Siegel Neural Networks

Xuan Son Nguyen **Aymeric Histace** **Nistor Grozavu**
ETIS, UMR 8051, CY Cergy Paris University, ENSEA, CNRS
{xuan-son.nguyen,aymeric.histace}@ensea.fr
nistor.grozavu@cyu.fr

Abstract

Riemannian symmetric spaces (RSS) such as hyperbolic spaces and symmetric positive definite (SPD) manifolds have become popular spaces for representation learning. In this paper, we propose a novel approach for building discriminative neural networks on Siegel spaces, a family of RSS that is largely unexplored in machine learning tasks. For classification applications, one focus of recent works is the construction of multiclass logistic regression (MLR) and fully-connected (FC) layers for hyperbolic and SPD neural networks. Here we show how to build such layers for Siegel neural networks. Our approach relies on the quotient structure of those spaces and the notation of vector-valued distance on RSS. We demonstrate the relevance of our approach on two applications, i.e., radar clutter classification and node classification. Our results successfully demonstrate state-of-the-art performance across all datasets.

1 Introduction

Deep neural networks are generally built upon the assumption that the data or features at hand exhibit Euclidean latent structure. Unfortunately, this assumption does not hold in many applications [1, 17] where the data or features lie on a multidimensional curved surface which is locally Euclidean. For such applications, those models often produce unsatisfactory results because their building blocks based on Euclidean geometry break the geometric stability principle that plays a crucial role in geometric deep learning architectures [8]. To deal with this issue, many Riemannian neural networks have been developed for solving a wide variety of machine learning problems [14, 17, 21, 23, 26, 36]. In this paper, we restrict our attention to discriminative neural networks with manifold-valued output.

Early works focus either on hyperbolic spaces [17, 36] or on matrix manifolds [21, 22, 23]. In an attempt to develop a unified framework for a more general setting, the authors of [30, 31, 32] leverage the gyro-structure of certain Riemannian manifolds. However, in the general case, their methods cannot provide an explicit form for the point-to-hyperplane distance which is at the heart of their proposed network building blocks [36] since the distance must be derived with respect to a specific Riemannian metric. The work in [33] alleviates this issue by deriving a closed form for the point-to-hyperplane distance associated with G-invariant Riemannian metrics on RSS. Although this work is applicable to Siegel spaces, the construction of the MLR and FC layers [33] from the derived distance is heavily based on the maximal abelian subspaces of those spaces. This can affect the ability of the resulting networks to learn rich representations and complex decision boundaries.

In this paper, we propose a novel approach for building neural networks on Siegel spaces. Those are among the most versatile RSS [28] and have many attractive theoretical properties. The two well-established models of Siegel spaces, i.e., the Siegel upper space and the Siegel disk [37] generalize the complex Poincaré upper plane and the complex Poincaré disk [19] to spaces of symmetric complex matrices [34]. SPD manifolds associated with affine-invariant Riemannian metrics [35] are also special cases of Siegel spaces associated with Siegel metrics [37]. Despite the potential of Siegel spaces in capturing rich geometrical structures, they are much less studied in the context

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

of deep learning compared to other RSS. Although few recent works [28, 39] use Siegel spaces as representation spaces, they only focus on learning and visualizing embeddings in natural language processing and graph tasks. Therefore, an effective framework for building discriminative Siegel neural networks is still missing. In summary, our contributions are the following:

- We propose a novel formulation of MLR layers for Siegel neural networks based on a gyrovector approach which has proven effective in building hyperbolic and SPD neural networks [17, 30, 36]. This formulation is then extended to the product space setting.
- We show that the notion of vector-valued distance [25] which captures the complete G-congruence invariant of pairs of points on RSS enables another formulation of MLR layers for Siegel neural networks in a natural way. This formulation leads to more compact MLR layers than those obtained by the first formulation.
- We introduce two variants of FC layers for Siegel neural networks.
- We build the first discriminative Siegel neural networks and evaluate them on the radar clutter classification and node classification tasks.

2 Mathematical Background

2.1 Siegel Spaces

The Siegel upper space $\mathbb{SH}_m$ is defined as

$$\mathbb{SH}_m = \{x = u + iv : u \in \mathrm{Sym}_m, v \in \mathrm{Sym}_m^+\},$$

where Sym_m and Sym_m^+ denote the space of $m \times m$ real symmetric matrices and that of $m \times m$ SPD matrices, respectively.

Another model for Siegel spaces is the Siegel disk defined by

$$\mathbb{SD}_m = \left\{w \in \mathrm{Sym}_{m,\mathbb{C}} : I_m - ww^H \in \mathbb{H}_m^+\right\},$$

where I_m is the $m \times m$ identity matrix, $\mathrm{Sym}_{m,\mathbb{C}}$ and $\mathbb{H}_m^+$ denote the space of $m \times m$ complex symmetric matrices and that of $m \times m$ Hermitian positive definite (HPD) matrices, respectively, and w^H is the conjugate transpose of w.

One can convert a point $x \in \mathbb{SH}_m$ to $\mathbb{SD}_m$ using the matrix Cayley transformation defined as

$$\varphi(x) = (x - iI_m)(x + iI_m)^{-1}.$$

The inverse matrix Cayley transformation that converts a point $w \in \mathbb{SD}_m$ to $\mathbb{SH}_m$ is given by

$$\varphi^{(-1)}(w) = i(I_m + w)(I_m - w)^{-1}.$$

In the following, we shall focus on the Siegel upper space model.

Quotient Structure Denote by

$$\mathrm{Sp}_{2m} = \left\{ \begin{bmatrix} a & b \\ c & d \end{bmatrix} : ab^T = ba^T, cd^T = dc^T, ad^T - bc^T = I_m \right\}$$

the real symplectic group. This group acts transitively on $\mathbb{SH}_m$ by the action $s[x] = (ax + b)(cx + d)^{-1}$, where $s = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathrm{Sp}_{2m}$ and $x \in \mathbb{SH}_m$. The stabilizer group of $x = iI_m \in \mathbb{SH}_m$ is the subgroup of symplectic orthogonal matrices SpO_{2m} defined as:

$$\mathrm{SpO}_{2m} = \left\{ \begin{bmatrix} a & b \\ -b & a \end{bmatrix} : a^Ta + b^Tb = I_m, a^Tb \in \mathrm{Sym}_m \right\} = \mathrm{Sp}_{2m} \cap \mathrm{O}_{2m},$$

where O_{2m} is the group of orthogonal matrices. We thus have the identification $\mathbb{SH}_m \cong \mathrm{Sp}_{2m} / \mathrm{SpO}_{2m}$. The element in Sp_{2m} that transforms iI_m to $x = u + iv \in \mathbb{SH}_m$ via the group action is given by the map $\phi(\cdot)$ in the following identification:

$$\begin{aligned} \psi : \mathbb{SH}_m &\to \mathrm{Sp}_{2m} / \mathrm{SpO}_{2m} \\ x &\mapsto \begin{bmatrix} v^{\frac{1}{2}} & uv^{-\frac{1}{2}} \\ \mathbf{0} & v^{-\frac{1}{2}} \end{bmatrix} \mathrm{SpO}_{2m} = \phi(x)\,\mathrm{SpO}_{2m}\,. \end{aligned}$$

