
Symmetric Spaces for Graph Embeddings: A Finsler-Riemannian Approach

Federico López¹ Beatrice Pozzetti² Steve Trettel³ Michael Strube¹ Anna Wienhard²

Abstract

Learning faithful graph representations as sets of vertex embeddings has become a fundamental intermediary step in a wide range of machine learning applications. We propose the systematic use of symmetric spaces in representation learning, a class encompassing many of the previously used embedding targets. This enables us to introduce a new method, the use of Finsler metrics integrated in a Riemannian optimization scheme, that better adapts to dissimilar structures in the graph. We develop a tool to analyze the embeddings and infer structural properties of the data sets. For implementation, we choose Siegel spaces, a versatile family of symmetric spaces. Our approach outperforms competitive baselines for graph reconstruction tasks on various synthetic and real-world datasets. We further demonstrate its applicability on two downstream tasks, recommender systems and node classification.

1. Introduction

The goal of representation learning is to embed real-world data, frequently modeled on a graph, into an ambient space. This embedding space can then be used to analyze and perform tasks on the discrete graph. The predominant approach has been to embed discrete structures in an Euclidean space. Nonetheless, data in many domains exhibit non-Euclidean features (Krioukov et al., 2010; Bronstein et al., 2017), making embeddings into Riemannian manifolds with a richer structure necessary. For this reason, embeddings into hyperbolic (Krioukov et al., 2009; Nickel & Kiela, 2017; Sala et al., 2018; López & Strube, 2020) and spherical spaces (Wilson et al., 2014; Liu et al., 2017; Xu & Durrett, 2018) have been developed. Recent work proposes to combine different curvatures through several layers (Chami et al.,

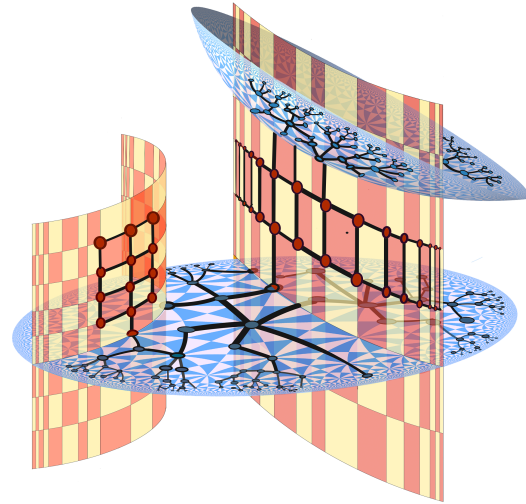


Figure 1. Symmetric spaces have a rich structure of totally geodesic subspaces, including flat subspaces (orange) and hyperbolic planes (blue). This compound, yet computationally tractable geometry allows isometric embeddings of many graphs, including those with subgraphs of dissimilar geometry. For example the graph embedded in the picture has both trees and grids as subgraphs.

2019; Bachmann et al., 2020; Grattarola et al., 2020), to enrich the geometry by considering Cartesian products of spaces (Gu et al., 2019; Tifrea et al., 2019; Skopek et al., 2020), or to use Grassmannian manifolds or the space of symmetric positive definite matrices (SPD) as a trade-off between the representation capability and the computational tractability of the space (Huang & Gool, 2017; Huang et al., 2018; Cruceru et al., 2020). A unified framework in which to encompass these various examples is still missing.

In this work, we propose the systematic use of symmetric spaces in representation learning: this is a class comprising all the aforementioned spaces. Symmetric spaces are Riemannian manifolds with rich symmetry groups which makes them algorithmically tractable. They have a compound geometry that simultaneously contains Euclidean as well as hyperbolic or spherical subspaces, allowing them to automatically adapt to dissimilar features in the graph. We develop a general framework to choose a Riemannian symmetric space and implement the mathematical tools required to learn graph embeddings (§2). Our systematic view

¹Heidelberg Institute for Theoretical Studies, Heidelberg, Germany ²Mathematical Institute, Heidelberg University, Heidelberg, Germany ³Department of Mathematics, Stanford University, California, USA. Correspondence to: Federico López <federico.lopez@h-its.org>.

enables us to introduce the use of Finsler metrics integrated with a Riemannian optimization scheme as a new method to achieve graph representations. Moreover, we use a vector-valued distance function on symmetric spaces to develop a new tool for the analysis of the structural properties of the embedded graphs.

To demonstrate a concrete implementation of our general framework, we choose Siegel spaces (Siegel, 1943); a family of symmetric spaces that has not been explored in geometric deep learning, despite them being among the most versatile symmetric spaces of non-positive curvature. Key features of Siegel spaces are that they are matrix versions of the hyperbolic plane, they contain many products of hyperbolic planes as well as copies of SPD as submanifolds, and they support Finsler metrics that induce the ℓ^1 or the ℓ^∞ metric on the Euclidean subspaces. As we verify in experiments, these metrics are well suited to embed graphs of mixed geometric features. This makes Siegel spaces with Finsler metrics an excellent device for embedding complex networks without a priori knowledge of their internal structure.

Siegel spaces are realized as spaces of symmetric matrices with coefficients in the complex numbers \mathbb{C} . By combining their explicit models and the general structure theory of symmetric spaces with the Takagi factorization (Takagi, 1924) and the Cayley transform (Cayley, 1846), we achieve a tractable and automatic-differentiable algorithm to compute distances in Siegel spaces (§4). This allows us to learn embeddings through Riemannian optimization (Bonnabel, 2011), which is easily parallelizable and scales to large datasets. Moreover, we highlight the properties of the Finsler metrics on these spaces (§3) and integrate them with the Riemannian optimization tools.

We evaluate the representation capacities of the Siegel spaces for the task of graph reconstruction on real and synthetic datasets. We find that Siegel spaces endowed with Finsler metrics outperform Euclidean, hyperbolic, Cartesian products of these spaces and SPD in all analyzed datasets. These results manifest the effectiveness and versatility of the proposed approach, particularly for graphs with varying and intricate structures.

To showcase potential applications of our approach in different graph embedding pipelines, we also test its capabilities for recommender systems and node classification. We find that our models surpass competitive baselines (constant-curvature, products thereof and SPD) for several real-world datasets.

