A Gyrovector Space Approach for Symmetric Positive Semi-definite Matrix Learning

Xuan Son Nguyen[0000−0002−2776−2254]

ETIS, UMR 8051, CY Cergy Paris Université, ENSEA, CNRS, Cergy, France
xuan-son.nguyen@ensea.fr

Abstract. Representation learning with Symmetric Positive Semi-definite (SPSD) matrices has proven effective in many machine learning problems. Recently, some SPSD neural networks have been proposed and shown promising performance. While these works share a common idea of generalizing some basic operations in deep neural networks (DNNs) to the SPSD manifold setting, their proposed generalizations are usually designed in an ad hoc manner. In this work, we make an attempt to propose a principled framework for building such generalizations. Our method is motivated by the success of hyperbolic neural networks (HNNs) that have demonstrated impressive performance in a variety of applications. At the heart of HNNs is the theory of gyrovector spaces that provides a powerful tool for studying hyperbolic geometry. Here we consider connecting the theory of gyrovector spaces and the Riemannian geometry of SPSD manifolds. We first propose a method to define basic operations, i.e., binary operation and scalar multiplication in gyrovector spaces of (full-rank) Symmetric Positive Definite (SPD) matrices. We then extend these operations to the low-rank SPSD manifold setting. Finally, we present an approach for building SPSD neural networks. Experimental evaluations on three benchmarks for human activity recognition demonstrate the efficacy of our proposed framework.

1 Introduction

SPSD matrices are computational objects that are commonly encountered in various applied areas such as medical imaging [2,35], shape analysis [42], drone classification [6], image recognition [10], and human behavior analysis [9,14,15,17,32,41]. Due to the non-Euclidean nature of SPSD matrices, traditional machine learning algorithms usually fail to obtain good results when it comes to analyzing such data. This has led to extensive studies on the Riemannian geometry of SPSD matrices [2,3,4,24,30,36].

In recent years, deep learning methods have brought breakthroughs in many fields of machine learning. Inspired by the success of deep learning and the modeling power of SPSD matrices, some recent works [6,7,8,9,17,32,34,47] have proposed different approaches for building SPSD neural networks. Although these networks have shown promising performance, their layers are usually designed in an ad hoc manner[1]. For example, in [8], the translation operation that is claimed to be analogous to adding a bias term in Euclidean neural networks is constructed from the action of the orthogonal group on SPD manifolds. However, this operation does not allow one to perform addition of two arbitrary SPD matrices and thus is not fully analogous to the vector addition in Euclidean spaces.

To tackle the above problem, we rely on the theory of gyrovector spaces [44,45] that has been successfully applied in the context of HNNs [12,40]. However, in order to apply this theory to the SPSD manifold setting, we first need to define the basic operations in gyrovector spaces of SPSD matrices. Although there are some works [1,16,18,19,20,21,28] studying gyrovector spaces of SPD matrices with an Affine-invariant (AI) [36] geometry, none of them provides a rigorous mathematical formulation for the connection between the basic operations in these spaces and the Riemannian geometry of SPD manifolds. In this paper, we take one step further toward a deeper understanding of gyrovector spaces of SPD matrices by proposing a principled method to define the basic operations in these spaces. We focus on three different geometries of SPD manifolds, i.e., Affine-invariant, Log-Euclidean (LE) [2], Log-Cholesky (LC) [24], and derive compact expressions for the basic operations associated with these geometries. To extend our method to low-rank SPSD manifolds, we make use of their quotient geometry [3,4]. We show how to define the loose versions of the basic operations on these manifolds, and study some of their properties. For SPSD matrix learning, we consider the full-rank and low-rank learning settings, and develop a class of Recurrent Neural Networks (RNNs) with flexible learning strategies in these settings.

2 Gyrovector Spaces

Gyrovector spaces form the setting for hyperbolic geometry in the same way that vector spaces form the setting for Euclidean geometry [44,45,46]. We first recap the definitions of gyrogroups and gyrocommutative gyrogroups proposed in [44,45,46]. For greater mathematical detail and in-depth discussion, we refer the interested reader to these papers.

Definition 1 (Gyrogroups [46]). *A pair $(G, \oplus)$ is a groupoid in the sense that it is a nonempty set, G, with a binary operation, $\oplus$. A groupoid $(G, \oplus)$ is a gyrogroup if its binary operation satisfies the following axioms for $a, b, c \in G$:*

(G1) There is at least one element $e \in G$ called a left identity such that $e \oplus a = a$.

(G2) There is an element $\ominus a \in G$ called a left inverse of a such that $\ominus a \oplus a = e$.

(G3) There is an automorphism $\mathrm{gyr}[a, b] : G \to G$ for each $a, b \in G$ such that

$$a \oplus (b \oplus c) = (a \oplus b) \oplus \mathrm{gyr}[a, b]c \quad \textit{(Left Gyroassociative Law).}$$

The automorphism $\mathrm{gyr}[a, b]$ is called the gyroautomorphism, or the gyration of G generated by a, b.

[1] In terms of basic operations used to build the network layers.

(G4) $\text{gyr}[a, b] = \text{gyr}[a \oplus b, b]$ *(Left Reduction Property).*

Definition 2 (Gyrocommutative Gyrogroups [46]). *A gyrogroup* $(G, \oplus)$ *is gyrocommutative if it satisfies*

$$a \oplus b = \text{gyr}[a, b](b \oplus a) \quad \textit{(Gyrocommutative Law).}$$

The following definition of gyrovector spaces is slightly different from Definition 3.2 in [46].

Definition 3 (Gyrovector Spaces). *A gyrocommutative gyrogroup* $(G, \oplus)$ *equipped with a scalar multiplication*

$$(t, x) \to t \odot x : \mathbb{R} \times G \to G$$

is called a gyrovector space if it satisfies the following axioms for $s, t \in \mathbb{R}$ *and* $a, b, c \in G$*:*

(V1) $1 \odot a = a, 0 \odot a = t \odot e = e,$ *and* $(-1) \odot a = \ominus a.$
(V2) $(s + t) \odot a = s \odot a \oplus t \odot a.$
(V3) $(st) \odot a = s \odot (t \odot a).$
(V4) $\text{gyr}[a, b](t \odot c) = t \odot \text{gyr}[a, b]c.$
(V5) $\text{gyr}[s \odot a, t \odot a] = \text{Id}$*, where* Id *is the identity map.*

Note that the axioms of gyrovector spaces considered in our work are more strict than those in [18,19,20,21]. Thus, many results proved in these works can also be applied to our case, which gives rise to interesting applications.