Riemannian Metric The Riemannian metric (also referred to as Symplectic metric) for the Siegel upper space model is given [37] by

$$ds_x^2 = 2\,\mathrm{Tr}(v^{-1}dxv^{-1}d\bar{x}), \tag{1}$$

where $x = u + iv \in \mathbb{SH}_m$, and $\mathrm{Tr}(\cdot)$ is the matrix trace (see Appendix 3.2 for further discussions). The associated Riemannian distance $d_{\mathbb{SH}}(x, y)$ between two points $x, y \in \mathbb{SH}_m$ is given by

$$d_{\mathbb{SH}}(x,y) = \sqrt{\sum_{j=1}^{m} \log^2\left(\frac{1+r_j^{\frac{1}{2}}}{1-r_j^{\frac{1}{2}}}\right)},$$

where $r_j, j = 1, \ldots, m$ are the eigenvalues of the cross-ratio $R(x, y)$ defined as

$$R(x,y) = (x-y)(x-\bar{y})^{-1}(\bar{x}-\bar{y})(\bar{x}-y)^{-1},$$

where $\bar{x}$ denotes the complex conjugate of x.

2.2 Riemannian Symmetric Spaces

This section briefly reviews key concepts from the theory of RSS for our work. We refer the interested reader to Appendix 3.1 for further discussions.

A symmetric space is a connected Riemannian manifold X with a geodesic-reversing isometry at each point. In other words, for each point $x \in X$ there is an isometry σ_x of X such that $\sigma_x(x) = x$ and the differential of σ_x at x is multiplication by -1 [7]. Siegel spaces belong to a family of RSS referred to as symmetric spaces of noncompact type or noncompact RSS. In the following, we refer to noncompact RSS as RSS or symmetric spaces. Let G be a connected noncompact semisimple Lie group with finite center, and let K be a maximal compact subgroup of G. Then a symmetric space X consists of the left cosets

$$X := G/K := \{x = gK | g \in G\}.$$

The action of G on $X = G/K$ is defined as $g[x] = g[hK] = ghK$ for $x = hK \in X, g, h \in G$. Let o be the origin K in X, then the map $\gamma : gK \mapsto g[o]$ is a diffeomorphism of G/K onto X.

Let $d(.,.)$ be the distance induced by the Riemannian metric. A *geodesic ray* in X is a map $\delta : [0, \infty) \to X$ such that $d(\delta(t), \delta(t')) = |t - t'|, \forall t, t' \geq 0$. A *geodesic line* in X is a map $\delta : \mathbb{R} \to X$ such that $d(\delta(t), \delta(t')) = |t - t'|, \forall t, t' \in \mathbb{R}$.

The geometry of X can be studied through the geometry of its *maximal flats* [2, 7, 19, 25]. A subspace $F \subset X$ is called a flat of dimension k (or a k-flat) if it is isometric to $\mathbb{R}^k$. The subspace F is called a maximal flat if it is not contained in any flat of bigger dimension. Since all maximal flats in X are isometric [19], they can be simultaneously identified with a model (maximal) flat F_{mod}.

Flats are decomposed into *Weyl chambers*. A Weyl chamber in a maximal flat F with tip at $x \in F$ is a connected component of the set of points $x' \in F \setminus \{x\}$ such that the geodesic line through x and x' is contained in a unique maximal flat [7]. Since G acts transitively on the set of Weyl chambers in X [19], they can be simultaneously identified with a Weyl chamber Δ. The subgroup of isometries of F which are induced by elements of G is isomorphic to a semidirect product $\mathbb{R}^r \rtimes W$. W is called the *Weyl group* of G and X.

Any symmetric space X is associated with a *boundary at infinity* ∂X constructed as the set of equivalence classes of geodesic rays in X. Two rays are considered equivalent if their images are a bounded distance apart [7]. The equivalence class of a geodesic ray δ is denoted by $\delta(\infty)$.

3 Proposed Approach

Our proposed point-to-hyperplane distances based on the quotient structure of Siegel spaces and the vector-valued distance are presented in Sections 3.1 and 3.2, respectively. In Section 3.3, we present our MLR models and introduce two variants of FC layers for Siegel neural networks. In our work, we focus on Siegel spaces but many of our results can also be stated for other RSS. To simplify the notation, we use the letters X, G, and K (see Section 2.2) to denote the spaces associated with Siegel spaces (see Section 2.1) unless otherwise stated.

3.1 Point-to-hyperplane Distances Based on the Quotient Structure of Siegel Spaces

3.1.1 Hyperplanes

We start with a formulation of Euclidean hyperplanes given as

$$\mathcal{H}^E_{a,b} = \{x \in \mathbb{R}^m : \langle x, a\rangle - b = 0\},$$

where $a \in \mathbb{R}^m \setminus \{\mathbf{0}\}$, $b \in \mathbb{R}$, and $\langle\cdot,\cdot\rangle$ is the Euclidean inner product. Hyperplane $\mathcal{H}^E_{a,b}$ can be reformulated [17] as

$$\mathcal{H}^E_{a,b} = \{x \in \mathbb{R}^m : \langle -p + x, a\rangle = 0\}, \tag{2}$$

where $p \in \mathbb{R}^m$ and $\langle p, a\rangle = b$.

To generalize Euclidean hyperplanes to our setting, we follow the approach in [31, 32] which relies on a binary operation, an inverse operation, and an inner product defined on the target space. Let $x = gK, y = hK \in X$ where $g, h \in G$. In the case of $\mathbb{SH}_m$, $g = \phi(x), h = \phi(y)$ where $\phi(u + iv) = \begin{bmatrix} v^{\frac{1}{2}} & uv^{-\frac{1}{2}} \\ \mathbf{0} & v^{-\frac{1}{2}} \end{bmatrix}, u + iv \in \mathbb{SH}_m$.