Related Work: Riemannian manifold learning has regained attention due to appealing geometric properties that allow methods to represent non-Euclidean data arising in several domains (Rubin-Delanchy, 2020). Our systematic approach to symmetric spaces comprises embeddings in hyperbolic

spaces (Chamberlain et al., 2017; Ganea et al., 2018; Nickel & Kiela, 2018; López et al., 2019), spherical spaces (Meng et al., 2019; Defferrard et al., 2020), combinations thereof (Bachmann et al., 2020; Grattarola et al., 2020; Law & Stam, 2020), Cartesian products of spaces (Gu et al., 2019; Tifrea et al., 2019), Grassmannian manifolds (Huang et al., 2018) and the space of symmetric positive definite matrices (SPD) (Huang & Gool, 2017; Cruceru et al., 2020), among others. We implement our method on Siegel spaces. To the best of our knowledge, we are the first work to apply them in Geometric Deep Learning.

Our general view allows us to endow Riemannian symmetric spaces with Finsler metrics, which have been applied in compressed sensing (Donoho & Tsai, 2008), for clustering categorical distributions (Nielsen & Sun, 2019), and in robotics (Ratliff et al., 2020). We provide strong experimental evidence that supports the intuition on how they offer a less distorted representation than Euclidean metrics for graphs with different structure. With regard to optimization, we derive the explicit formulations to employ a generalization of stochastic gradient descent (Bonnabel, 2011) as a Riemannian adaptive optimization method (Bécigneul & Ganea, 2019).

2. Symmetric Spaces for Embedding Problems

Riemannian symmetric spaces (RSS) are Riemannian manifolds with large symmetry groups, which makes them amenable to analytical tools as well as to explicit computations. A key feature of (non-compact) RSS is that they offer a rich combination of geometric features, including many subspaces isometric to Euclidean, hyperbolic spaces and products thereof. This makes them an excellent target tool for learning embeddings of complex networks without a priori knowledge of their internal structure.

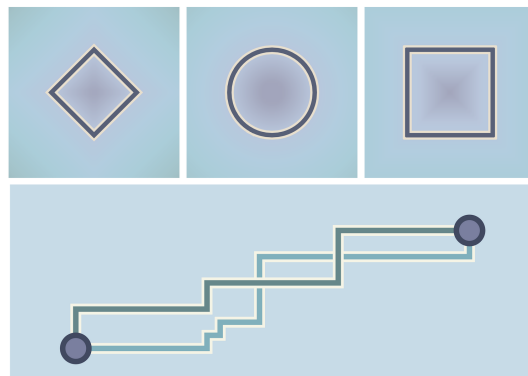


Figure 2. Above, from left to right: the unit spheres for the ℓ^1 , ℓ^2 (Euclidean), and ℓ^∞ metrics on the plane. Below: Distance minimizing geodesics are not necessarily unique in Finsler geometry. The two paths shown have the same (minimal) ℓ^1 length.

First, we introduce two aspects of the general theory of RSS to representation learning: Finsler distances and vector-valued distances. These give us, respectively, a concrete method to obtain better graph representations, and a new tool to analyze graph embeddings. Then, we describe our general implementation framework for RSS.

Finsler Distances: Riemannian metrics are not well adapted to represent graphs. For example, though a two dimensional grid intuitively looks like a plane, any embedding of it in the Euclidean plane \mathbb{R}^2 necessarily distorts some distances by a factor of at least $\sqrt{2}$. This is due to the fact that while in the Euclidean plane length minimizing paths (geodesics) are unique, in graphs there are generally several shortest paths (see Figure 2). Instead, it is possible to find an abstract isometric embedding of the grid in \mathbb{R}^2 if the latter is endowed with the ℓ^1 (or taxicab) metric. This is a first example of a Finsler distance. Another Finsler distance on \mathbb{R}^n that plays a role in our work is the ℓ^∞ metric. See Appendix A.4 for a brief introduction.

RSS do not only support a Riemannian metric, but a whole family of Finsler distances with the same symmetry group (group of isometries). For the reasons explained above, these Finsler metrics are more suitable to embed complex networks. We verify these assumptions through concrete experiments in Section 5. Since Finsler metrics are in general not convex, they are less suitable for optimization problems. Due to this, we propose to combine the Riemannian and Finsler structure, by using a Riemannian optimization scheme, with loss functions based on the Finsler metric.

Vector-valued Distance: In Euclidean space, in the sphere or in hyperbolic space, the only invariant of two points is their distance. A pair of points can be mapped to any other pair of points iff their *distance* is the same. Instead, in a general RSS the invariant between two points is a *distance vector* in \mathbb{R}^n , where n is the rank of the RSS. This is, two pairs of points can be separated by the same distance, but have different *distance vectors*. This vector-valued distance gives us a new tool to analyze graph embeddings, as we illustrate in Section 6.

The dimension of the space in which the vector-valued distance takes values in defines the *rank* of the RSS. Geometrically, this represents the largest Euclidean subspace which can be isometrically embedded (hence, hyperbolic and spherical spaces are of rank -1). The symmetries of an RSS fixing such a *maximal flat* form a finite group — the Weyl group of the RSS. In the example of Siegel spaces discussed below, the Weyl group acts by permutations and reflections of the coordinates, allowing us to canonically represent each vector-valued distance as an n -tuple of non-increasing positive numbers. Such a uniform choice of standard representative for all vector-valued distances is a fundamental domain for this group action, known as a *Weyl*

Toolkit 1 Computing Distances

- 1: **Input from Model:** Choice of basepoint m , maximal flat F , identification $\phi: F \rightarrow \mathbb{R}^n$, choice of Weyl Chamber $C \subset \mathbb{R}^n$, and Finsler norm $\|\cdot\|_F$ on \mathbb{R}^n .
 - 2: Given $p, q \in M$:
 - 3: Compute $g \in G$ such that $g(p) = m$ and $g(q) \in F$.
 - 4: Compute $v' = \phi(g(q)) \in \mathbb{R}^n$, and $h \in G$ the Weyl group element such that $h(v') = v \in C$.
 - 5: The **Vector-valued Distance (VVD)** is $\text{vDist}(p, q) = v$.
 - 6: The **Riemannian Distance (RD)** is $d^R(p, q) = \sqrt{\sum_i v_i^2}$.
 - 7: The **Finsler Distance (FD)** is $d^F(p, q) = \|v\|_F$.
 - 8: **For a product** $\prod M_i$, the VVD is the vector $(\text{vDist}(p_i, q_i))$ of VVDs for each M_i . The RD, FD satisfy the pythagorean theorem: $d^X(p, q)^2 = \sum_i d^{X_i}(p_i, q_i)^2$, for $X \in \{R, F\}$.
-

chamber for the RSS.