3 Proposed Approach

For simplicity of exposition, we will concentrate on real matrices. Denote by Sym_n^+ the space of $n \times n$ SPD matrices, $\text{S}_{n,p}^+$ the space of $n \times n$ SPSD matrices of rank $p < n$, $\text{V}_{n,p}$ the space of $n \times p$ matrices with orthonormal columns, O_n the space of $n \times n$ orthonormal matrices. Let $\mathcal{M}$ be a Riemannian homogeneous space, $T_{\mathbf{P}}\mathcal{M}$ be the tangent space of $\mathcal{M}$ at $\mathbf{P} \in \mathcal{M}$. Denote by $\exp(\mathbf{P})$ and $\log(\mathbf{P})$ the usual matrix exponential and logarithm of $\mathbf{P}$, $\text{Exp}_{\mathbf{P}}(\mathbf{W})$ the exponential map at $\mathbf{P}$ that associates to a tangent vector $\mathbf{W} \in T_{\mathbf{P}}\mathcal{M}$ a point of $\mathcal{M}$, $\text{Log}_{\mathbf{P}}(\mathbf{Q})$ the logarithmic map of $\mathbf{Q} \in \mathcal{M}$ at $\mathbf{P}$. Let $\mathcal{T}_{\mathbf{P} \to \mathbf{Q}}(\mathbf{W})$ be the parallel transport of $\mathbf{W}$ from $\mathbf{P}$ to $\mathbf{Q}$ along geodesics connecting $\mathbf{P}$ and $\mathbf{Q}$.

Definition 4. *The binary operation* $\mathbf{P} \oplus \mathbf{Q}$ *where* $\mathbf{P}, \mathbf{Q} \in \mathcal{M}$ *is obtained by projecting* $\mathbf{Q}$ *in the tangent space at the identity element* $\mathbf{I} \in \mathcal{M}$ *with the logarithmic map, computing the parallel transport of this projection from* $\mathbf{I}$ *to* $\mathbf{P}$ *along geodesics connecting* $\mathbf{I}$ *and* $\mathbf{P}$*, and then projecting it back on the manifold with the exponential map, i.e.,*

$$\mathbf{P} \oplus \mathbf{Q} = \text{Exp}_{\mathbf{P}}(\mathcal{T}_{\mathbf{I} \to \mathbf{P}}(\text{Log}_{\mathbf{I}}(\mathbf{Q}))). \tag{1}$$

Definition 5. *The scalar multiplication* $t \otimes \mathbf{P}$ *where* $t \in \mathbb{R}$ *and* $\mathbf{P} \in \mathcal{M}$ *is obtained by projecting* $\mathbf{P}$ *in the tangent space at the identity element* $\mathbf{I} \in \mathcal{M}$ *with the logarithmic map, multiplying this projection by the scalar* t *in* $T_{\mathbf{I}}\mathcal{M}$*, and then projecting it back on the manifold with the exponential map, i.e.,*

$$t \otimes \mathbf{P} = \mathrm{Exp}_{\mathbf{I}}(t \, \mathrm{Log}_{\mathbf{I}}(\mathbf{P})). \tag{2}$$

In the next sections, we will define gyrovector spaces of SPD matrices with the Affine-invariant, Log-Euclidean, and Log-Cholesky geometries (see supplementary material for a review of these geometries).

3.1 Affine-invariant Gyrovector Spaces

We first examine SPD manifolds with the Affine-invariant geometry. Lemma 1 gives a compact expression for the binary operation.

Lemma 1. *For* $\mathbf{P}, \mathbf{Q} \in \mathrm{Sym}_n^+$*, the binary operation* $\mathbf{P} \oplus_{ai} \mathbf{Q}$ *is given as*

$$\mathbf{P} \oplus_{ai} \mathbf{Q} = \mathbf{P}^{\frac{1}{2}} \mathbf{Q} \mathbf{P}^{\frac{1}{2}}. \tag{3}$$

Proof. See supplementary material.

The identity element of Sym_n^+ is the $n \times n$ identity matrix $\mathbf{I}_n$. Then, from Eq. (3), the inverse of $\mathbf{P}$ is given by

$$\ominus_{ai} \mathbf{P} = \mathbf{P}^{-1}.$$

Lemma 2. *For* $\mathbf{P} \in \mathrm{Sym}_n^+$*,* $t \in \mathbb{R}$*, the scalar multiplication* $t \otimes_{ai} \mathbf{P}$ *is given as*

$$t \otimes_{ai} \mathbf{P} = \mathbf{P}^t. \tag{4}$$

Proof. See supplementary material.

Definition 6 (Affine-invariant Gyrovector Spaces). *Define a binary operation* $\oplus_{ai}$ *and a scalar multiplication* $\otimes_{ai}$ *by Eqs. (3) and (4), respectively. Define a gyroautomorphism generated by* $\mathbf{P}$ *and* $\mathbf{Q}$ *as*

$$\mathrm{gyr}_{ai}[\mathbf{P}, \mathbf{Q}]\mathbf{R} = F_{ai}(\mathbf{P}, \mathbf{Q}) \mathbf{R} (F_{ai}(\mathbf{P}, \mathbf{Q}))^{-1}, \tag{5}$$

where $F_{ai}(\mathbf{P}, \mathbf{Q}) = (\mathbf{P}^{\frac{1}{2}} \mathbf{Q} \mathbf{P}^{\frac{1}{2}})^{-\frac{1}{2}} \mathbf{P}^{\frac{1}{2}} \mathbf{Q}^{\frac{1}{2}}$.

Theorem 1. *Gyrogroups* $(\mathrm{Sym}_n^+, \oplus_{ai})$ *with the scalar multiplication* $\otimes_{ai}$ *form gyrovector spaces* $(\mathrm{Sym}_n^+, \oplus_{ai}, \otimes_{ai})$.

Proof. See supplementary material.

3.2 Log-Euclidean Gyrovector Spaces

We now study SPD manifolds with the Log-Euclidean geometry.

Lemma 3. *For $\mathbf{P}, \mathbf{Q} \in \mathrm{Sym}_n^+$, the binary operation $\mathbf{P} \oplus_{le} \mathbf{Q}$ is given as*

$$\mathbf{P} \oplus_{le} \mathbf{Q} = \exp(\log(\mathbf{P}) + \log(\mathbf{Q})). \tag{6}$$

Proof. See supplementary material.

From Lemma 3, the inverse of $\mathbf{P}$ is given by

$$\ominus_{le}\mathbf{P} = \mathbf{P}^{-1}.$$

Lemma 4. *For $\mathbf{P} \in \mathrm{Sym}_n^+$, $t \in \mathbb{R}$, the scalar multiplication $t \otimes_{le} \mathbf{P}$ is given by*

$$t \otimes_{le} \mathbf{P} = \mathbf{P}^t. \tag{7}$$

Proof. See supplementary material.

Definition 7 (Log-Euclidean Gyrovector Spaces). *Define a binary operation $\oplus_{le}$ and a scalar multiplication $\otimes_{le}$ by Eqs. (6) and (7), respectively. Define a gyroautomorphism generated by $\mathbf{P}$ and $\mathbf{Q}$ as*

$$\mathrm{gyr}_{le}[\mathbf{P}, \mathbf{Q}] = \mathrm{Id}\,.$$

Theorem 2. *Gyrogroups $(\mathrm{Sym}_n^+, \oplus_{le})$ with the scalar multiplication $\otimes_{le}$ form gyrovector spaces $(\mathrm{Sym}_n^+, \oplus_{le}, \otimes_{le})$.*

Proof. See supplementary material.

The conclusion of Theorem 2 is not surprising since it is known [2] that the space of SPD matrices with the Log-Euclidean geometry has a vector space structure. This vector space structure is given by the operations proposed in [2] that turn out to be the binary operation and scalar multiplication in Log-Euclidean gyrovector spaces.