Definition 3.1 ([33])**.** *The binary operation $\oplus$ and inverse operation $\ominus$ are defined as*

$$x \oplus y = ghK, \quad \ominus x = g^{-1}K.$$

We propose the following inner product.

Definition 3.2. *The inner product $\langle\cdot,\cdot\rangle_{\mathbb{S}}$ on X is defined as*

$$\langle x, y\rangle_{\mathbb{S}} = \langle \log(gg^T), \log(hh^T)\rangle,$$

where $\log(\cdot)$ denotes the matrix logarithm.

Our proposed inner product is motivated by Proposition 3.3. (see Appendix 4.1 for its proof).

Proposition 3.3. *The inner product $\langle\cdot,\cdot\rangle_{\mathbb{S}}$ agrees with the Riemannian distance, i.e.,*

$$\| \ominus x \oplus y\|_{\mathbb{S}} \propto d_{\mathbb{SH}}(x, y),$$

where $x, y \in X$, and the norm $\|\cdot\|_{\mathbb{S}}$ is induced by the inner product $\langle\cdot,\cdot\rangle_{\mathbb{S}}$. Furthermore, the inner product $\langle\cdot,\cdot\rangle_{\mathbb{S}}$ is invariant under the action of K, i.e., for any $k \in K$,

$$\langle x, y\rangle_{\mathbb{S}} = \langle k[x], k[y]\rangle_{\mathbb{S}}.$$

Note that both properties in Proposition 3.3 are satisfied by the inner products in [31, 33] and the second property is also satisfied by the one in [20]. Note also that these properties hold for the more general case in which G is the general linear group or its subgroup, and K is the group of orthogonal matrices or its subgroup (see Appendix 4.1). We are now ready to define hyperplanes.

Definition 3.4. *Let $a, p \in X$. Then hyperplanes on X are defined as*

$$\mathcal{H}_{a,p} = \{x \in X : \langle \ominus p \oplus x, a\rangle_{\mathbb{S}} = 0\}.$$

Segments of the form $\ominus p \oplus x$ can be regarded as Siegel analogs of Euclidean lines. Thus, $\mathcal{H}_{a,p}$ has a similar interpretation as a Euclidean hyperplane, i.e., the former contains a fixed point $p \in X$ and any point $x \in X$ such that the segment $\ominus p \oplus x$ is orthogonal to a fixed direction a. Therefore, hyperplanes as given in Definition 3.4 are natural extensions of Euclidean hyperplanes.

3.1.2 Point-to-hyperplane Distance

The distance $\bar{d}(x, \mathcal{H}_{a,p})$ between a point $x \in X$ and a hyperplane $\mathcal{H}_{a,p}$ given in Definition 3.4 can be formulated [31] as

$$\bar{d}(x, \mathcal{H}_{a,p}) = \sin(\angle xp\bar{q})d(x, p),$$

where $\angle xp\bar{q}$ is the gyroangle [31, 41] (see Appendix 3.4) between $\ominus p \oplus x$ and $\ominus p \oplus \bar{q}$, and $\bar{q}$ is computed as

$$\bar{q} = \underset{q \in \mathcal{H}_{a,p} \setminus \{p\}}{\arg\max} \left(\frac{\langle \ominus p \oplus q, \ominus p \oplus x\rangle_{\mathbb{S}}}{\| \ominus p \oplus q\|_{\mathbb{S}} \| \ominus p \oplus x\|_{\mathbb{S}}} \right),$$

By convention, $\sin(\angle xpq) = 0$ for any $x, q \in \mathcal{H}_{a,p}$. Theorem 3.5 gives a closed form for the point-to-hyperplane distance on Siegel spaces (see Appendix 4.2 for its proof).

Theorem 3.5. *Let $x, a, p \in X$ and let $\mathcal{H}_{a,p}$ be a hyperplane as given in Definition 3.4. Then*

$$\bar{d}(x, \mathcal{H}_{a,p}) = \frac{|\langle \log(\phi(p)^{-1}\phi(x)\phi(x)^T\phi(p)^{-T}), \log(\phi(a)\phi(a)^T)\rangle|}{\|\log(\phi(a)\phi(a)^T)\|},$$

where $\|\cdot\|$ denotes the Euclidean norm, and the map $\phi(\cdot)$ is given in Section 2.1.

3.1.3 Product Spaces

We now extend the above method to the product space setting. Let X be defined as the Cartesian product $X = X_1 \times \ldots \times X_L$, where $X_j = G_j/K_j, j = 1, \ldots, L$ are RSS, G_j is a connected noncompact semisimple Lie group with finite center, K_j is a maximal compact subgroup of G_j. Here we focus on the Cartesian product of SPD and Siegel spaces. Each point $x \in X$ can be described through its coordinates $x = (x_1, \ldots, x_L), x_j \in X_j, j = 1, \ldots, L$. In this setting, one has simple decompositions of the tangent space, the exponential map, and the squared Riemannian distance [16, 18, 40]. When X_j is an SPD space, G_j is the general linear group and K_j is the group of orthogonal matrices (see Appendix 3.3). Thus one can define the binary operation, inverse operation, and inner product on X_j as in Definitions 3.1 and 3.2, and the results in Proposition 3.3 still hold. By abuse of notation, we shall use the same notations for those operations as in Section 3.1.1.

Definition 3.6. *Let $x = (x_1, \ldots, x_L)$, $y = (y_1, \ldots, y_L) \in X, x_j, y_j \in X_j, j = 1, \ldots, L$. The binary operation $\oplus$ and inverse operation $\ominus$ on X are defined as*

$$x \oplus y = (x_1 \oplus y_1, \ldots, x_L \oplus y_L), \ \ominus x = (\ominus x_1, \ldots, \ominus x_L).$$

Definition 3.7. *The inner product $\langle \cdot, \cdot \rangle_{\mathbb{S}}$ on X is defined as*

$$\langle x, y \rangle_{\mathbb{S}} = \sum_{j=1}^{L} \langle x_j, y_j \rangle_{\mathbb{S}}.$$

The following theorem (see Appendix 4.3 for its proof) extends Theorem 3.5 to the considered setting.