Implementation Schema: The general theory of RSS not only unifies many spaces previously applied in representation learning, but also systematises their implementation. Using standard tools of this theory, we provide a general framework to implement the mathematical methods required to learn graph embeddings in a given RSS.

Step 1, choosing an RSS: We may utilize the classical theory of symmetric spaces to inform our choice of RSS. Every symmetric space M can be decomposed into an (almost) product $M = M_1 \times \dots \times M_k$ of irreducible symmetric spaces. Apart from twelve exceptional examples, there are eleven infinite families irreducible symmetric spaces — see Helgason (1978) for more details, or Appendix A, Table 6. Each family of irreducible symmetric space has a distinct family of symmetry groups, which in turn determines many mathematical properties of interest (for instance, the symmetry group determines the shape of the Weyl chambers, which determines the admissible Finsler metrics). Given a geometric property of interest, the theory of RSS allows one to determine which (if any) symmetric spaces enjoy it. For example, we choose Siegel spaces also because they admit Finsler metrics induced by the ℓ^1 metric on flats, which agrees with the intrinsic metric on grid-like graphs.

Step 2, choosing a model of the RSS: Having selected an RSS, we must also select a model: a space M representing its points equipped with an action of its symmetry group G . Such a choice is of practical, rather than theoretical concern: the points of M should be easy to work with, and the symmetries of G straightforward to compute and apply. Each RSS may have many already-understood models in the literature to select from. In our example of Siegel spaces, we implement two distinct models, selected because both their points and symmetries may be encoded by $n \times n$ matrices. See Section 3.

Implementing a product of symmetric spaces requires implementing each factor simultaneously. Given models

Toolkit 2 Computing Local Geometry

- 1: **Input From Model:** Geodesic reflections $\sigma_p \in G$, the metric tensor $\langle \cdot, \cdot \rangle$, basepoint $m \in M$, orthogonal decomposition $\text{stab}(m) \oplus \mathfrak{p} = \mathfrak{g}$, and identification $\phi: T_m M \rightarrow \mathfrak{p}$.
- 2: Given $f: M \rightarrow \mathbb{R}$, a geodesic γ , or $v \in T_m M$ respectively:
- 3: The **Riemannian Gradient** of f is computed from the metric tensor by solving $\langle \text{grad}_R(f), - \rangle = df(-)$
- 4: **Parallel Transport** along γ is achieved by the differentials $(d\tau_t)_{\gamma(t_0)}$ of transvections $\tau_t = \sigma_{\gamma(t/2)} \sigma_{\gamma(t_0)}$ along γ .
- 5: The **Riemannian Exponential** $\exp_m^R(v) = g(m)$ is the matrix exponential $g = \exp(\phi(v)) \in G$ applied to m .
- 6: **For a product** $\prod M_i$ the Riemannian gradient, Parallel Transport, and Exponential map are computed component-wise.

M_1, \dots, M_k with symmetry groups G_1, \dots, G_k , the product $M = M_1 \times \dots \times M_k$ has as its points $m = (m_1, \dots, m_k)$ the k -tuples with $m_i \in M_i$, with the group $G = G_1 \times \dots \times G_k$ acting componentwise. This general implementation of products directly generalizes products of constant curvature spaces.

Step 3, computing distances: Given a choice of RSS, the fundamental quantity to compute is a distance function on M , typically used in the loss function. In contrast to general Riemannian manifolds, the rich symmetry of RSS allows this computation to be factored into a sequence of geometric steps. See Toolkit 1 for a schematic implementation using data from the standard theory of RSS (choice of maximal flat, Weyl chamber, and Finsler norm) and Algorithm 1 for a concrete implementation in the Siegel spaces.

Step 4, computing gradients: To perform gradient-based optimization, the Riemannian gradient of these distance functions is required. Depending on the Riemannian optimization methods used, additional local geometry including parallel transport and the exponential map may be useful (Bonnabel, 2011; Bécigneul & Ganea, 2019). See Toolkit 2 for the relationships of these components to elements of the classical theory of RSS.

See Appendix A and B for a review of the general theory relevant to this schema, and for an explicit implementation in the Siegel spaces.

3. Siegel Space

We implement the general aspects of the theory of RSS outlined above in the Siegel spaces HypSPD_n (Siegel, 1943), a versatile family of non-compact RSS, which has not yet been explored in geometric deep learning. The simplicity and the versatility of the Siegel space make it particularly suited for representation learning. We highlight some of its main features.

Models: HypSPD_n admits concrete and tractable matrix models generalizing the Poincaré disk and the upper half

plane model of the hyperbolic space. Both are open subsets of the space $\text{Sym}(n, \mathbb{C})$ of symmetric $n \times n$ -matrices over \mathbb{C} . HypSPD_n has $n(n+1)$ dimensions.

The bounded symmetric domain model for HypSPD_n generalizes the Poincaré disk. It is given by:¹

$$\mathcal{B}_n := \{Z \in \text{Sym}(n, \mathbb{C}) \mid \text{Id} - Z^*Z \gg 0\}; \quad (1)$$

The Siegel upper half space model for HypSPD_n generalizes the upper half plane model of the hyperbolic plane by:

$$\mathcal{S}_n := \{Z = X + iY \in \text{Sym}(n, \mathbb{C}) \mid Y \gg 0\}. \quad (2)$$

An explicit isomorphism from \mathcal{B}_n to \mathcal{S}_n is given by the Cayley transform, a matrix analogue of the familiar map from the Poincaré disk to upper half space model of the hyperbolic plane:

$$Z \mapsto i(Z + \text{Id})(Z - \text{Id})^{-1}.$$

Hyperbolic Plane over SPD: The Siegel space HypSPD_n contains SPD_n as a totally geodesic submanifold, and in fact, it can be considered as a hyperbolic plane over SPD . The role that real lines play in the hyperbolic plane, in HypSPD_n is played by SPD_n . This is illustrated in Figure 3b.

Totally Geodesic Subspaces: The Siegel space HypSPD_n contains n -dimensional Euclidean subspaces, products of n -copies of hyperbolic planes, SPD_n as well as products of Euclidean and hyperbolic spaces as totally geodesic subspaces (see Figure 3). It thus has a richer pattern of submanifolds than, for example, SPD . In particular, HypSPD_n contains more products of hyperbolic planes than SPD_n : in HypSPD_n we need 6 real dimension to contain $\mathbb{H}^2 \times \mathbb{H}^2$ and 12 real dimension to contain $(\mathbb{H}^2)^3$, whereas in SPD_n we would need 9 (resp. 20) dimensions for this.

