior performance against the vanilla LogEig MLR. Among all SPD MLRs, the ones induced by (1,1)-LEM achieve the best performance on this dataset. Although the SPD MLRs induced by standard LEM and LCM are slightly worse than the AIM-based gyro SPD MLR, our SPD MLRs with proper hyper-parameters achieve comparable or even better performance than the AIM-based gyro SPD MLR. For both (α,β)-LEM ((θ)-LCM), the associated SPD MLR with proper hyper-parameters (α,β) (θ) outperforms the standard SPD MLR induced by the standard metrics, demonstrating the effectiveness of our parameterization.

HDM05: Following [29], three architectures are evaluated on this dataset: [93, 30], [93, 70, 30], and [93, 70, 50, 30]. The SPD MLR under the standard LEM is already saturated on this dataset and performs similarly to the vanilla LogEig MLR. This phenomenon might be attributed to the equivalence of a LogEig MLR with an SPD MLR optimized by LEM, which is detailed in Prop. 5.1. Nevertheless, the SPD MLR based on (θ)-LCM achieves the best performance under different network architectures, improving the vanilla SPDNet by a large margin. Particularly, (0.5)-LCM demonstrates a clear advantage over the vanilla LogEig MLR. Besides, the LCM-based SPD MLR consistently performs better than the Gyro-AIM SPD MLR. More interestingly, the power deformation not only improves the absolute accuracy of the LCM-based SPD MLR but also reduces the standard deviation, indicating the significance of our deformation. These phenomena demonstrate the advantage of our framework's versatility.

Backbone	Classifier	Inter-session	Inter-subject
SPDDSMBN	LogEig MLR	53.83±9.77	49.68±7.88
	Gyro-AIM	53.36±9.92	50.65±8.13
	(1,0)-LEM	53.16±9.73	51.41±7.98
	(1)-LCM	55.71±8.57	51.60±8.43
	(1.5)-LCM	**56.43±8.79**	**51.65±5.90**

Table 5. Results of SPDDSMBN with different classifiers on the Hinss2021 dataset under inter-subject and inter-session scenarios. The presented results are the ones of balanced accuracy under the leaving 5% out cross-validation scenario.

Hinss2021: Following [31], we adopt the architecture of [40,20]. The results (mean±std) of leaving 5% out cross-validation are presented in Tab. 5. Once again, our intrinsic classifiers demonstrate improved performance compared to the baseline in the inter-session and inter-subject scenarios. The SPD MLRs based on (θ)-LCM achieve the best perfor-

mance (increase 2.6% for inter-session and 1.97% for inter-subject), indicating that this metric can faithfully capture the geometry of data in the Hinss2021 dataset. Besides, the SPD MLR based on powered-deformed LCM shows the least standard deviation compared with other classifiers, demonstrating the significance of the power deformation. These findings highlight the adaptability and versatility of our framework, as it can effectively leverage different Riemannian metrics based on the intrinsic geometry of the data.

Methods	Radar	HDM05	Hinss2021	
			inter-session	inter-subject
Baseline	1.36	1.95	0.18	8.31
MLR-Gyro-AIM	1.75	31.64	0.38	13.3
MLR-LEM	1.5	4.7	0.24	10.13
MLR-LCM	**1.35**	**3.29**	**0.18**	**8.35**

Table 6. Comparison of training efficiency (s/epoch) of SPDNet (SPDDSMBN) under different classifiers. The most efficient MLR is highlighted in **bold**.

6.5. Model efficiency

We adopt the deepest architectures, namely [20, 16, 14, 12, 10, 8] for the Radar dataset, [93, 70, 50, 30] for the HDM05 dataset, and [40, 20] for the Hinss2021 dataset. For simplicity, we focus on the SPD MLRs induced by standard metrics, *i.e*. LEM and LCM. We also implement AIM-based gyro SPD MLR. The average training time (in seconds) per epoch is reported in Tab. 6. In general, when compared to the AIM-based gyro SPD MLR, LEM- and LCM-based SPD MLRs exhibit superior efficiency, especially when dealing with a larger number of classes. Due to the computational complexity of AIM, the AIM-based SPD MLR involves more matrix computation, incurring higher computational costs. In contrast, due to the rapid computation of PEMs, the PEM-based SPD MLR is more computationally efficient. This contrast becomes more obvious when dealing with a huge number of classes, as each class requires an SPD parameter, which needs to be processed by Riemannian computations. For instance, on the HDM05 dataset, which comprises 117 classes, the LEM- and LCM-based SPD MLRs require only one-ninth training time compared to the AIM-based gyro SPD MLR.