Theorem 3.8. *Let $\mathcal{H}_{a,p}$ be a hyperplane as given in Definition 3.4, where $a = (a_1, \ldots, a_L), p = (p_1, \ldots, p_L), a_j = w_j K_j, p_j = h_j K_j \in X_j, w_j, h_j \in G_j, j = 1, \ldots, L$, and let $x = (x_1, \ldots, x_L) \in X$ where $x_j = g_j K_j \in X_j, g_j \in G_j$. Then*

$$\bar{d}(x, \mathcal{H}_{a,p}) = \frac{|\sum_{j=1}^{L} \langle \log(h_j^{-1} g_j g_j^T h_j^{-T}), \log(w_j w_j^T)\rangle|}{\sqrt{\sum_{j=1}^{L} \|\log(w_j w_j^T)\|^2}}.$$

3.2 Point-to-hyperplane Distances Based on Vector-Valued Distances

As shown in [36], the formulation of Euclidean hyperplanes in Eq. (2) has an over-parameterization issue, i.e., it increases the number of parameters from $m + 1$ to $2m$ in each class. Our formulation of hyperplanes in Section 3.1.1 (see Definition 3.4) follows that formulation and thus suffers from a similar issue. In this section, we propose another method for constructing the point-to-hyperplane distance which results in more compact MLR layers for Siegel neural networks.

3.2.1 Hyperplanes

We start with a similar formulation of Euclidean hyperplanes in Eq. (2) but use a different parameterization. Given $p \in \mathbb{R}^m$ and $\xi \in \partial\mathbb{R}^m$, the Euclidean hyperplane $\mathcal{H}^E_{\xi,p}$ parameterized by p and ξ can be defined [33] by

$$\mathcal{H}^E_{\xi,p} = \{x \in \mathbb{R}^m : \langle p - x, a \rangle = 0\} = \{x \in \mathbb{R}^m : \langle \mathrm{vec}(x, p), a \rangle = 0\},$$

where ξ is the equivalence class of the geodesic ray $\delta(t) = t\frac{a}{\|a\|}, a \in \mathbb{R}^m \setminus \{\mathbf{0}\}$, and the function $\mathrm{vec}(x, p) = p - x$ denotes the translation carrying x to p.

In a symmetric space, a natural analog of the function $\mathrm{vec}(\cdot, \cdot)$ is the vector-valued distance function [24, 25]. Given two points $x, y \in X$, one computes a G-invariant distance by first transforming (via the G-action) x and y to x' and y' on the model flat F_{mod}, respectively, and then identifying the

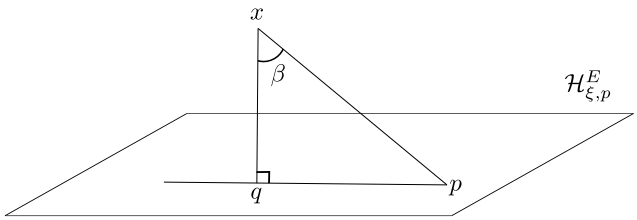

Figure 1: The distance between a point $x \in \mathbb{R}^m$ and a Euclidean hyperplane $\mathcal{H}^E_{\xi,p}$.

translation modulo the action of the Weyl group carrying x' to y'. Note that in $\mathbb{R}^m$, the projections of x and p on a maximal flat are precisely x and p, respectively. The domain of the resulting distance function, which is a fundamental domain for the action of the Weyl group on the translations, can be canonically identified with the Weyl chamber Δ. The above observation motivates the following definition.

Definition 3.9. *Let $d_\Delta(\cdot,\cdot) : X \times X \to \Delta$ be the vector-valued distance function on X. Let $p \in X$, $\xi \in \partial X$, and let $a_\xi \in \Delta$ be such that ξ is the equivalent class of the geodesic ray $\delta(t) = k\exp(ta_\xi)K, k \in K$. Then hyperplane $\mathcal{H}_{\xi,p}$ is defined as*

$$\mathcal{H}_{\xi,p} = \{x \in X : \langle d_\Delta(x,p), a_\xi\rangle = 0\}.$$

A hyperplane given in Definition 3.9 has a clear interpretation, i.e., it contains a fixed point $p \in X$ and any point $x \in X$ such that the vector-valued distance between x and p is orthogonal to a fixed direction a_ξ. We note that the notion of vector-valued distance has been employed in [28, 29] for learning and visualizing embeddings in natural language processing and graph tasks. However, none of those works reveals the analogies discussed above for defining Siegel hyperplanes.

3.2.2 Point-to-hyperplane Distance

Let $p \in \mathbb{R}^m$ and $\xi \in \partial\mathbb{R}^m$. The distance $\bar{d}(x, \mathcal{H}^E_{\xi,p})$ between a point $x \in \mathbb{R}^m$ and hyperplane $\mathcal{H}^E_{\xi,p}$ can be computed (see Fig. 1) as

$$\bar{d}(x, \mathcal{H}^E_{\xi,p}) = d(x,p)\cos(\beta),$$

where β is the angle between the segments $[x,p]$ and $[x,q]$, and q is the projection of x on $\mathcal{H}^E_{\xi,p}$. By convention, $\bar{d}(x, \mathcal{H}^E_{\xi,p}) = 0, \forall x \in \mathcal{H}^E_{\xi,p}$. The above equation can be rewritten as

$$\bar{d}(x, \mathcal{H}^E_{\xi,p}) = d(x,p)\cos\angle_x(p,\xi),$$

where $\angle_x(p,\xi)$ denotes the angle at x between the geodesic segment $[x,p]$ and the geodesic ray which issues from x and is in the class ξ. We generalize the above equation to our setting.

Definition 3.10. *Let $p \in X$, $\xi \in \partial X$, and let $\mathcal{H}_{\xi,p}$ be a hyperplane in X. Then the (signed) distance $\bar{d}(x, \mathcal{H}_{\xi,p})$ between a point $x \in X$ and hyperplane $\mathcal{H}_{\xi,p}$ is defined as*

$$\bar{d}(x, \mathcal{H}_{\xi,p}) = d(x,p)\cos\angle_x(p,\xi).$$

Deriving a closed form of the point-to-hyperplane distance for applications from Definition 3.10 is not trivial. However, one can obtain an upper bound of this distance which is given in Proposition 3.11 (see Appendix 4.4 for its proof).

Proposition 3.11. *Let $x, p \in X$, $\xi \in \partial X$, and let $a_\xi \in \Delta$ be such that ξ is the equivalent class of the geodesic ray $\delta(t) = k\exp(ta_\xi)K$, where $k \in K$ and $\exp(\cdot)$ is the matrix exponential. Then*

$$\bar{d}(x, \mathcal{H}_{\xi,p}) \leq \langle d_\Delta(x,p), a_\xi\rangle .$$

Note that the point-to-hyperplane distance in Section 3.1 as well as those in [17, 31, 32] are obtained by solving an optimization problem in Euclidean spaces. This is different from our method in this section which estimates an upper bound of the point-to-hyperplane distance on the target spaces.