Finsler Metrics: The Siegel space supports a Finsler metric F_1 that induces the ℓ^1 metric on the Euclidean subspaces. As already remarked, the ℓ^1 metric is particularly suitable for representing product graphs, or graphs that contain product subgraphs. Among all possible Finsler metrics supported by HypSPD_n , we focus on F_1 and F_∞ (the latter induces the ℓ^∞ metric on the flat).

Scalability: Like all RSS, HypSPD_n has a *dual* – an RSS with similar mathematical properties but reversed curvature – generalizing the duality of \mathbb{H}^2 and \mathbb{S}^2 . We focus on HypSPD_n over its dual for scalability reasons. The dual is a nonnegatively curved RSS of finite diameter, and thus does not admit isometric embeddings of arbitrarily large graphs. HypSPD_n , being nonpositively curved and infinite diameter, does not suffer from this restriction. See Appendix B.10 for details on its implementation and experiments with the dual.

¹For a real symmetric matrix $Y \in \text{Sym}(n, \mathbb{R})$ we write $Y \gg 0$ to indicate that Y is positive definite.

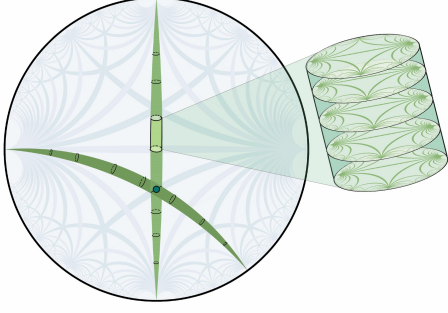
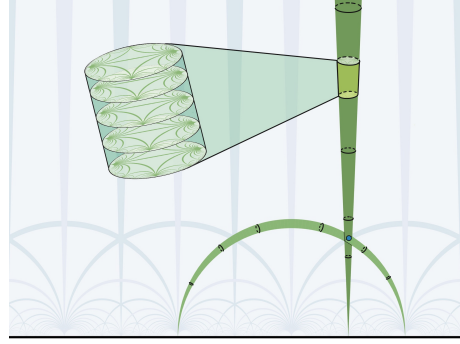

 (a) Bounded Domain Model \mathcal{B}_n

 (b) Siegel Upper Half Space \mathcal{S}_n

Figure 3. a) Every point of the disk is a complex symmetric n -dimensional matrix. b) A hyperbolic plane over SPD. \mathcal{S}_2 is a 6 dimensional manifold, the green lines represent totally geodesic submanifolds isometric to SPD that intersect in exactly one point. In dimension 2, SPD is isometric to the product of a hyperbolic plane and the line

4. Implementation

A complex number $z \in \mathbb{C}$ can be written as $z = x + iy$ where $x, y \in \mathbb{R}$ and $i^2 = -1$. Analogously a complex symmetric matrix $Z \in \text{Sym}(n, \mathbb{C})$ can be written as $Z = X + iY$, where $X = \Re(Z), Y = \Im(Z) \in \text{Sym}(n, \mathbb{R})$ are symmetric matrices with real entries. We denote by $Z^* = X - iY$ the complex conjugate matrix.

Distance Functions: To compute distances we apply either Riemannian or Finsler distance functions to the vector-valued distance. These computations are described in Algorithm 1, which is a concrete implementation of Toolkit 1. Specifically, step 2 moves one point to the basepoint, step 4 moves the other into our chosen flat, step 5 identifies this with \mathbb{R}^n and step 6 returns the vector-valued distance, from which all distances are computed. We employ the Takagi factorization to obtain eigenvalues and eigenvectors of complex symmetric matrices in a tractable manner with automatic differentiation tools (see Appendix B.2).

Complexity of Distance Algorithm: Calculating distance between two points Z_1, Z_2 in either \mathcal{S}_n or \mathcal{B}_n spaces implies computing multiplications, inversions and diagonalizations

of $n \times n$ matrices. We find that the cost of the distance computation with respect to the matrix dimensions is $\mathcal{O}(n^3)$. We prove this in Appendix D.

Riemannian Optimization with Finsler Distances: With the proposed matrix models of the Siegel space, we optimize objectives based on the Riemannian or Finsler distance functions in the embeddings space. To overcome the lack of convexity of Finsler metrics, we combine the Riemannian and the Finsler structure, by using a Riemannian optimization scheme (Bonnabel, 2011) with a loss function based on the Finsler metric. In Algorithm 2 we provide a way to compute the Riemannian gradient from the Euclidean gradient obtained via automatic differentiation. This is a direct implementation of Toolkit 2 Item 3.

To constrain the embeddings to remain within the Siegel space, we utilize a projection from the ambient space to our model. More precisely, given ϵ and a point $Z \in \text{Sym}(n, \mathbb{C})$, we compute a point $Z_\epsilon^{\mathcal{S}}$ (resp. $Z_\epsilon^{\mathcal{B}}$) close to the original point lying in the ϵ -interior of the model. For \mathcal{S}_n , starting from $Z = X + iY$ we orthogonally diagonalize $Y = K^t D K$, and then modify $D = \text{diag}(d_i)$ by setting each diagonal entry to $\max\{d_i, \epsilon\}$. An analogous projection is defined on the bounded domain \mathcal{B}_n , see Appendix B.8.

Algorithm 1 Computing Distances

- 1: Given two points $Z_1, Z_2 \in \mathcal{S}_n$:
 - 2: Define $Z_3 = \sqrt{\Im(Z_1)^{-1}}(Z_2 - \Re(Z_1))\sqrt{\Im(Z_1)^{-1}} \in \mathcal{S}_n$
 - 3: Define $W = (Z_3 - i\text{Id})(Z_3 + i\text{Id})^{-1} \in \mathcal{B}_n$
 - 4: Use the Takagi factorization to write $W = \bar{K} D K^*$ for D real diagonal, and K unitary.
 - 5: Define $v_i = \log \frac{1+d_i}{1-d_i}$ for d_i the diagonal entries of D .
 - 6: Order the v_i so that $v_1 \geq v_2 \geq \dots \geq 0$. The **Vector-valued Distance** is $\text{vDist}(Z_1, Z_2) = (v_1, v_2, \dots, v_n)$.
 - 7: The **Riemannian distance** is $d^R(Z_1, Z_2) := \sqrt{\sum_{i=1}^n v_i^2}$.
 - 8: The **Finsler distance inducing the ℓ^1 -metric** is $d^{F^1}(Z_1, Z_2) := \sum_{i=1}^n v_i$.
 - 9: The **Finsler distance inducing the ℓ^∞ -metric** is $d^{F^\infty}(Z_1, Z_2) := \max\{v_i\} = v_1$.
-