6.6. Additional discussions on Gyro SPD MLR

Theoretically speaking, AIM should generally be more powerful than LEM and LCM, as AIM enjoys affine invariance, which is powerful for modeling covariance matrices. However, as shown in Tabs. 3 to 5, the improvement of Gyro-AIM SPD MLR for SPDNet is not very significant and is outperformed by our deformed SPD MLR. This could be attributed to two reasons. Firstly, as we discussed at the end of Sec. 3.1, Gyro-AIM MLR did not solve the real margin distance. Instead, its margin distance is defined by Gyro distance, which is defined by the tangent space at the identity. However, the most natural distance in manifolds is the geodesic distance. Therefore, this might undermine the overall performance of Gyro-AIM MLR. On the contrary, our deformed MLRs are developed by the true margin distance. Secondly, the deformation we adopt can interpolate between different metrics, capturing more vibrant geometry and benefiting our SPD MLR.

Besides, our framework in Thm. 3.8 enjoys better flexibility than Gyro SPD MLR [42]. Gyro SPD MLR relies on the gyro structures. Given a Riemannian metric, one should first verify whether the induced gyro operations [42, Eqs. (1-2)] conforms with the 10 axioms of the gyro space [40, Defs. 2.1 - 2.3]. Besides, one should also solve the induced Gyro SPD MLR [42] based on the verified gyro vector space. The above process is a case-by-case process. However, for PEMs, SPD MLRs can be readily obtained by our Thm. 3.8, such as the ones induced by LEM, LCM, and their deformed metrics.

7. Conclusion

This paper provides a framework for building SPD MLR under any PEM. We showcase our framework under the parameterized metrics of LEM and LCM. Our framework also provides an intrinsic explanation for the widely used LogEig classifier. The consistent superior performance in extensive experiments also supports our claims. As a future avenue, our framework can also be applied to other kinds of PEMs.

Limitations and future work. This paper constructs SPD MLRs under PEMs, including LEM, LCM, and their variants. In the future, we will develop SPD MLRs under other metrics, such as AIM. Notably, though AIM-based SPD MLR has been developed [42], the margin distance is pseudo-gyrodistance, which does not solve Eq. (14). Besides, several Euclidean backbones involve SPD features to be classified [17, 19, 53]. However, the SPD features in these backbones are usually of large dimensions, bringing computational burdens for the Riemannian optimization in our SPD MLR. In the future, we will explore accelerated optimization to apply our MLR to Euclidean backbones.

Acknowledgements

This work was partly supported by the MUR PNRR project FAIR (PE00000013) funded by the NextGenerationEU, the EU Horizon project ELIAS (No. 101120237), and a donation from Cisco. The authors also gratefully acknowledge the financial support from the China Scholarship Council (CSC).

References

[1] Marc Arnaudon, Frédéric Barbaresco, and Le Yang. Riemannian medians and means with applications to radar signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 7(4):595–604, 2013. 1

[2] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. *Fast and simple computations on tensors with log-Euclidean metrics.* PhD thesis, INRIA, 2005. 1, 2

[3] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018. 6

[4] Silvere Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013. 3

[5] Daniel Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for SPD neural networks. In *Advances in Neural Information Processing Systems*, 2019. 1, 5, 6, 7

[6] Daniel A Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Exploring complex time-series representations for Riemannian machine learning of radar data. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3672–3676. IEEE, 2019. 1

[7] Rudrasis Chakraborty. ManifoldNorm: Extending normalizations on Riemannian manifolds, 2020. 1

[8] 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. *Advances in Neural Information Processing Systems*, 31, 2018. 1

[9] Rudrasis Chakraborty, Jose Bouza, Jonathan Manton, and Baba C Vemuri. Manifoldnet: A deep neural network for manifold-valued data with applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[10] Kaixuan Chen, Jie Song, Shunyu Liu, Na Yu, Zunlei Feng, Gengshi Han, and Mingli Song. Distribution knowledge embedding for graph pooling. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):7898–7908, 2023. 1

[11] Kai-Xuan Chen, Jie-Yi Ren, Xiao-Jun Wu, and Josef Kittler. Covariance descriptors on a Gaussian manifold and their application to image set classification. *Pattern Recognition*, 107:107463, 2020. 1