3.3 Neural Networks on Siegel Spaces

In this section, we show how to construct MLR layers for Siegel neural networks using the tools introduced in Sections 3.1 and 3.2. We also propose two types of FC layers which are crucial building blocks in the context of deep neural networks.

3.3.1 MLR Layers

We follow the approach in [17, 27] for building Riemannian MLR. Given M classes, (Euclidean) MLR computes the probability of each of the output classes as

$$p(y=j|x) = \frac{\exp(a_j^T x - b_j)}{\sum_{j=1}^{M} \exp(a_j^T x - b_j)} \propto \exp(a_j^T x - b_j), \tag{3}$$

where x is an input sample, $b_j \in \mathbb{R}$, $x, a_j \in \mathbb{R}^m, j = 1, \ldots, M$, and $\exp(\cdot)$ is the ordinary exponential function (by abuse of notation). As shown in [27], Eq. (3) can be rewritten as

$$p(y=j|x) \propto \exp(\text{sign}(a_j^T x - b_j)\|a_j\|\bar{d}(x, \mathcal{H}^E_{a_j,b_j})),$$

where $\bar{d}(x, \mathcal{H}^E_{a_j,b_j})$ is the distance from point x to hyperplane $\mathcal{H}^E_{a_j,b_j}$ (see Section 3.1.1). In our case, a hyperplane can be parameterized by two elements in X (see Definition 3.4), or by an element in X and an element in Δ (see Definition 3.9). We replace the expression in the argument of the function $\exp(\cdot)$ by the distances in Theorems 3.5 and 3.8 as well as the upper bound of the point-to-hyperplane distance in Proposition 3.11. The final formulations of our MLR layers are given in Appendix 1.

3.3.2 FC Layers

The FC layer with group action (AFC) Let $a + ib \in \mathbb{SH}_m$. Then the element in Sp_{2m} mapping iI_m to $a + ib$ via the group action (see Section 2.1) is given by

$$\phi(a+ib) = \begin{bmatrix} b^{\frac{1}{2}} & ab^{-\frac{1}{2}} \\ \mathbf{0} & b^{-\frac{1}{2}} \end{bmatrix}.$$

Given an input $x \in \mathbb{SH}_m$, the output of the AFC layer is obtained by taking the group action $\phi(a+ib)[x]$. This leads us to the following construction.

Definition 3.12. *Let $x = u + iv \in \mathbb{SH}_m$ be the input of the AFC layer. Then the output of the AFC layer is given by:*

$$t = (b^{\frac{1}{2}} u b^{\frac{1}{2}} + a) + i b^{\frac{1}{2}} v b^{\frac{1}{2}},$$

where $a \in \text{Sym}_m$ and $b \in \text{Sym}^+_m$ are the parameters of the layer.

We have that $b^{\frac{1}{2}} u b^{\frac{1}{2}} + a \in \text{Sym}_m$ and $b^{\frac{1}{2}} v b^{\frac{1}{2}} \in \text{Sym}^+_m$ by construction. Hence, the AFC layer always outputs points on $\mathbb{SH}_m$. The transformation performed by the AFC layer can be interpreted as a translation of the input x by $a + ib$ (see Section 3.1.1).

The FC layer for dimensionality reduction (DFC) Based on the definition of the AFC layer, another type of FC layers for Siegel neural networks can also be built using a method similar to [21].

Definition 3.13. *Let St_{m,m_2} be the space of $m \times m_2$ real matrices ($m > m_2$) with mutually orthogonal columns of unit length (the compact Stiefel manifold), and let $x = u + iv \in \mathbb{SH}_m$ be the input of the DFC layer. Then the output of the DFC layer is given by:*

$$t = (b^T u b + a) + i b^T v b,$$

where $a \in \text{Sym}_{m_2}$ and $b \in \text{St}_{m,m_2}$ are the parameters of the layer.

Our FC layers generalize some FC layers in previous works. Specifically, when $u = 0$ and $a = 0$, the imaginary part $b^{\frac{1}{2}} v b^{\frac{1}{2}}$ of the output of the AFC layer corresponds to the transformation performed by the affine-invariant translation layer [31], and the imaginary part $b^T v b$ of the output of the DFC layer corresponds to the transformation performed by the well-known Bimap layer [21]. In [38], the authors also proposed FC layers for neural networks on RSS. However, these layers are different from our FC layers in some aspects. First, the former include activation functions which are not used in the latter. Second, the former do not output points on the considered spaces, as opposed to the latter which always output points on these spaces.

Method	Dataset 1 $(3, 600, 3600)$	Dataset 2 $(4, 100, 2000)$	Dataset 3 $(5, 80, 1600)$	Dataset 4 $(6, 50, 500)$
kNN [11]	76.22±0.0	93.00±0.0	76.75±0.0	73.20±0.0
SPDNet [21]	63.44±0.11	41.50±0.12	45.88±0.15	66.80±0.04
SPDNetBN [9]	62.67±0.10	45.10±0.08	45.75±0.15	68.40±0.04
MLR-AI [31]	65.61±0.15	47.40±0.12	46.12±0.17	67.60±0.04
GyroSpd++ [32]	62.24±0.16	46.20±0.14	48.25±0.19	67.80±0.08
SiegelNet-DFC-QMLR$_{\mathrm{Sym}_m^+ \times \mathbb{SH}_m^{q-1}}$ (Ours)	40.78±0.23	82.70±0.18	74.88±0.21	71.20±0.08
SiegelNet-AFC-QMLR$_{\mathrm{Sym}_m^+ \times \mathbb{SH}_m^{q-1}}$ (Ours)	**80.94±0.14**	**96.50±0.12**	**91.00±0.18**	**85.60±0.06**

Table 1: Results (mean accuracy ± standard deviation) computed over 10 runs for radar clutter classification. The tuple (m, M, s) below each dataset indicates the signal dimension m, the number of classes M, and the size of the dataset s.

4 Related Work

Existing MLR models on Riemannian manifolds are generally built on either SPD manifolds [13, 31] and their low-rank counterparts [32] or hyperbolic spaces [5, 17, 27, 36]. Many of them [17, 31, 32, 36] leverage the gyro-structures of the Poincaré ball and SPD manifolds. The work in [33] proposes MLR and FC layers for neural network on RSS which rely on the construction of Busemann functions. The work in [38] analyzes some existing hyperbolic and SPD neural networks from the perspective of harmonic analysis on RSS. It mainly concerns with a constructive proof of the universal approximation property of finite neural networks on RSS. Our method in Section 3.1 is inspired by the works in [17, 31, 32] and focuses on Siegel spaces. Our method in Sections 3.2 explores the connection between the point-to-hyperplane distance and the vector-valued distance which has not been investigated in previous works.