Algorithm 2 Computing Riemannian Gradient

- 1: Given $f : \mathcal{S}_n \rightarrow \mathbb{R}$ and $Z = X + iY \in \mathcal{S}_n$:
 - 2: Compute the Euclidean gradient $\text{grad}_E(f)$ at Z of f obtained via automatic differentiation (see Appendix B.6).
 - 3: The **Riemannian gradient** is $\text{grad}_R(f) = Y \cdot \text{grad}_E(f) \cdot Y$.
-

Symmetric Spaces for Graph Embeddings

(V , E)	4D GRID (625, 2000)		TREE (364, 363)		TREE \times GRID (496, 1224)		TREE \times TREE (225, 420)		TREE \diamond GRIDS (775, 1270)		GRID \diamond TREES (775, 790)	
	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
\mathbb{E}^{20}	11.24 \pm 0.00	100.00	3.92 \pm 0.04	42.30	9.81 \pm 0.00	83.32	9.78 \pm 0.00	96.03	3.86 \pm 0.02	34.21	4.28 \pm 0.04	27.50
\mathbb{H}^{20}	25.23 \pm 0.05	63.74	0.54\pm0.02	100.00	17.21 \pm 0.21	83.16	20.59 \pm 0.11	75.67	14.56 \pm 0.27	44.14	14.62 \pm 0.13	30.28
$\mathbb{E}^{10} \times \mathbb{H}^{10}$	11.24 \pm 0.00	100.00	1.19 \pm 0.04	100.00	9.20 \pm 0.01	100.00	9.30 \pm 0.04	98.03	2.15 \pm 0.05	58.23	2.03 \pm 0.01	97.88
$\mathbb{H}^{10} \times \mathbb{H}^{10}$	18.74 \pm 0.01	78.47	0.65 \pm 0.02	100.00	13.02 \pm 0.91	88.01	8.61 \pm 0.03	97.63	1.08 \pm 0.06	77.20	2.80 \pm 0.65	84.88
SPD ₆	11.24 \pm 0.00	100.00	1.79 \pm 0.02	55.92	9.23 \pm 0.01	99.73	8.83 \pm 0.01	98.49	1.56 \pm 0.02	62.31	1.83 \pm 0.00	72.17
\mathcal{S}_4^R	11.27 \pm 0.01	100.00	1.35 \pm 0.02	78.53	9.13 \pm 0.01	99.92	8.68 \pm 0.02	98.03	1.45 \pm 0.09	72.49	1.54 \pm 0.08	76.66
$\mathcal{S}_4^{F_\infty}$	5.92 \pm 0.06	99.61	1.23 \pm 0.28	99.56	4.81 \pm 0.55	99.28	3.31 \pm 0.06	99.95	10.88 \pm 0.19	63.52	10.48 \pm 0.21	72.53
$\mathcal{S}_4^{F_1}$	0.01\pm0.00	100.00	0.76 \pm 0.02	91.57	0.81\pm0.08	100.00	1.08\pm0.16	100.00	1.03\pm0.00	78.71	0.84\pm0.06	80.52
\mathcal{B}_4^R	11.28 \pm 0.01	100.00	1.27 \pm 0.05	74.77	9.24 \pm 0.13	99.22	8.74 \pm 0.09	98.12	2.88 \pm 0.32	72.55	2.76 \pm 0.11	96.29
$\mathcal{B}_4^{F_\infty}$	7.32 \pm 0.16	97.92	1.51 \pm 0.13	99.73	8.70 \pm 0.87	96.40	4.26 \pm 0.26	99.70	6.55 \pm 1.77	73.80	7.15 \pm 0.85	90.51
$\mathcal{B}_4^{F_1}$	0.39 \pm 0.02	100.00	0.77 \pm 0.02	94.64	0.90 \pm 0.08	100.00	1.28 \pm 0.16	100.00	1.09 \pm 0.03	76.55	0.99 \pm 0.01	81.82

Table 1. Results for synthetic datasets. Lower D_{avg} is better. Higher mAP is better. Metrics are given as percentage.

5. Graph Reconstruction

We evaluate the representation capabilities of the proposed approach for the task of graph reconstruction.²

Setup: We embed graph nodes in a transductive setting. As input and evaluation data we take the shortest distance in the graph between every pair of connected nodes. Unlike previous work (Gu et al., 2019; Cruceru et al., 2020) we do not apply any scaling, neither in the input graph distances nor in the distances calculated on the space. We experiment with the loss proposed in Gu et al. (2019), which minimizes the relation between the distance in the space, compared to the distance in the graph, and captures the average distortion. We initialize the matrix embeddings in the Siegel upper half space by adding small symmetric perturbations to the matrix basepoint iId . For the Bounded model, we additionally map the points with the Cayley transform (see Appendix B.7). In all cases we optimize with RSGD (Bonnabel, 2011) and report the average of 5 runs.

Baselines: We compare our approach to constant-curvature baselines, such as Euclidean (\mathbb{E}) and hyperbolic (\mathbb{H}) spaces (we compare to the Poincaré model (Nickel & Kiela, 2017) since the Bounded Domain model is a generalization of it), Cartesian products thereof ($\mathbb{E} \times \mathbb{H}$ and $\mathbb{H} \times \mathbb{H}$) (Gu et al., 2019), and symmetric positive definite matrices (SPD) (Cruceru et al., 2020) in low and high dimensions. Preliminary experiments on the dual of HypSPD_n and on spherical spaces showed poor performance thus we do not compare to them (see Appendix B.12). To establish a fair comparison, each model has the same number of free parameters. This is, the spaces \mathcal{S}_n and \mathcal{B}_n have $n(n+1)$ parameters, thus we compare to baselines of the same dimensionality.³ All implementations are taken from Geopt (Kochurov et al., 2020).

Metrics: Following previous work (Sala et al., 2018; Gu

²Code available at <https://github.com/fedelopez77/sympa>.

³We also consider comparable dimensionalities for SPD_n, which has $n(n+1)/2$ parameters.

et al., 2019), we measure the quality of the learned embeddings by reporting *average distortion* D_{avg} , a global metric that considers the explicit value of all distances, and *mean average precision* mAP, a ranking-based measure for local neighborhoods (local metric) as fidelity measures.