3.3 Log-Cholesky Gyrovector Spaces

In this section, we focus on SPD manifolds with the Log-Cholesky geometry. Following the notations in [24], let $\lfloor \mathbf{A} \rfloor$ be a matrix of the same size as matrix $\mathbf{A}$ whose (i, j) element is $\mathbf{A}_{ij}$ if $i > j$ and is zero otherwise, $\mathbb{D}(\mathbf{A})$ is a diagonal matrix of the same size as matrix $\mathbf{A}$ whose (i, i) element is $\mathbf{A}_{ii}$. Denote by $\mathscr{L}(\mathbf{P})$ the lower triangular matrix obtained from the Cholesky decomposition of matrix $\mathbf{P} \in \mathrm{Sym}_n^+$, i.e., $\mathbf{P} = \mathscr{L}(\mathbf{P})\mathscr{L}(\mathbf{P})^T$.

Lemma 5. *For $\mathbf{P}, \mathbf{Q} \in \mathrm{Sym}_n^+$, the binary operation $\mathbf{P} \oplus_{lc} \mathbf{Q}$ is given as*

$$\begin{aligned}\mathbf{P} \oplus_{lc} \mathbf{Q} = &\big(\lfloor \mathscr{L}(\mathbf{P}) \rfloor + \lfloor \mathscr{L}(\mathbf{Q}) \rfloor + \mathbb{D}(\mathscr{L}(\mathbf{P}))\mathbb{D}(\mathscr{L}(\mathbf{Q}))\big). \\ &\big(\lfloor \mathscr{L}(\mathbf{P}) \rfloor + \lfloor \mathscr{L}(\mathbf{Q}) \rfloor + \mathbb{D}(\mathscr{L}(\mathbf{P}))\mathbb{D}(\mathscr{L}(\mathbf{Q}))\big)^T.\end{aligned} \tag{8}$$

Proof. See supplementary material.

From Eq. (8), the inverse of $\mathbf{P}$ is given by

$$\ominus_{lc}\mathbf{P} = \big(- \lfloor \mathscr{L}(\mathbf{P}) \rfloor + \mathbb{D}(\mathscr{L}(\mathbf{P}))^{-1} \big) \big(- \lfloor \mathscr{L}(\mathbf{P}) \rfloor + \mathbb{D}(\mathscr{L}(\mathbf{P}))^{-1} \big)^T.$$

Lemma 6. *For $\mathbf{P} \in \mathrm{Sym}_n^+$, $t \in \mathbb{R}$, the scalar multiplication $t \otimes_{lc} \mathbf{P}$ is given as*

$$t \otimes_{lc} \mathbf{P} = \big(t\lfloor \mathscr{L}(\mathbf{P}) \rfloor + \mathbb{D}(\mathscr{L}(\mathbf{P}))^t \big) \big(t\lfloor \mathscr{L}(\mathbf{P}) \rfloor + \mathbb{D}(\mathscr{L}(\mathbf{P}))^t \big)^T. \tag{9}$$

Proof. See supplementary material.

Definition 8 (Log-Cholesky Gyrovector Spaces). *Define a binary operation $\oplus_{lc}$ and a scalar multiplication $\otimes_{lc}$ by Eqs. (8) and (9), respectively. Define a gyroautomorphism generated by $\mathbf{P}$ and $\mathbf{Q}$ as*

$$\mathrm{gyr}_{lc}[\mathbf{P}, \mathbf{Q}] = \mathrm{Id}\,.$$

Theorem 3. *Gyrogroups $(\mathrm{Sym}_n^+, \oplus_{lc})$ with the scalar multiplication $\otimes_{lc}$ form gyrovector spaces $(\mathrm{Sym}_n^+, \oplus_{lc}, \otimes_{lc})$.*

Proof. See supplementary material.

3.4 Parallel Transport in Gyrovector Spaces of SPD Matrices

We now show a hidden analogy between Euclidean spaces and gyrovector spaces of SPD matrices studied in the previous sections. In the following, we drop the subscripts in the notations of the basic operations and the gyroautomorphism in these gyrovector spaces for simplicity of notation.

Lemma 7. *Let $\mathbf{P}_0, \mathbf{P}_1, \mathbf{Q}_0, \mathbf{Q}_1 \in \mathrm{Sym}_n^+$. If $\mathrm{Log}_{\mathbf{P}_1}(\mathbf{Q}_1)$ is the parallel transport of $\mathrm{Log}_{\mathbf{P}_0}(\mathbf{Q}_0)$ from $\mathbf{P}_0$ to $\mathbf{P}_1$ along geodesics connecting $\mathbf{P}_0$ and $\mathbf{P}_1$, then*

$$\ominus\mathbf{P}_1 \oplus \mathbf{Q}_1 = \mathrm{gyr}[\mathbf{P}_1, \ominus\mathbf{P}_0](\ominus\mathbf{P}_0 \oplus \mathbf{Q}_0).$$

Proof. See supplementary material.

Lemma 7 reveals a strong link between the Affine-invariant, Log-Euclidean, and Log-Cholesky geometries of SPD manifolds and hyperbolic geometry, as the algebraic definition [44] of parallel transport in a gyrovector space agrees with the classical parallel transport of differential geometry. In the gyrolanguage [44,45,46], Lemma 7 states that the gyrovector $\ominus\mathbf{P}_1 \oplus \mathbf{Q}_1$ is the gyrovector $\ominus\mathbf{P}_0 \oplus \mathbf{Q}_0$ gyrated by a gyroautomorphism. This gives a characterization of the parallel transport that is fully analogous to that of the parallel transport in Euclidean and hyperbolic spaces. Note that this characterization also agrees with the reinterpretation of addition and subtraction in a Riemannian manifold using logarithmic and exponential maps [36]. In the case of Log-Euclidean geometry, like the conclusion of Theorem 2, the result of Lemma 7 agrees with [2] which shows that the space of SPD matrices with the Log-Euclidean geometry has a vector space structure.

3.5 Low-rank SPSD Manifolds

In this section, we extend the basic operations proposed in Sections 3.1, 3.2, and 3.3 to the case of low-rank SPSD manifolds. We will see that defining a binary operation and a scalar multiplication on these manifolds that verify the axioms of gyrovector spaces is not trivial. However, based on the basic operations in gyrovector spaces of SPD matrices, one can still define the loose versions of these operations that are useful for applications.