5 Experiments

This section reports results of our experiments on the radar clutter classification and node classification tasks. For further details, please refer to Appendix 1 in which we present more experimental results on human action recognition and Riemannian generative modeling.

5.1 Radar Clutter Classification

Radar clutter classification aims at recognizing different types of radar clutter which is the information recorded by a radar related to seas, forests, fields, cities and other environmental elements surrounding the radar [10]. Due to the scarcity of publicly available radar datasets for the task, our experiments are performed using simulated radar signals[1] which are commonly assumed to be stationary centered autoregressive (AR) Gaussian time series [3, 4, 6, 10]. The AR model is given by

$$u_n + \sum_{j=1}^{q} c_j u_{n-j} = v_n,$$

where q ($q > 1$) is the order of the AR model, $u_n \in \mathbb{C}^m$ is the vector of signals at time n, $c_j \in \mathbb{C}^{m\times m}, j = 1, \ldots, q$ are the prediction coefficients (AR parameters), and $v_n \in \mathbb{C}^m$ is the prediction error at time n which is assumed to be a multidimensional Gaussian random variable (detailed descriptions of the construction of our datasets are provided in Appendix 1.1). To compute an input data for our networks from a time series, we parameterize the time series as $(p_0, w_1, \ldots, w_{q-1}) \in \mathbb{H}_m^+ \times \mathbb{SD}_m^{q-1}$, where $p_0 \in \mathbb{H}_m^+$ and $w_1, \ldots, w_{q-1} \in \mathbb{SD}_m$ (see Appendix 1.1). We note that methods dealing with data that lie on these product spaces have already been studied in previous works [4, 10, 11]. These representation spaces are endowed with a natural metric inspired by information geometry [4, 10]. We discard the imaginary part of the component p_0 and map it to an SPD matrix $\tilde{p}_0$ (see Appendix 1.1). Each component w_i is converted to $z_i \in \mathbb{SH}_m$ using the inverse matrix Cayley transformation (see Section 2.1). The input data is thus represented by point $(\tilde{p}_0, z_1, \ldots, z_{q-1}) \in \mathrm{Sym}_m^+ \times \mathbb{SH}_m^{q-1}$.

[1] https://github.com/nguyenxuanson10/synthetic-data

Method	Glass	Iris	Zoo
kNN [11]	29.65±0.0	31.66±0.0	33.33±0.0
LogEig classifier [28]	41.54±4.22	34.33±3.46	51.04±3.53
SiegelNet-BFC-BMLR [33]	41.12±3.86	37.26±2.53	48.12±3.08
SiegelNet-AFC-VMLR (Ours)	42.06±4.23	36.94±3.68	50.86±3.26
SiegelNet-AFC-QMLR$_{\mathbb{SH}_m}$ (Ours)	**45.79±4.66**	**38.20±3.03**	**53.37±4.23**

Table 2: Results (mean accuracy ± standard deviation) computed over 10 runs for node classification.

Method	Glass	Iris	Zoo
SiegelNet-BMLR [33]	40.55±3.50	36.94±2.09	46.43±3.64
SiegelNet-VMLR (Ours)	41.78±4.11	36.89±3.73	50.38±3.47
SiegelNet-QMLR$_{\mathbb{SH}_m}$ (Ours)	**42.61±3.26**	**37.52±2.54**	**52.00±4.78**

Table 3: Comparison (mean accuracy ± standard deviation) of MLR models on Siegel spaces.

Each of our networks consists of an FC (AFC or DFC) layer and a MLR layer built on the distance in Theorem 3.8. The sizes of the parameter b in the DFC layer are set to 3×2, 4×3, 5×3, and 6×4 for the experiments on datasets 1, 2, 3, and 4, respectively. We compare our approach to the following methods: (1) k-Nearest Neighbors (kNN) based on the Kähler distance [11] which is among the very few works for supervised classification in the product space $\mathbb{H}_m^+ \times \mathbb{SD}_m^{q-1}$; and (2) state-of-the-art SPD neural networks [9, 21, 31, 32] which use the real parts of the covariance matrices estimated from the time series as input data (the real parts are mapped to SPD matrices as above). We use default settings for SPD models as in the original papers (see Appendix 1.1). Results in Tab. 1 show that SiegelNet-AFC-QMLR$_{\mathrm{Sym}_m^+ \times \mathbb{SH}_m^{q-1}}$ yields the best performance in terms of mean accuracy across all the datasets. It is able to improve upon kNN, the second best method, by a margin of 4.71%, 3.5%, 14.25%, and 12.39% on datasets 1, 2, 3, and 4, respectively. There are large gaps in the performance of our models, yet in most cases, our worst model still outperforms SPD models by large margins. The results of our networks and kNN demonstrate the representation power of Siegel spaces in the considered application. This is also confirmed by our experiments (see Appendix 1.1) in which the performance of SiegelNet-AFC-QMLR$_{\mathrm{Sym}_m^+ \times \mathbb{SH}_m^{q-1}}$ drops drastically when the coordinates associated with the product space $\mathbb{SH}_m^{q-1}$ (i.e., $z_1, \ldots, z_{q-1}$) are removed from the input data.

5.2 Node Classification

We perform node classification experiments on Glass, Iris, and Zoo datasets from the UCI Machine Learning Repository [15][2]. Like [28], our main aim is to demonstrate the applicability of our approach on Siegel spaces, and we do not necessarily seek state-of-the-art results for the target task.

To create input data which are graph node embeddings on Siegel spaces, we optimize a distance-based loss function [18, 28]. Given the distances $\{d_G(j_1, j_2)\}_{j_1, j_2=1}^M$ between all pairs of connected nodes j_1 and j_2, the loss function is given by:

$$\mathcal{L}(x) = \sum_{j_1, j_2=1}^{M} \left| \left(\frac{d_{\mathbb{SH}}(x_{j_1}, x_{j_2})}{d_G(j_1, j_2)} \right)^2 - 1 \right|,$$

where x_{j_1} and x_{j_2} are the node representations on the embedding space of nodes j_1 and j_2, respectively, and $d_{\mathbb{SH}}(\cdot, \cdot)$ is the distance function given in Section 2.1. This loss function captures the average distortion. We use the cosine distance to compute a complete input distance graph from the original features of the data points [12, 28]. After the node embeddings[3] are learned, they are used as input features for all methods. In our experiments, the embedding dimension is set to 6.