Synthetic Graphs: As a first step, we investigate the representation capabilities of different geometric spaces on synthetic graphs. Previous work has focused on graphs with pure geometric features, such as grids, trees, or their Cartesian products (Gu et al., 2019; Cruceru et al., 2020), which mix the grid- and tree-like features globally. We expand our analysis to rooted products of trees and grids. These graphs mix features at different levels and scales. Thus, they reflect to a greater extent the complexity of intertwining and varying structure in different regions, making them a better approximation of real-world datasets. We consider the rooted product TREE \diamond GRIDS of a tree and 2D grids, and GRID \diamond TREES, of a 2D grid and trees. More experimental details, hyperparameters, formulas and statistics about the data are present in Appendix C.3.

We report the results for synthetic graphs in Table 1. We find that the Siegel space with Finsler metrics significantly outperform constant curvature baselines in all graphs, except for the tree, where they have competitive results with the hyperbolic models. We observe that Siegel spaces with the Riemannian metric perform on par with the matching geometric spaces or with the best-fitting product of spaces across graphs of pure geometry (grids and Cartesian products of graphs). However, the F_1 metric outperforms the Riemannian and F_∞ metrics in all graphs, for both models. This is particularly noticeable for the 4D GRID, where the distortion achieved by F_1 models is almost null, matching the intuition of less distorted grid representations through the taxicab metric.

Even when the structure of the data conforms to the geometry of baselines, the Siegel spaces with the Finsler-Riemannian approach are able to outperform them by automatically adapting to very dissimilar patterns without any a priori estimates of the curvature or other features of the

Symmetric Spaces for Graph Embeddings

(V , E)	USCA312 (312, 48516)	BIO-DISEASOME (516, 1188)		CSPHD (1025, 1043)		EUROROAD (1039, 1305)		FACEBOOK (4039, 88234)	
	D_{avg}	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
\mathbb{E}^{20}	0.18±0.01	3.83±0.01	76.31	4.04±0.01	47.37	4.50±0.00	87.70	3.16±0.01	32.21
\mathbb{H}^{20}	2.39±0.02	6.83±0.08	91.26	22.42±0.23	60.24	43.56±0.44	54.25	3.72±0.00	44.85
$\mathbb{E}^{10} \times \mathbb{H}^{10}$	0.18±0.00	2.52±0.02	91.99	3.06±0.02	73.25	4.24±0.02	89.93	2.80±0.01	34.26
$\mathbb{H}^{10} \times \mathbb{H}^{10}$	0.47±0.18	2.57±0.05	95.00	7.02±1.07	79.22	23.30±1.62	75.07	2.51±0.00	36.39
SPD ₆	0.21±0.02	2.54±0.00	82.66	2.92±0.11	57.88	19.54±0.99	92.38	2.92±0.05	33.73
S_4^R	0.28±0.03	2.40±0.02	87.01	4.30±0.18	59.95	29.21±0.91	84.92	3.07±0.04	30.98
$S_4^{F_\infty}$	0.57±0.08	2.78±0.49	93.95	27.27±1.00	59.45	46.82±1.02	72.03	1.90±0.11	45.58
$S_4^{F_1}$	0.18±0.02	1.55±0.04	90.42	1.50±0.03	64.11	3.79±0.07	94.63	2.37±0.07	35.23
B_4^R	0.24±0.07	2.69±0.10	89.11	28.65±3.39	62.66	53.45±2.65	48.75	3.58±0.10	30.35
$B_4^{F_\infty}$	0.21±0.04	4.58±0.63	90.36	26.32±6.16	54.94	52.69±2.28	48.75	2.18±0.18	39.15
$B_4^{F_1}$	0.18±0.07	1.54±0.02	90.41	2.96±0.91	67.58	21.98±0.62	91.63	5.05±0.03	39.87

Table 2. Results for real-world datasets. Lower D_{avg} is better. Higher mAP is better. Metrics are given as percentage.

graph. This showcases the flexibility of our models, due to its enhanced geometry and higher expressivity.

For graphs with mixed geometric features (rooted products), Cartesian products of spaces cannot arrange these compound geometries into separate Euclidean and hyperbolic subspaces. RSS, on the other hand, offer a less distorted representation of these tangled patterns by exploiting their richer geometry which mixes hyperbolic and Euclidean features. Moreover, they reach a competitive performance on the local neighborhood reconstruction, as the mean precision shows. Results for more dimensionalities are given in Appendix F.

Real-world Datasets: We compare the models on two road networks, namely USCA312 of distances between North American cities and EUROROAD between European cities, BIO-DISEASOME, a network of human disorders and diseases with reference to their genetic origins (Goh et al., 2007), a graph of computer science Ph.D. advisor-advisee relationships (Nooy et al., 2011), and a dense social network from Facebook (McAuley & Leskovec, 2012). These graphs have been analyzed in previous work as well (Gu et al., 2019; Cruceru et al., 2020).

We report the results in Table 2. On the USCA312 dataset, which is the only weighted graph under consideration, the

Siegel spaces perform on par with the compared target manifolds. For all other datasets, the model with Finsler metrics outperforms all baselines. In line with the results for synthetic datasets, the F_1 metric exhibits an outstanding performance across several datasets.

Overall, these results show the strong reconstruction capabilities of RSS for real-world data as well. It also indicates that vertices in these real-world dataset form networks with a more intricate geometry, which the Siegel space is able to unfold to a better extent.

High-dimensional Spaces: In Table 3 we compare the approach in high-dimensional spaces (rank 17 which is equal to 306 free parameters), also including spherical spaces \mathbb{S} . The results show that our models operate well with larger matrices, where we see further improvement in our distortion and mean average precision over the low dimensional spaces of rank 4. We observe that even though we notably increase the dimensions of the baselines to 306, the Siegel models of rank 4 (equivalent to 20 dimensions) significantly outperform them. These results match the expectation that the richer variable curvature geometry of RSS better adapts to graphs with intricate geometric structures.

6. Analysis of the Embedding Space

One reason to embed graphs into Riemannian manifolds is to use geometric properties of the manifold to analyze

	TREE \times GRID		GRID \diamond TREES		BIO-DISEASOME	
	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
S_4^R	9.13	99.92	1.54	76.66	2.40	87.01
$S_4^{F_\infty}$	4.81	99.28	10.48	72.53	2.78	93.95
$S_4^{F_1}$	<u>0.81</u>	100.00	<u>0.84</u>	80.52	1.55	90.42
\mathbb{E}^{306}	9.80	85.14	2.81	67.69	3.52	88.45
\mathbb{H}^{306}	17.31	82.97	15.92	27.14	7.04	91.46
\mathbb{S}^{306}	73.78	35.36	81.67	58.26	70.91	84.61
$\mathbb{E}^{153} \times \mathbb{H}^{153}$	9.14	100.00	1.52	<u>97.85</u>	2.36	95.65
$\mathbb{S}^{153} \times \mathbb{S}^{153}$	60.71	6.93	70.00	5.64	55.51	19.51
S_{17}^R	9.19	99.89	1.31	75.45	2.13	93.14
$S_{17}^{F_\infty}$	4.82	97.45	11.45	94.09	<u>1.50</u>	<u>98.27</u>
$S_{17}^{F_1}$	0.03	100.00	0.27	99.23	0.73	99.09

Table 3. Results for different datasets in high-dimensional spaces. Best result is **bold**, second best underlined.