We adopt the quotient manifold representation of $\mathrm{S}^+_{n,p}$ proposed by [3,4], i.e.,

$$\mathrm{S}^+_{n,p} \cong (\mathrm{V}_{n,p} \times \mathrm{Sym}^+_p) / \mathrm{O}(p).$$

This representation is based on the decomposition

$$\mathbf{P} = \mathbf{U}_P \overline{\mathbf{P}} \mathbf{U}_P^T \tag{10}$$

of any matrix $\mathbf{P} \in \mathrm{S}^+_{n,p}$, where $\mathbf{U}_P \in \mathrm{V}_{n,p}$ and $\overline{\mathbf{P}} \in \mathrm{Sym}^+_p$. Here, each element of $\mathrm{S}^+_{n,p}$ can be seen as a flat p-dimensional ellipsoid in $\mathbb{R}^n$ [3]. The flat ellipsoid belongs to a p-dimensional subspace spanned by the columns of $\mathbf{U}_P$, while the $p \times p$ SPD matrix $\overline{\mathbf{P}}$ defines the shape of the ellipsoid in Sym^+_p. This suggests a natural adaptation of the binary operations and scalar multiplications in gyrovector spaces of SPD matrices to the case of low-rank SPSD manifolds. That is, for any $\mathbf{P}, \mathbf{Q} \in \mathrm{S}^+_{n,p}$, the binary operation will operate on the SPD matrices that define the shapes of $\mathbf{P}$ and $\mathbf{Q}$. Note, however, that the decomposition (10) is not unique and defined up to an orthogonal transformation

$$\mathbf{U}_P \mapsto \mathbf{U}_P \mathbf{O}, \overline{\mathbf{P}} \mapsto \mathbf{O}^T \overline{\mathbf{P}} \mathbf{O},$$

where $\mathbf{O} \in \mathrm{O}_n$. In other words, $(\mathbf{U}_P \mathbf{O}, \mathbf{O}^T \overline{\mathbf{P}} \mathbf{O})$ also forms a decomposition of $\mathbf{P}$ for arbitrary $\mathbf{O} \in \mathrm{O}_n$. This is problematic since in general, the binary operations defined in the previous sections is not invariant to orthogonal transformations. To resolve this ambiguity, we resort to canonical representation [3,4,11]. Our key idea is to identify a common subspace and then rotate the ranges of $\mathbf{P}$ and $\mathbf{Q}$ to this subspace. The SPD matrices that define the shapes of $\mathbf{P}$ and $\mathbf{Q}$ are rotated accordingly based on the corresponding rotations in order to reflect the changes of the ranges of $\mathbf{P}$ and $\mathbf{Q}$. These rotations are determined as follows. For any $\mathbf{U}, \mathbf{V} \in \mathrm{Gr}_{n,p}$, among all geodesics joining $\mathbf{U}$ and $\mathbf{V}$, we consider the one joining their canonical bases. Let $\mathbf{U}_V$ be the canonical base in the orbit of $\mathbf{U}$, $\mathbf{V}_U$ be the canonical base in the orbit of $\mathbf{V}$. These bases are given [3,4,11] by

$$\mathbf{U}_V = \mathbf{U}\mathbf{O}_{\mathbf{U}\to\mathbf{V}}, \mathbf{V}_U = \mathbf{V}\mathbf{O}_{\mathbf{V}\to\mathbf{U}},$$

where $\mathbf{O}_{\mathbf{U}\to\mathbf{V}}$ and $\mathbf{O}_{\mathbf{V}\to\mathbf{U}}$ are orthogonal matrices computed from a singular value decomposition (SVD) of $\mathbf{U}^T\mathbf{V}$, i.e.,

$$\mathbf{U}^T\mathbf{V} = \mathbf{O}_{\mathbf{U}\to\mathbf{V}} \boldsymbol{\Sigma} \mathbf{O}^T_{\mathbf{V}\to\mathbf{U}},$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix whose diagonal entries are the singular values of $\mathbf{U}^T\mathbf{V}$.

Let $\mathbf{U}_e$ be a common subspace used to define the loose versions of the basic operations on $\mathrm{S}^+_{n,p}$. In a special case where all matrices are supported by the same subspace, then $\mathbf{U}_e$ can be identified as this subspace. By abuse of language, we will use the same terminologies in gyrovector spaces to refer to the loose versions in the following definitions.

Definition 9. *Let* $\mathbf{P}, \mathbf{Q} \in \mathrm{S}^+_{n,p}$, $\mathbf{P} = \mathbf{U}_P \overline{\mathbf{P}} \mathbf{U}_P^T$, *and* $\mathbf{Q} = \mathbf{U}_Q \overline{\mathbf{Q}} \mathbf{U}_Q^T$. *The binary operation* $\mathbf{P} \oplus_{spsd} \mathbf{Q}$ *is defined as*

$$\mathbf{P} \oplus_{spsd} \mathbf{Q} = \mathbf{U}_e (\mathbf{O}^T_{\mathbf{U}_P \to \mathbf{U}_e} \overline{\mathbf{P}} \mathbf{O}_{\mathbf{U}_P \to \mathbf{U}_e} \oplus \mathbf{O}^T_{\mathbf{U}_Q \to \mathbf{U}_e} \overline{\mathbf{Q}} \mathbf{O}_{\mathbf{U}_Q \to \mathbf{U}_e}) \mathbf{U}_e^T.$$

Let $\mathbf{I}_{n,p} = \begin{bmatrix} \mathbf{I}_p & 0 \\ 0 & 0 \end{bmatrix}$ be the identity element of $\mathrm{S}^+_{n,p}$. The inverse of $\mathbf{P} \in \mathrm{S}^+_{n,p}$ where $\mathbf{P} = \mathbf{U}_P \overline{\mathbf{P}} \mathbf{U}_P^T$ is defined as

$$\ominus_{spsd} \mathbf{P} = \mathbf{U}_e \big(\ominus (\mathbf{O}^T_{\mathbf{U}_P \to \mathbf{U}_e} \overline{\mathbf{P}} \mathbf{O}_{\mathbf{U}_P \to \mathbf{U}_e}) \big) \mathbf{U}_e^T.$$

Definition 10. *Let* $\mathbf{P} \in \mathrm{S}^+_{n,p}$, $t \in \mathbb{R}$, *and* $\mathbf{P} = \mathbf{U}_P \overline{\mathbf{P}} \mathbf{U}_P^T$. *The scalar multiplication* $t \otimes_{spsd} \mathbf{P}$ *is defined as*

$$t \otimes_{spsd} \mathbf{P} = \mathbf{U}_e (t \otimes \mathbf{O}^T_{\mathbf{U}_P \to \mathbf{U}_e} \overline{\mathbf{P}} \mathbf{O}_{\mathbf{U}_P \to \mathbf{U}_e}) \mathbf{U}_e^T.$$

Definition 11. *Let* $\mathbf{P}, \mathbf{Q}, \mathbf{R} \in \mathrm{S}^+_{n,p}$, $\mathbf{P} = \mathbf{U}_P \overline{\mathbf{P}} \mathbf{U}_P^T$, $\mathbf{Q} = \mathbf{U}_Q \overline{\mathbf{Q}} \mathbf{U}_Q^T$, *and* $\mathbf{R} = \mathbf{U}_R \overline{\mathbf{R}} \mathbf{U}_R^T$. *The gyroautomorphism in* $\mathrm{S}^+_{n,p}$ *is defined by*

$$\begin{aligned} &\mathrm{gyr}_{spsd}[\mathbf{P}, \mathbf{Q}]\mathbf{R} \\ &= \mathbf{U}_e \big(\mathrm{gyr}[\mathbf{O}^T_{\mathbf{U}_P \to \mathbf{U}_e} \overline{\mathbf{P}} \mathbf{O}_{\mathbf{U}_P \to \mathbf{U}_e}, \mathbf{O}^T_{\mathbf{U}_Q \to \mathbf{U}_e} \overline{\mathbf{Q}} \mathbf{O}_{\mathbf{U}_Q \to \mathbf{U}_e}] \mathbf{O}^T_{\mathbf{U}_R \to \mathbf{U}_e} \overline{\mathbf{R}} \mathbf{O}_{\mathbf{U}_R \to \mathbf{U}_e} \big) \mathbf{U}_e^T. \end{aligned}$$

It can be seen that spaces $\mathrm{S}^+_{n,p}$, when equipped with the basic operations defined in this section, do not form gyrovector spaces. However, Theorem 4 states that they still verify some important properties of gyrovector spaces.