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

[13] Ziheng Chen, Tianyang Xu, Zhiwu Huang, Yue Song, Xiao-Jun Wu, and Nicu Sebe. Adaptive Riemannian metrics on SPD manifolds. *arXiv preprint arXiv:2303.15477*, 2023. 2, 3

[14] Ziheng Chen, Tianyang Xu, Xiao-Jun Wu, Rui Wang, Zhiwu Huang, and Josef Kittler. Riemannian local mechanism for SPD neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7104–7112, 2023. 1, 5

[15] Ziheng Chen, Yue Song, Yunmei Liu, and Nicu Sebe. A Lie group approach to Riemannian batch normalization. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 5

[16] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian Geometry*. Springer, 1992. 2, 1

[17] Melih Engin, Lei Wang, Luping Zhou, and Xinwang Liu. DeepKSPD: Learning kernel-matrix-based SPD representation for fine-grained image recognition. In *Proceedings of the European Conference on Computer Vision*, pages 612–627, 2018. 8

[18] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018. 1, 3, 4

[19] Zhi Gao, Yuwei Wu, Xingyuan Bu, Tan Yu, Junsong Yuan, and Yunde Jia. Learning a robust representation via a deep network on symmetric positive definite manifolds. *Pattern Recognition*, 92:1–12, 2019. 8

[20] Zhi Gao, Yuwei Wu, Mehrtash Harandi, and Yunde Jia. A robust distance measure for similarity-based classification on the spd manifold. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3230–3244, 2019. 1

[21] Alexandre Gramfort. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7, 2013. 6

[22] Johann Guilleminot and Christian Soize. Generalized stochastic approach for constitutive equation in linear elasticity: a random matrix model. *International Journal for Numerical Methods in Engineering*, 90(5):613–635, 2012. 1

[23] Andi Han, Bamdev Mishra, Pratik Kumar Jawanpuria, and Junbin Gao. On Riemannian optimization over positive definite matrices with the Bures-Wasserstein geometry. *Advances in Neural Information Processing Systems*, 34:8940–8953, 2021. 6

[24] Mehrtash Harandi, Mathieu Salzmann, and Richard Hartley. Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):48–62, 2018. 1

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[26] Marcel F. Hinss, Ludovic Darmet, Bertille Somon, Emilie Jahanpour, Fabien Lotte, Simon Ladouce, and Raphaëlle N. Roy. An EEG dataset for cross-session mental workload estimation: Passive BCI competition of the Neuroergonomics Conference 2021, 2021. 6

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 1

[28] Xiaoqiang Hua, Yongqiang Cheng, Hongqiang Wang, Yuliang Qin, Yubo Li, and Wenpeng Zhang. Matrix CFAR detectors based on symmetrized Kullback–Leibler and total Kullback–Leibler divergences. *Digital Signal Processing*, 69:106–116, 2017. 1

[29] Zhiwu Huang and Luc Van Gool. A Riemannian network for SPD matrix learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 1, 5, 6, 7

[30] Vinay Jayaram and Alexandre Barachant. MOABB: trustworthy algorithm benchmarking for BCIs. *Journal of Neural Engineering*, 15(6):066011, 2018. 6

[31] Reinmar Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe. SPD domain-specific batch normalization to crack interpretable unsupervised domain adaptation in EEG. *Advances in Neural Information Processing Systems*, 35:6219–6235, 2022. 1, 6, 7

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012. 1

[33] Guy Lebanon and John Lafferty. Hyperplane margin classifiers on the multinomial manifold. In *Proceedings of the twenty-first international conference on Machine learning*, page 66, 2004. 3, 4

[34] John M Lee. *Introduction to smooth manifolds*. Springer, 2013. 1

[35] Zhenhua Lin. Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1353–1370, 2019. 1, 2, 3

[36] Federico López, Beatrice Pozzetti, Steve Trettel, Michael Strube, and Anna Wienhard. Vector-valued distance and Gyrocalculus on the space of symmetric positive definite matrices. *Advances in Neural Information Processing Systems*, 34:18350–18366, 2021. 1

[37] Maher Moakher. On the averaging of symmetric positive-definite tensors. *Journal of Elasticity*, 82(3):273–296, 2006. 1

[38] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Documentation mocap database HDM05. Technical report, Universität Bonn, 2007. 6

[39] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 13379–13389, 2021. 1, 5