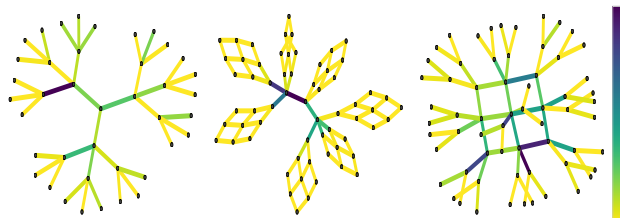


Figure 4. Edge coloring of $S_2^{F_1}$ for a tree (left), and a rooted product of TREE \diamond GRIDS (center), and of GRID \diamond TREES.

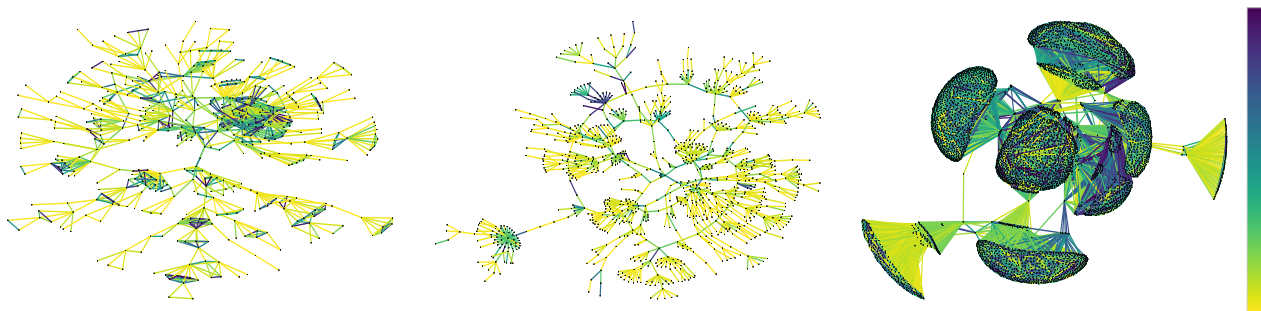


Figure 5. Edge coloring of S_2^{F1} for BIO-DISEASOME (left) and CSPHD (center) and FACEBOOK (right). Edge colors indicate the angle of the vector-valued distance for each edge, on a linear scale from 0 (yellow) to $\pi/4$ (blue).

the structure of the graph. Embeddings into hyperbolic spaces, for example, have been used to infer and visualize hierarchical structure in data sets (Nickel & Kiela, 2018). Visualizations in RSS are difficult due to their high dimensionality. As a solution we use the vector-valued distance function in the RSS to develop a new tool to visualize and to analyze structural properties of the graphs.

We focus on HypSPD_2 , the Siegel space of rank $k = 2$, where the vector-valued distance is just a vector in a cone in \mathbb{R}^2 . We take edges (Z_i, Z_j) and assign the angle of the vector $\text{vDist}(Z_i, Z_j) = (v_1, v_2)$ (see Algorithm 1, step 6) to each edge in the graph. This angle assignment provides a continuous edge coloring that can be leveraged to find structure in graphs.

We see in Figure 4 that the edge coloring makes the large-scale structure of the tree (blue/green edges) and the leaves (yellow edges) visible. This is even more striking for the rooted products. In $\text{TREE} \diamond \text{GRIDS}$ the edge coloring distinguishes the hyperbolic parts of the graph (blue edges) and the Euclidean parts (yellow edges). For the $\text{GRID} \diamond \text{TREES}$, the Euclidean parts are labelled by blue/green edges and the hyperbolic parts by yellow edges. Thus, even though we trained the embedding only on the metric, it automatically adapts to other features of the graph.

In the edge visualizations for real-world datasets (Figure 5), the edges in the denser connected parts of the graph have a higher angle, as it can be seen for the BIO-DISEASOME and FACEBOOK data sets. For CSPHD, the tree structure is emphasized by the low angles.

This suggests that the continuous values that we assign to edges are a powerful tool to automatically discover dissimilar patterns in graphs. This can be further used in efficient clustering of the graph. In Appendix E we give similar visualizations for the Riemannian metric and the F_∞ Finsler metric, showing that also with respect to exhibiting structural properties, the F_1 metric performs best.

7. Downstream Tasks

We also evaluate the representation capabilities of Siegel spaces on two downstream tasks: recommender systems and node classification.

7.1. Recommender Systems

Our method can be applied in downstream tasks that involve embedding graphs, such as recommender systems. These systems mine user-item interactions and recommend items to users according to the distance/similarity between their respective embeddings (Hsieh et al., 2017).

Setup: Given a set of observed user-item interactions $\mathcal{T} = \{(u, v)\}$, we follow a metric learning approach (Vinh Tran et al., 2020) and learn embeddings by optimizing the following hinge loss function:

$$\mathcal{L} = \sum_{(u,v) \in \mathcal{T}} \sum_{(u,w) \notin \mathcal{T}} [m + d_{\mathbb{K}}(\mathbf{u}, \mathbf{v})^2 - d_{\mathbb{K}}(\mathbf{u}, \mathbf{w})^2]_+ \quad (3)$$

where \mathbb{K} is the target space, w is an item the user has not interacted with, $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{K}$, $m > 0$ is the hinge margin and $[z]_+ = \max(0, z)$. To generate recommendations, for each user u we rank the items \mathbf{v}_i according to their distance to \mathbf{u} . Since it is very costly to rank all the available items, we randomly select 100 samples which the user has not interacted with, and rank the ground truth amongst these samples (He et al., 2017). We adopt normalized discounted cumulative gain (nDG) and hit ratio (HR), both at 10, as ranking evaluation metrics for recommendations. More experimental details and data stats in Appendix C.4.