Each of our networks consists of an AFC layer and a MLR (QMLR$_{\mathbb{SH}_m}$ or VMLR) layer. The QMLR$_{\mathbb{SH}_m}$ and VMLR layers are built using the distances in Theorem 3.5 and the upper bound of $\bar{d}(x, \mathcal{H}_{\xi,p})$ in Proposition 3.11, respectively. We compare our networks to the following methods:

[2] https://archive.ics.uci.edu/datasets

[3] https://github.com/nguyenxuanson10/synthetic-data

(1) kNN based on the distance function $d_{\mathbb{SH}}(\cdot,\cdot)$; (2) LogEig classifier [28]; and (3) SiegelNet-BFC-BMLR which consists of an FC (BFC) layer and a MLR (BMLR) layer based on Busemann functions [33]. Results in Tab. 2 show that SiegelNet-AFC-QMLR$_{\mathbb{SH}_m}$ gives the best mean accuracies across all the datasets. In terms of mean accuracy, SiegelNet-AFC-VMLR surpasses SiegelNet-BFC-BMLR on Glass and Zoo datasets. SiegelNet-AFC-VMLR also surpasses the LogEig classifier on Glass and Iris datasets. Tab. 3 reports the results of SiegelNet-BFC-BMLR and our networks without FC layers. It can be observed that our MLR models achieve higher (mean) accuracies than the BMLR model. Specifically, SiegelNet-QMLR$_{\mathbb{SH}_m}$ improves upon SiegelNet-BMLR by a margin of 2.06%, 0.58%, and 5.57% on Glass, Iris, and Zoo datasets, respectively. SiegelNet-AFC-QMLR$_{\mathbb{SH}_m}$ is able to improve by 3.17%, 0.67%, and 1.36% w.r.t. SiegelNet-QMLR$_{\mathbb{SH}_m}$ on Glass, Iris, and Zoo datasets, respectively, demonstrating the effectiveness of the AFC layer. Although SiegelNet-VMLR is outperformed by SiegelNet-QMLR$_{\mathbb{SH}_m}$, it is important to note that the model size of the former is about two times smaller than that of the latter (see Appendix 1.2).

6 Limitation of Our Approach

A limitation of our method in Section 3.1 is that our formulation of Siegel hyperplanes suffers from an over-parameterization issue. We alleviate this problem by reparameterizing Siegel hyperplanes as proposed in Section 3.2. However, this new parameterization does not yield competitive performance compared to the original one.

Our methods rely on operations on Siegel spaces which are generally expensive. Our method in Section 5.1 suffers from high computational cost in the setting of high-dimensional radar signals. Similarly, the loss function in Section 5.2 is based on the average distortion, for which the distances over all pairs of points must be computed during training. Since the computation of the Riemannian distance between two points on a Siegel space (see Section 2.1) is based on eigenvalue decomposition, our method in Section 5.2 is computationally expensive when it comes to learning on large graphs.

Like hyperbolic and SPD spaces, Siegel spaces are spaces of non-positive curvature. Therefore, our method in Section 5.2 does not allow isometric embeddings of graphs with a different curvature property, e.g., non-negative curvature. Although it can still be applied in this case, the learned node embedding may not preserve the curvature property of the embedded graph, leading to poor performance. Furthermore, like other graph embedding approaches, low-dimensional embeddings on Siegel spaces are not able to capture complex relationships within data which can affect the performance of our method.

7 Conclusion

We have proposed Riemannian MLR and FC layers which enable the construction of effective Siegel neural networks. Our MLR layers are built upon the quotient structure of Siegel spaces and the concept of vector-valued distance on RSS. Our FC layers are based on the action of the real symplectic group on Siegel spaces. We have provided experimental evaluations demonstrating state-of-the-art performance of our approach in the radar clutter classification and node classification tasks.

There are several potential improvements and extensions to Siegel neural networks that could be addressed as future work. Based on our experimental results, it can be observed that the DFC layer gives inferior performance compared to the AFC layer. It is therefore desirable to develop alternative layers for the DFC layer which are able to achieve better performance. Also, important building blocks such as convolutional layers, batch normalization layers, pooling layers, and attention layers are not studied in our work. Those are crucial to the development of effective deep Siegel neural networks.

Acknowledgments

We are grateful for the constructive comments and feedback from the anonymous reviewers.

References

[1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Fast and Simple Computations on Tensors with Log-Euclidean Metrics. Technical Report RR-5584, INRIA, 2005. 1

[2] W. Ballmann. *Lectures on Spaces of Nonpositive Curvature*. Birkhäuser, 2012. 3

[3] F. Barbaresco. Super-resolution Spectrum Analysis Regularization: Burg, Capon and AGO-antagonistic Algorithms. In *EUSIPCO-96*, pages 1–4, 1996. 8

[4] F. Barbaresco. Information Geometry of Covariance Matrix: Cartan-Siegel Homogeneous Bounded Domains, Mostow/Berger Fibration and Fréchet Median. *Matrix Information Geometry*, pages 199–255, 2013. 8

[5] A. Bdeir, K. Schwethelm, and N. Landwehr. Fully Hyperbolic Convolutional Neural Networks for Computer Vision. In *ICLR*, 2024. 8

[6] J. B. Billingsley. *Low-angle Radar Land Clutter, Measurements and Empirical Models*. William Andrew Publishing, 2002. 8

[7] M. Bridson and A. Häfliger. *Metric Spaces of Non-Positive Curvature*. Springer Berlin Heidelberg, 2011. 3

[8] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *CoRR*, abs/2104.13478, 2021. 1

[9] D. A. Brooks, O. Schwander, F. Barbaresco, J.-Y. Schneider, and M. Cord. Riemannian Batch Normalization for SPD Neural Networks. In *NeurIPS*, pages 15463–15474, 2019. 8, 9

[10] Y. Cabanes. *Multidimensional Complex Stationary Centered Gaussian Autoregressive Time Series Machine Learning in Poincaré and Siegel Disks: Application for Audio and Radar Clutter Classification*. Theses, 2022. 8

[11] Y. Cabanes and F. Nielsen. Classification in the Siegel Space for Vectorial Autoregressive Data. In *Geometric Science of Information: 5th International Conference*, pages 693–700, 2021. 8, 9