Theorem 4. *The basic operations on* $\mathrm{S}^+_{n,p}$ *verify the Left Gyroassociative Law, Left Reduction Property, Gyrocommutative Law, and axioms (V2), (V3), and (V4).*

Proof. See supplementary material.

3.6 SPSD Neural Networks

Motivated by the work of [12] that develops RNNs on hyperbolic spaces, in this section, we propose a class of RNNs on SPSD manifolds. It is worth mentioning that the operations defined in Sections 3.1, 3.2, and 3.3 as well as those constructed below can be used to build other types of neural networks on SPSD manifolds, e.g., convolutional neural networks. We leave this for future work.

In addition to the basic operations, we need to generalize some other operations of Euclidean RNNs to the SPSD manifold setting. Here we focus on two operations, i.e., vector-matrix multiplication and pointwise nonlinearity. Other operations [12] are left for future work.

Vector-matrix multiplication. If $\mathbf{P} \in \mathrm{S}^+_{n,p}$ and $\mathbf{W} \in \mathbb{R}^p, \mathbf{W} \geq 0$, then the vector-matrix multiplication $\mathbf{W} \otimes_v \mathbf{P}$ is given by

$$\mathbf{W} \otimes_v \mathbf{P} = \mathbf{U} \operatorname{diag}(\mathbf{W} * \mathbf{V})\mathbf{U}^T,$$

where $\mathbf{P} = \mathbf{U} \operatorname{diag}(\mathbf{V})\mathbf{U}^T$ is the eigenvalue decomposition of $\mathbf{P}$, and $*$ denotes the element-wise multiplication.

Pointwise nonlinearity. If φ is a pointwise nonlinear activation function, then the pointwise nonlinearity $\varphi^{\otimes_a}(\mathbf{P})$ is given by

$$\varphi^{\otimes_a}(\mathbf{P}) = \mathbf{U} \operatorname{diag}(\max(\epsilon \mathbf{I}, \varphi(\mathbf{V})))\mathbf{U}^T,$$

where $\epsilon > 0$ is a rectification threshold, and $\mathbf{P} = \mathbf{U} \operatorname{diag}(\mathbf{V})\mathbf{U}^T$ is the eigenvalue decomposition of $\mathbf{P}$. Note that the above operations preserve the range and the positive semi-definite property.

We first consider the case of full-rank learning. In this case, we adapt a class of models that are invariant to time rescaling [43] to the SPD manifold setting using the gyro-chain-rule in gyrovector spaces of SPD matrices (see supplementary material for the derivation). We obtain the following update equations:

$$\mathbf{P}_t = \varphi^{\otimes_a}(\mathbf{W}_h \otimes_v \mathbf{H}_{t-1} + \mathbf{W}_x \otimes_v \phi(\mathbf{X}_t)), \tag{11}$$

$$\mathbf{H}_t = \mathbf{H}_{t-1} \oplus \alpha \otimes ((\ominus \mathbf{H}_{t-1}) \oplus \mathbf{P}_t), \tag{12}$$

where $\mathbf{X}_t \in \mathrm{Sym}^+_n$ is the input at frame t, $\mathbf{H}_{t-1}, \mathbf{H}_t \in \mathrm{Sym}^+_n$ are the hidden states at frames $t-1$ and t, respectively, $\mathbf{W}_h, \mathbf{W}_x \in \mathbb{R}^n$, $\mathbf{W}_h, \mathbf{W}_x > 0$, and $\alpha \in \mathbb{R}$ are learnable parameters, $\phi(\mathbf{X}_t)$ is any mapping[2] that transforms $\mathbf{X}_t$ into a SPD matrix.

The above model can be extended to the low-rank case by viewing $\mathbf{H}_{t-1}$ and $\mathbf{H}_t$ as the SPD matrices that define the shapes of the SPSD matrices representing the hidden states at frames $t-1$ and t, respectively. If we assume that the hidden states are supported by the same subspace, then it is tempting to design the mapping $\phi(\mathbf{X}_t)$ such that it gives the SPD matrix that defines the shape of $\mathbf{X}_t$, once the range of $\mathbf{X}_t$ is rotated to that subspace. The algorithm in both the full-rank and low-rank cases can be summarized in Algorithm 1. The procedure SpdRotate() performs the computations of the mapping $\phi(.)$, and $\mathbf{X}^u[:, : p]$ denotes the matrix containing the first p columns of $\mathbf{X}^u$. The parameter $\mathbf{W}$ can be either learned or fixed up front which offers flexible learning strategies.

Complexity Analysis. In the full-rank learning setting, the most expensive operations are performed at steps 1, 3, 4, and 5, each of them has $O(n^3)$ complexity. In the low-rank learning setting, the most expensive operation is the eigenvalue decomposition at step 1 that has $O(n^3)$ complexity, while steps 3, 4, and 5 are $O(p^3)$ operations. Thus, when $p \ll n$, the algorithm in the low-rank learning setting is faster than the one in the full-rank learning setting.

[2] We drop parameters to simplify notations.

Algorithm 1: SPSD-RNN

Data: $\mathbf{X}_t \in \mathrm{S}^+_{n,p}, \mathbf{H}^{spd}_{t-1} \in \mathrm{Sym}^+_p, \mathbf{W} \in \mathrm{V}_{n,p}$

Result: $\mathbf{H}^{spd}_t \in \mathrm{Sym}^+_p$

$\mathbf{X}^s, \mathbf{X}^u \leftarrow \mathrm{EIG}(\mathbf{X}_t)$; /* $\mathbf{X}^s$: eigenvalues, $\mathbf{X}^u$: eigenvectors */

$\mathbf{X}^u \leftarrow \mathbf{X}^u[:, : p]$; /* Eigenvalues are arranged in descending order */

$\mathbf{X}^{spd} \leftarrow \mathrm{SpdRotate}(\mathbf{X}_t, \mathbf{X}^u, \mathbf{W})$;

$\mathbf{P}_t = \varphi^{\otimes_a}(\mathbf{W}_h \otimes_v \mathbf{H}^{spd}_{t-1} + \mathbf{W}_x \otimes_v \mathbf{X}^{spd})$;

$\mathbf{H}^{spd}_t = \mathbf{H}^{spd}_{t-1} \oplus \alpha \otimes ((\ominus \mathbf{H}^{spd}_{t-1}) \oplus \mathbf{P}_t)$;

4 Experiments

Our networks, referred to as **SPSD-AI**, **SPSD-LE**, and **SPSD-LC** were implemented with Tensorflow framework. The networks were trained using Adadelta optimizer for 500 epochs. The ReLU function was used for the pointwise nonlinearity. The learning rate and parameter ϵ for the pointwise nonlinearity were set respectively to 10^{-2} and 10^{-4}.