[40] Xuan Son Nguyen. The Gyro-structure of some matrix manifolds. In *Advances in Neural Information Processing Systems*, pages 26618–26630, 2022. 1, 8

[41] Xuan Son Nguyen. A Gyrovector space approach for symmetric positive semi-definite matrix learning. In *Proceedings of the European Conference on Computer Vision*, pages 52–68, 2022. 1

[42] Xuan Son Nguyen and Shuo Yang. Building neural networks on matrix manifolds: A Gyrovector space approach. *arXiv preprint arXiv:2305.04560*, 2023. 1, 3, 4, 5, 6, 8

[43] Xuan Son Nguyen, Luc Brun, Olivier Lézoray, and Sébastien Bougleux. A neural network based on SPD manifold learning for skeleton-based hand gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12036–12045, 2019. 5

[44] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006. 1, 2

[45] Yue Song, Nicu Sebe, and Wei Wang. Why approximate matrix square root outperforms accurate SVD in global covariance pooling? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1115–1123, 2021. 1

[46] Yue Song, Nicu Sebe, and Wei Wang. On the eigenvalues of global covariance pooling for fine-grained visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3554–3566, 2022. 1

[47] Yue Song, Nicu Sebe, and Wei Wang. Fast differentiable matrix square root and inverse square root. *arXiv preprint arXiv:2201.12543*, 2022. 1

[48] Suvrit Sra and Reshad Hosseini. Conic geometric optimization on the manifold of positive definite matrices. *SIAM Journal on Optimization*, 25(1):713–739, 2015. 6

[49] Yann Thanwerdas and Xavier Pennec. Is affine-invariance well defined on SPD matrices? a principled continuum of metrics. In *Geometric Science of Information: 4th International Conference, GSI 2019, Toulouse, France, August 27–29, 2019, Proceedings 4*, pages 502–510. Springer, 2019. 5, 7

[50] Yann Thanwerdas and Xavier Pennec. The geometry of mixed-Euclidean metrics on symmetric positive definite matrices. *Differential Geometry and its Applications*, 81: 101867, 2022. 4, 5

[51] Yann Thanwerdas and Xavier Pennec. O (n)-invariant Riemannian metrics on SPD matrices. *Linear Algebra and its Applications*, 661:163–201, 2023. 2, 4

[52] Loring W.. Tu. *An introduction to manifolds*. Springer, 2011. 2, 4, 1

[53] Qilong Wang, Peihua Li, and Lei Zhang. G2DeNet: Global Gaussian distribution embedding network and its application to visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2017. 8

[54] Qilong Wang, Jiangtao Xie, Wangmeng Zuo, Lei Zhang, and Peihua Li. Deep CNNs meet global covariance pooling: Better representation and generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2582–2597, 2020. 1

[55] Rui Wang, Xiao-Jun Wu, and Josef Kittler. SymNet: A simple symmetric positive definite manifold deep learning method for image set classification. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5):2208–2222, 2021. 1, 5

[56] Rui Wang, Xiao-Jun Wu, Ziheng Chen, Tianyang Xu, and Josef Kittler. DreamNet: A deep Riemannian manifold network for SPD matrix learning. In *Proceedings of the Asian Conference on Computer Vision*, pages 3241–3257, 2022. 1, 5

[57] Rui Wang, Xiao-Jun Wu, Ziheng Chen, Tianyang Xu, and Josef Kittler. Learning a discriminative SPD manifold neural network for image set classification. *Neural networks*, 151: 94–110, 2022. 1

[58] Rui Wang, Xiao-Jun Wu, Ziheng Chen, Cong Hu, and Josef Kittler. SPD manifold deep metric learning for image set classification. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2024. 1

[59] Or Yair, Mirela Ben-Chen, and Ronen Talmon. Parallel transport on the cone manifold of SPD matrices for domain adaptation. *IEEE Transactions on Signal Processing*, 67(7): 1797–1811, 2019. 6

[60] Tong Zhang, Wenming Zheng, Zhen Cui, Yuan Zong, Chaolong Li, Xiaoyan Zhou, and Jian Yang. Deep manifold-to-manifold transforming network for skeleton-based action recognition. *IEEE Transactions on Multimedia*, 22(11): 2926–2937, 2020. 1

[61] Xingjian Zhen, Rudrasis Chakraborty, Nicholas Vogt, Barbara B Bendlin, and Vikas Singh. Dilated convolutional neural networks for sequential manifold-valued data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10621–10631, 2019. 1