[12] I. Chami, A. Gu, V. Chatziafratis, and C. Ré. From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering. In *NeurIPS*, 2020. 9

[13] Z. Chen, Y. Song, G. Liu, R. R. Kompella, X. Wu, and N. Sebe. Riemannian Multinomial Logistics Regression for SPD Neural Networks. *CoRR*, abs/2305.11288, 2024. 8

[14] Z. Chen, Y. Song, X. Wu, and N. Sebe. Gyrogroup Batch Normalization. In *ICLR*, 2025. 1

[15] D. Dua and C. Graff. UCI machine learning repository, 2017. 9

[16] F. A. Ficken. The Riemannian and Affine Differential Geometry of Product-Spaces. *Annals of Mathematics*, 40(4):892–913, 1939. 5

[17] O.-E. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic neural networks. In *NeurIPS*, pages 5350–5360, 2018. 1, 2, 4, 6, 7, 8

[18] A. Gu, F. Sala, B. Gunel, and C. Ré. Learning Mixed-Curvature Representations in Products of Model Spaces. In *ICLR*, 2019. 5, 9

[19] S. Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*. ISSN. Elsevier Science, 1979. 1, 3

[20] S. Helgason. *Geometric Analysis on Symmetric Spaces*. Mathematical surveys and monographs. American Mathematical Society, 1994. 4

[21] Z. Huang and L. V. Gool. A Riemannian Network for SPD Matrix Learning. In *AAAI*, pages 2036–2042, 2017. 1, 7, 8, 9

[22] Z. Huang, C. Wan, T. Probst, and L. V. Gool. Deep Learning on Lie Groups for Skeleton-Based Action Recognition. In *CVPR*, pages 6099–6108, 2017. 1

[23] Z. Huang, J. Wu, and L. V. Gool. Building Deep Networks on Grassmann Manifolds. In *AAAI*, pages 3279–3286, 2018. 1

[24] M. Kapovich, B. Leeb, and J. J. Millson. The Generalized Triangle Inequalities in Symmetric Spaces and Buildings with Applications to Algebra. *CoRR*, abs/math/0210256, 2005. 5

[25] M. Kapovich, B. Leeb, and J. Porti. Anosov Subgroups: Dynamical and Geometric Characterizations. *CoRR*, abs/1703.01647, 2017. 2, 3, 5

[26] I. Katsman, E. M. Chen, S. Holalkere, A. Asch, A. Lou, S.-N. Lim, and C. D. Sa. Riemannian Residual Neural Networks. *CoRR*, abs/2310.10013, 2023. 1

[27] G. Lebanon and J. Lafferty. Hyperplane Margin Classifiers on the Multinomial Manifold. In *ICML*, page 66, 2004. 7, 8

[28] F. López, B. Pozzetti, S. Trettel, M. Strube, and A. Wienhard. Symmetric Spaces for Graph Embeddings: A Finsler-Riemannian Approach. In *ICML*, pages 7090–7101, 2021. 1, 2, 6, 9, 10

[29] F. López, B. Pozzetti, S. Trettel, M. Strube, and A. Wienhard. Vector-valued Distance and Gyrocalculus on the Space of Symmetric Positive Definite Matrices. In *NeurIPS*, pages 18350–18366, 2021. 6

[30] X. S. Nguyen. The Gyro-Structure of Some Matrix Manifolds. In *NeurIPS*, pages 26618–26630, 2022. 1, 2

[31] X. S. Nguyen and S. Yang. Building Neural Networks on Matrix Manifolds: A Gyrovector Space Approach. In *ICML*, pages 26031–26062, 2023. 1, 4, 6, 7, 8, 9

[32] X. S. Nguyen, S. Yang, and A. Histace. Matrix Manifold Neural Networks++. In *ICLR*, 2024. 1, 4, 6, 8, 9

[33] X. S. Nguyen, S. Yang, and A. Histace. Neural Networks on Symmetric Spaces of Noncompact Type. In *ICLR*, 2025. 1, 4, 5, 8, 9, 10

[34] F. Nielsen. The Siegel–Klein Disk: Hilbert Geometry of the Siegel Disk Domain. *Entropy*, 22(9), 2020. 1

[35] X. Pennec. *Statistical Computing on Manifolds for Computational Anatomy*. Habilitation à diriger des recherches, Université Nice Sophia-Antipolis, 2006. 1

[36] R. Shimizu, Y. Mukuta, and T. Harada. Hyperbolic Neural Networks++. *CoRR*, abs/2006.08210, 2021. 1, 2, 5, 8

[37] C. L. Siegel. Symplectic Geometry. *American Journal of Mathematics*, 65:1–86, 1943. 1, 3

[38] S. Sonoda, I. Ishikawa, and M. Ikeda. Fully-Connected Network on Noncompact Symmetric Space and Ridgelet Transform based on Helgason-Fourier Analysis. In *ICML*, pages 20405–20422, 2022. 7, 8

[39] D. Taha, W. Zhao, J. M. Riestenberg, and M. Strube. Normed Spaces for Graph Embedding. *CoRR*, abs/2312.01502, 2023. 2

[40] P. K. Turaga and A. Srivastava. *Riemannian Computing in Computer Vision*. Springer Publishing Company, Incorporated, 2015. 5

[41] A. A. Ungar. *Analytic Hyperbolic Geometry in N Dimensions: An Introduction*. CRC Press, 2014. 4

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We clearly state the paper's scope in the abstract and introduction, and our contributions at the end of the introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Please refer to Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: We clearly state the assumptions for each theoretical result and provide all the proofs in Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide all details needed to reproduce our experimental results in the main paper and Appendix.

   Guidelines:

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

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

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

Answer: [No]

Justification: The datasets used for our experiments will be made available upon acceptance of the paper. In the main paper and Appendix, we already give details on our experimental settings and implementation which would be sufficient to reproduce our experimental results.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We provide all details regarding our experiments in the main paper and Appendix. Those details would be sufficient for the reader to understand and reproduce our experimental results.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: We report mean accuracy and standard deviation over several runs for all competing methods.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the information about the computer resources used for our experiments. We also provide a complexity analysis (memory and time) in Appendix 1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We already checked the NeurIPS Code of Ethics and think that our research conducted in the paper conform with it.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Please refer to Appendix 2.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: We think this question is not applied to our paper.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly cite all works and provide all links to codes and data used for our paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: We do not provide any assets in our work.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: Our paper does not deal with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification: Our paper does not deal with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We did not use LLMs in any process of our work (writing, coding, etc,)

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.