4.1 Datasets and Experimental Settings

We use HDM05 [31], FPHA [13], and NTU RBG+D 60 (NTU60) [39] datasets. These datasets include three different types of human activities: body actions (HDM05), hand actions (FPHA), and interaction actions (NTU60).

HDM05 dataset. It has 2337 sequences of 3D skeleton data classified into 130 classes. Each frame contains the 3D coordinates of 31 body joints. We use all the action classes and follow the experimental protocol [15] in which 2 subjects are used for training and the remaining 3 subjects are used for testing.

FPHA dataset. It has 1175 sequences of 3D skeleton data classified into 45 classes. Each frame contains the 3D coordinates of 21 hand joints. We follow the experimental protocol [13] in which 600 sequences are used for training and 575 sequences are used for testing.

NTU60 dataset. It has 56880 sequences of 3D skeleton data classified into 60 classes. Each frame contains the 3D coordinates of 25 or 50 body joints. We use the mutual (interaction) actions and follow the cross-subject (X-Sub) and cross-view (X-View) experimental protocols [39]. For the X-Sub protocol, this results in 7319 and 3028 sequences for training and testing, respectively. For the X-View protocol, the numbers of training and testing sequences are 6889 and 3458, respectively.

4.2 Implementation Details

In order to retain the correlation of neighboring joints [5,49] and to increase feature interactions encoded by covariance matrices, we first identify a closest left (right) neighbor of every joint based on their distance to the hip (wrist)

Dataset	HDM05		FPHA		NTU60 (X-Sub)		NTU60 (X-View)	
	FR	LR	FR	LR	FR	LR	FR	LR
Accuracy (%)	81.32	74.19	96.58	90.94	95.86	89.78	97.44	90.15
Training time (min)	1.09	0.66	0.64	0.39	9.19	5.61	8.56	5.26

Table 1. Accuracies and training times (minutes) per epoch of SPSD-AI for the full-rank (FR) and low-rank (LR) learning settings (the experiment is conducted on 1 NVIDIA 1080 GPU).

Dataset	HDM05		FPHA		NTU60 (X-Sub)		NTU60 (X-View)	
	C	NC	C	NC	C	NC	C	NC
Accuracy (%)	74.19	41.37	90.94	63.41	89.78	56.82	90.15	58.34

Table 2. Accuracies of SPSD-AI when canonical representation is used (C) and not used (NC).

joint[3], and then combine the 3D coordinates of each joint and those of its left (right) neighbor to create a feature vector for the joint. Thus, a 6-dim feature vector is created for every joint. The input SPD data are computed as follows. For any frame t, a mean vector $\boldsymbol{\mu}_t$ and a covariance matrix $\boldsymbol{\Sigma}_t$ are computed from the set of feature vectors of the frame and then combined [29] to create a SPD matrix as

$$\mathbf{Y}_t = \begin{bmatrix} \boldsymbol{\Sigma}_t + \boldsymbol{\mu}_t(\boldsymbol{\mu}_t)^T & \boldsymbol{\mu}_t \\ (\boldsymbol{\mu}_t)^T & 1 \end{bmatrix}.$$

The lower part of matrix $\log(\mathbf{Y}_t)$ is flattened to obtain a vector $\tilde{\mathbf{v}}_t$. All vectors $\tilde{\mathbf{v}}_t$ within a time window $[t, t+c-1]$ where c is a predefined value ($c = 10$ in our experiments) are used to compute a covariance matrix as $\mathbf{Z}_t = \frac{1}{c}\sum_{i=t}^{t+c-1}(\tilde{\mathbf{v}}_i - \bar{\mathbf{v}}_t)(\tilde{\mathbf{v}}_i - \bar{\mathbf{v}}_t)^T$, where $\bar{\mathbf{v}}_t = \frac{1}{c}\sum_{i=t}^{t+c-1}\tilde{\mathbf{v}}_i$. Matrix $\mathbf{Z}_t$ is then the input data at frame t of the networks.

For classification, the network output is projected to the tangent space at the identity matrix using the logarithmic map. The lower part of the resulting matrix is flattened and fed to a fully-connected layer. Cross-entropy loss is used to optimize the network. Please refer to the supplementary material for more implementation details.

4.3 Ablation Study

In this section, we discuss the impact of different components of SPSD-AI on its performance.

Full-rank vs. low-rank learning. In this experiment, we compare the accuracies and computation times of SPSD-AI in the full-rank and low-rank learning

[3] For joints having more than one left (right) neighbor, one of them can be chosen.

Method	Accuracy (%)	#Param. (MB)	Year
SPD-SRU [8]	42.26	0.05	2018
SPDNet [17]	58.44	0.12	2017
SPDNetBN [6]	62.54	0.13	2019
HypGRU [12]	55.47	0.23	2018
MS-G3D [27]	70.38	2.93	2020
ST-TR [38]	69.75	4.62	2021
SPSD-AI	**81.32**	0.31	
SPSD-LE	77.46	0.31	
SPSD-LC	73.52	0.31	

Table 3. Accuracies of our networks and state-of-the-art methods on HDM05 dataset.

settings. The value of p is set to 14. Results are reported in Tab. 1 (see supplementary material for more results w.r.t different settings of p). In terms of accuracy, the network trained in the full-rank learning setting overpasses the one trained in the low-rank learning setting. In terms of computation time, however, the former is slower than the latter. This experiment highlights the advantage of each of the learning settings. In practice, the value of p can be adjusted to offer a good compromise between accuracy and computation time.

Canonical representation-based learning. This experiment is conducted to illustrate the efficacy of canonical representation in the low-rank case. For a fair comparison, we evaluate the accuracies of SPSD-AI in the two following settings. In the first setting, canonical representation is used and the parameter $\mathbf{W}$ is fixed up front ($\mathbf{W} = \mathbf{I}_{n,p}$). In the second setting, canonical representation is not used, i.e., $\mathbf{X}^{spd}$ at line 3 of Algorithm 1 is computed as

$$\mathbf{X}^{spd} = (\mathbf{X}^u)^T \operatorname{diag}(\mathbf{X}^s[: p])\mathbf{X}^u, \tag{13}$$

where $\mathbf{X}^s[: p]$ denotes the vector containing the first p elements of $\mathbf{X}^s$, and diag(.) forms a diagonal matrix from a vector. The value of p is set to 14. Results are presented in Tab. 2. In all cases, canonical representation yields significant improvements in accuracy. The average performance gain over all the datasets is more than 30%.

In the following, we report results of our networks in the full-rank case.

4.4 Results on HDM05 Dataset

Results of our networks and state-of-the-art methods on HDM05 dataset are presented in Tab. 3. The implementations of SPD-SRU[4], SPDNet[5], SPDNetBN[6],

[4] https://github.com/zhenxingjian/SPD-SRU/tree/master
[5] https://github.com/zhiwu-huang/SPDNet
[6] https://papers.nips.cc/paper/2019/hash/6e69ebbfad976d4637bb4b39de261bf7-Abstract.html

Method	Accuracy (%)	#Param. (MB)	Year
SPD-SRU [8]	78.57	0.02	2018
SPDNet [17]	87.65	0.04	2017
SPDNetBN [6]	88.52	0.05	2019
HypGRU [12]	58.61	0.16	2018
ST-TS-HGR-NET [33]	93.22	-	2019
SRU-HOS-NET [34]	94.61	-	2020
HMM-HPEV-Net [25]	90.96	-	2020
SAGCN-RBi-IndRNN [22]	90.26	-	2021
MS-G3D [27]	88.61	2.90	2020
ST-TR [38]	86.32	4.59	2021
SPSD-AI	**96.58**	0.11	
SPSD-LE	91.84	0.11	
SPSD-LC	89.73	0.11	

Table 4. Accuracies of our networks and state-of-the-art methods on FPHA dataset.

HypGRU[7], MS-G3D[8], and ST-TR[9] are rendered publicly available by the authors of [8], [17], [6], [12], [27], and [38] respectively. HypGRU achieves the best results when the data are projected to hyperbolic spaces before they are fed to the networks, and all its layers are based on hyperbolic geometry. The hidden dimension for HypGRU is set to 200. For SPDNet and SPDNetBN, we compute a covariance matrix to represent an input sequence as in [17]. The sizes of the covariance matrix are 93×93. Our networks outperform all the competing networks. In particular, they beat the state-of-the-art MS-G3D for 3D skeleton-based action recognition. Furthermore, our networks have about 9 times fewer parameters than MS-G3D. Our networks also have superior performance than the SPD and HNN models.

4.5 Results on FPHA Dataset

For SPDNet and SPDNetBN, the sizes of the input covariance matrix are 60×60, and the sizes of the transformation matrices are set to 60×50, 50×40, and 40×30. Results of our networks and state-of-the-art methods on FPHA dataset are given in Tab. 4. SPSD-AI is the best performer on this dataset. SPSD-LE gives the second best result among our networks. It is slightly better than HMM-HPEV-Net and SAGCN-RBi-IndRNN, and outperforms MS-G3D and ST-TR by more than 3%. SPSD-LC achieves the lowest accuracy among our networks. However, it is superior to HypGRU and compares favorably to some state-of-the-art SPD models, i.e., SPD-SRU, SPDNet, and SPDNetBN. Note that MS-G3D is outperformed by our networks despite having 26 times more parameters.

[7] https://github.com/dalab/hyperbolic_nn
[8] https://github.com/kenziyuliu/MS-G3D
[9] https://github.com/Chiaraplizz/ST-TR

Method	X-Sub	X-View	#Param. (X-Sub,MB)	Year
ST-LSTM [26]	83.0	87.3	-	2016
ST-GCN [48]	87.05	91.02	-	2018
AS-GCN [23]	91.22	93.46	-	2019
LSTM-IRN [37]	90.5	93.5	-	2019
SPD-SRU [8]	66.25	68.11	0.004	2018
SPDNet [17]	73.26	74.82	0.03	2017
SPDNetBN [6]	75.84	76.96	0.04	2019
HypGRU [12]	88.26	89.43	0.013	2018
MS-G3D [27]	93.25	95.73	2.89	2020
ST-TR [38]	92.18	94.69	4.58	2021
2S-DRAGCN$_{joint}$ [50]	93.20	95.58	3.57	2021
GeomNet [32]	93.62	96.32	-	2021
SPSD-AI	**95.86**	**97.44**	0.03	
SPSD-LE	90.74	91.18	0.03	
SPSD-LC	88.03	88.56	0.03	

Table 5. Accuracies (%) of our networks and state-of-the-art methods on NTU60 dataset (interaction actions).

4.6 Results on NTU60 Dataset

For SPDNet and SPDNetBN, the sizes of the input covariance matrix are 150×150, and the sizes of the transformation matrices are set to 150×100, 100×60, and 60×30. Results of our networks and state-of-the-art methods on NTU60 dataset are reported in Tab. 5. Again, SPSD-AI gives the best results among the competing methods. It outperforms the state-of-the-art methods 2S-DRAGCN and GeomNet for 3D skeleton-based human interaction recognition. We observe that on all the datasets, SPSD-AI trained in the low-rank learning setting (see Tab. 1) competes favorably with SPD-SRU, SPDNet, and SPDNetBN. This suggests that integrating our method for low-rank matrix learning into these networks might also lead to effective solutions for SPSD matrix learning.

5 Conclusion

We presented a method for constructing the basic operations in gyrovector spaces of SPD matrices. We then studied gyrovector spaces of SPD matrices with the Affine-invariant, Log-Euclidean, and Log-Cholesky geometries. We proposed the loose versions of the basic operations in low-rank SPSD manifolds by extending those previously defined in gyrovector spaces of SPD matrices. Finally, we developed a class of RNNs for SPSD matrix learning, and provided the experimental evaluations on three benchmarks for human activity recognition to demonstrate the effectiveness of our networks.

References

1. Abe, T., Hatori, O.: Generalized Gyrovector Spaces and a MazurUlam Theorem. Publicationes Mathematicae Debrecen **87**, 393–413 (2015) 2
2. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Fast and Simple Computations on Tensors with Log-Euclidean Metrics. Tech. Rep. RR-5584, INRIA (2005) 1, 2, 5, 6
3. Bonnabel, S., Collard, A., Sepulchre, R.: Rank-preserving Geometric Means of Positive Semi-definite Matrices. Linear Algebra and its Applications **438**, 3202–3216 (2013) 1, 2, 7
4. Bonnabel, S., Sepulchre, R.: Riemannian Metric and Geometric Mean for Positive Semidefinite Matrices of Fixed Rank. SIAM Journal on Matrix Analysis and Applications **31**(3), 1055–1070 (2009) 1, 2, 7
5. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning Mid-level Features for Recognition. In: CVPR. pp. 2559–2566 (2010) 10
6. Brooks, D.A., Schwander, O., Barbaresco, F., Schneider, J.Y., Cord, M.: Riemannian Batch Normalization for SPD Neural Networks. In: NeurIPS. pp. 15463–15474 (2019) 1, 12, 13, 14
7. Chakraborty, R., Bouza, J., Manton, J., Vemuri, B.C.: ManifoldNet: A Deep Neural Network for Manifold-valued Data with Applications. TPAMI pp. 1–1 (2020) 1
8. Chakraborty, R., Yang, C.H., Zhen, X., Banerjee, M., Archer, D., Vaillancourt, D.E., Singh, V., Vemuri, B.C.: A Statistical Recurrent Model on the Manifold of Symmetric Positive Definite Matrices. In: NeurIPS. pp. 8897–8908 (2018) 1, 2, 12, 13, 14
9. Dong, Z., Jia, S., Zhang, C., Pei, M., Wu, Y.: Deep Manifold Learning of Symmetric Positive Definite Matrices with Application to Face Recognition. In: AAAI. pp. 4009–4015 (2017) 1
10. Engin, M., Wang, L., Zhou, L., Liu, X.: DeepKSPD: Learning Kernel-matrix-based SPD Representation for Fine-grained Image Recognition. In: ECCV. pp. 629–645 (2018) 1
11. Gallivan, K., Srivastava, A., Liu, X., Van Dooren, P.: Efficient Algorithms for Inferences on Grassmann Manifolds. In: IEEE Workshop on Statistical Signal Processing. pp. 315–318 (2003) 7
12. Ganea, O., Becigneul, G., Hofmann, T.: Hyperbolic neural networks. In: NeurIPS. pp. 5350–5360 (2018) 2, 8, 9, 12, 13, 14
13. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations. In: CVPR. pp. 409–419 (2018) 10
14. Harandi, M., Salzmann, M., Hartley, R.: Dimensionality Reduction on SPD Manifolds: The Emergence of Geometry-Aware Methods. TPAMI **40**, 48–62 (2018) 1
15. Harandi, M.T., Salzmann, M., Hartley, R.: From Manifold to Manifold: Geometry-Aware Dimensionality Reduction for SPD Matrices. In: ECCV. pp. 17–32 (2014) 1, 10
16. Hatori, O.: Examples and Applications of Generalized Gyrovector Spaces. Results in Mathematics **71**, 295–317 (2017) 2
17. Huang, Z., Gool, L.V.: A Riemannian Network for SPD Matrix Learning. In: AAAI. pp. 2036–2042 (2017) 1, 12, 13, 14
18. Kim, S.: Distributivity on the Gyrovector Spaces. Kyungpook Mathematical Journal **55**, 13–20 (2015) 2, 3

19. Kim, S.: Gyrovector Spaces on the Open Convex Cone of Positive Definite Matrices. Mathematics Interdisciplinary Research **1**(1), 173–185 (2016) 2, 3
20. Kim, S.: Operator Inequalities and Gyrolines of the Weighted Geometric Means. CoRR **abs/2009.10274** (2020) 2, 3
21. Kim, S.: Ordered Gyrovector Spaces. Symmetry **12**(6) (2020) 2, 3
22. Li, C., Li, S., Gao, Y., Zhang, X., Li, W.: A Two-stream Neural Network for Pose-based Hand Gesture Recognition. IEEE Transactions on Cognitive and Developmental Systems pp. 1–1 (2021) 13
23. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. In: CVPR. pp. 3595–3603 (2019) 14
24. Lin, Z.: Riemannian Geometry of Symmetric Positive Definite Matrices via Cholesky Decomposition. SIAM Journal on Matrix Analysis and Applications **40**(4), 1353–1370 (2019) 1, 2, 5
25. Liu, J., Liu, Y., Wang, Y., Prinet, V., Xiang, S., Pan, C.: Decoupled Representation Learning for Skeleton-Based Gesture Recognition. In: CVPR. pp. 5750–5759 (2020) 13
26. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. In: ECCV. pp. 816–833 (2016) 14
27. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. In: CVPR. pp. 143–152 (2020) 12, 13, 14
28. López, F., Pozzetti, B., Trettel, S., Strube, M., Wienhard, A.: Vector-valued Distance and Gyrocalculus on the Space of Symmetric Positive Definite Matrices. In: NeurIPS. pp. 18350–18366 (2021) 2
29. Lovrić, M., Min-Oo, M., Ruh, E.A.: Multivariate Normal Distributions Parametrized As a Riemannian Symmetric Space. Journal of Multivariate Analysis **74**(1), 36–48 (2000) 11
30. Massart, E., Absil, P.A.: Quotient Geometry with Simple Geodesics for the Manifold of Fixed-Rank Positive-Semidefinite Matrices. SIAM Journal on Matrix Analysis and Applications **41**(1), 171–198 (2020) 1
31. Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., Weber, A.: Documentation Mocap Database HDM05. Tech. Rep. CG-2007-2, Universität Bonn (June 2007) 10
32. Nguyen, X.S.: GeomNet: A Neural Network Based on Riemannian Geometries of SPD Matrix Space and Cholesky Space for 3D Skeleton-Based Interaction Recognition. In: ICCV. pp. 13379–13389 (2021) 1, 14
33. Nguyen, X.S., Brun, L., Lézoray, O., Bougleux, S.: A Neural Network Based on SPD Manifold Learning for Skeleton-based Hand Gesture Recognition. In: CVPR. pp. 12036–12045 (2019) 13
34. Nguyen, X.S., Brun, L., Lézoray, O., Bougleux, S.: Learning Recurrent High-order Statistics for Skeleton-based Hand Gesture Recognition. In: ICPR. pp. 975–982 (2020) 1, 13
35. Pennec, X.: Statistical Computing on Manifolds for Computational Anatomy. Habilitation à diriger des recherches, Université Nice Sophia-Antipolis (2006) 1
36. Pennec, X., Fillard, P., Ayache, N.: A Riemannian Framework for Tensor Computing. Tech. Rep. RR-5255, INRIA (2004) 1, 2, 6
37. Perez, M., Liu, J., Kot, A.C.: Interaction Relational Network for Mutual Action Recognition. CoRR **abs/1910.04963** (2019) 14

38. Plizzari, C., Cannici, M., Matteucci, M.: Skeleton-based Action Recognition via Spatial and Temporal Transformer Networks. Computer Vision and Image Understanding **208**, 103219 (2021) 12, 13, 14
39. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In: CVPR. pp. 1010–1019 (2016) 10
40. Shimizu, R., Mukuta, Y., Harada, T.: Hyperbolic Neural Networks++. In: ICLR (2021) 2
41. Sukthanker, R.S., Huang, Z., Kumar, S., Endsjo, E.G., Wu, Y., Gool, L.V.: Neural Architecture Search of SPD Manifold Networks. In: IJCAI. pp. 3002–3009 (2021) 1
42. Tabia, H., Laga, H., Picard, D., Gosselin, P.H.: Covariance Descriptors for 3D Shape Matching and Retrieval. In: CVPR. pp. 4185–4192 (2014) 1
43. Tallec, C., Ollivier, Y.: Can Recurrent Neural Networks Warp Time? In: ICLR (2018) 9
44. Ungar, A.A.: Beyond the Einstein Addition Law and Its Gyroscopic Thomas Precession: The Theory of Gyrogroups and Gyrovector Spaces. Fundamental Theories of Physics, vol. 117, Springer, Netherlands (2002) 2, 6
45. Ungar, A.A.: Analytic Hyperbolic Geometry: Mathematical Foundations and Applications. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ (2005) 2, 6
46. Ungar, A.A.: Analytic Hyperbolic Geometry in N Dimensions: An Introduction. CRC Press (2014) 2, 3, 6
47. Wang, R., Wu, X.J., Kittler, J.: SymNet: A Simple Symmetric Positive Definite Manifold Deep Learning Method for Image Set Classification. IEEE Transactions on Neural Networks and Learning Systems pp. 1–15 (2021) 1
48. Yan, S., Xiong, Y., Lin, D.: Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In: AAAI. pp. 7444–7452 (2018) 14
49. Yang, X., Tian, Y.: Super Normal Vector for Activity Recognition Using Depth Sequences. In: CVPR. pp. 804–811 (2014) 10
50. Zhu, L., Wan, B., Li, C., Tian, G., Hou, Y., Yuan, K.: Dyadic Relational Graph Convolutional Networks for Skeleton-based Human Interaction Recognition. Pattern Recognition **115**, 107920 (2021) 14