Data: We evaluate the different models over two MovieLens datasets (ML-1M and ML-100K) (Harper & Konstan, 2015), LAST.FM, a dataset of artist listening records (Cantador et al., 2011), and MEETUP, crawled from Meetup.com (Pham et al., 2015). To generate evaluation splits, the penultimate and last item the user has interacted with are withheld as dev and test set respectively.

Results: We report the performance for all analyzed models in Table 4. While in the Movies datasets, the Riemannian

	ML-1M		ML-100K		LASTFM		MEETUP	
	HR@10	nDG	HR@10	nDG	HR@10	nDG	HR@10	nDG
\mathbb{E}^{20}	46.9±0.6	22.7	54.6±1.0	28.7	55.4±0.3	24.6	69.8±0.4	46.4
\mathbb{H}^{20}	46.0±0.5	23.0	53.4±1.0	28.2	54.8±0.5	24.9	71.8±0.5	48.5
$\mathbb{E}^{10} \times \mathbb{H}^{10}$	52.0±0.7	27.4	53.1±1.3	27.9	45.5±0.9	18.9	70.7±0.2	47.5
$\mathbb{H}^{10} \times \mathbb{H}^{10}$	46.7±0.6	23.0	54.8±0.9	29.1	55.0±0.9	24.6	71.7±0.1	48.8
SPD_6	45.8±1.0	22.1	53.3±1.4	28.0	55.4±0.2	25.3	70.1±0.6	46.5
S_4^R	53.8±0.3	27.7	55.7±0.9	28.6	53.1±0.5	24.8	65.8±1.2	43.4
$S_4^{F_\infty}$	45.9±0.9	22.7	52.5±0.3	27.5	53.8±1.7	32.5	69.0±0.5	46.4
$S_4^{F_1}$	52.9±0.6	27.2	55.6±1.3	29.4	61.1±1.2	38.0	74.9±0.1	52.8

Table 4. Results for recommender system datasets.

model marginally outperforms the baselines, in the other two cases the F_1 model achieves the highest performance by a larger difference. These systems learn to model users’ preferences, and embeds users and items in the space, in a way that is exploited for the task of generating recommendations. In this manner we demonstrate how downstream tasks can profit from the enhanced graph representation capacity of our models, and we highlight the flexibility of the method, in this case applied in combination with a collaborative metric learning approach (Hsieh et al., 2017).

7.2. Node Classification

Our proposed graph embeddings can be used in conjunction with standard machine learning pipelines, such as downstream classification. To demonstrate this, and following the procedure of Chami et al. (2020), we embed three hierarchical clustering datasets based on the cosine distance between their points, and then use the learned embeddings as input features for a Euclidean logistic regression model. Since the node embeddings lie in different metric spaces, we apply the corresponding logarithmic map to obtain a “flat” representation before classifying. For the Siegel models of dimension n , we first map each complex matrix embedding $Z = X + iY$ to $[(Y + XY^{-1}X, XY^{-1}), (Y^{-1}X, Y^{-1})] \in SPD_{2n}$, this is the natural realisation of $HypSPD_n$ as a totally geodesic submanifold of SPD_{2n} , and then we apply the LogEig map (Huang & Gool, 2017), which yields a representation in a flat space. More experimental details in Appendix C.5.

Results are presented in Table 5. In all cases we see that the embeddings learned by our models capture the structural properties of the dataset, so that a simple classifier can separate the nodes into different clusters. They offer the best

Dataset	IRIS	ZOO	GLASS
\mathbb{E}^{20}	83.3±1.1	88.7±1.8	67.2±2.5
\mathbb{H}^{20}	84.0±0.6	87.3±1.5	62.8±2.0
$\mathbb{E}^{10} \times \mathbb{H}^{10}$	85.6±1.1	88.0±1.4	64.8±4.3
$\mathbb{H}^{10} \times \mathbb{H}^{10}$	87.8±1.4	87.3±1.5	63.4±3.4
SPD_6	88.0±1.6	88.7±2.2	66.9±2.0
S_4^R	88.0±0.5	88.7±2.2	66.6±2.4
$S_4^{F_\infty}$	89.1±0.5	88.7±2.5	65.2±3.0
$S_4^{F_1}$	89.3±1.1	90.7±1.5	67.5±3.9
B_4^R	86.0±1.9	88.7±1.4	65.5±3.1
$B_4^{F_\infty}$	84.4±0.0	87.3±1.9	65.6±1.7
$B_4^{F_1}$	85.6±1.4	89.3±2.8	64.2±1.7

Table 5. Accuracy for node classification based on its embedding.

performance in the three datasets. This suggests that embeddings in Siegel spaces learn meaningful representations that can be exploited into downstream tasks. Moreover, we showcase how to map these embeddings to “flat” vectors; in this way they can be integrated with classical Euclidean network layers.

8. Conclusions & Future Work

Riemannian manifold learning has regained attention due to appealing geometric properties that allow methods to represent non-Euclidean data arising in several domains. We propose the systematic use of symmetric spaces to encompass previous work in representation learning, and develop a toolkit that allows practitioners to choose a Riemannian symmetric space and implement the mathematical tools required to learn graph embeddings. We introduce the use of Finsler metrics integrated with a Riemannian optimization scheme, which provide a significantly less distorted representation over several data sets. As a new tool to discover structure in the graph, we leverage the vector-valued distance function on a RSS. We implement these ideas on Siegel spaces, a rich class of RSS that had not been explored in geometric deep learning, and we develop tractable and mathematically sound algorithms to learn embeddings in these spaces through gradient-descent methods. We showcase the effectiveness of the proposed approach on conventional as well as new datasets for the graph reconstruction task, and in two downstream tasks. Our method ties or outperforms constant-curvature baselines without requiring any previous assumption on geometric features of the graphs. This shows the flexibility and enhanced representation capacity of Siegel spaces, as well as the versatility of our approach.

As future directions, we consider applying the vector-valued distance in clustering and structural analysis of graphs, and the development of deep neural network architectures adapted to the geometry of RSS, specifically Siegel spaces. A further interesting research direction is to use geometric transition between symmetric spaces to extend the approach demonstrated by curvature learning à la Gu et al. (2019). We plan to leverage the structure of the Siegel space of a hyperbolic plane over SPD to analyze medical imaging data, which is often given as symmetric positive definite matrices, see Pennec (2020).

Acknowledgements

This work has been supported by the German Research Foundation (DFG) as part of the Research Training Group AIPHES under grant No. GRK 1994/1, as well as under Germany’s Excellence Strategy EXC-2181/1 - 390900948 (the Heidelberg STRUCTURES Cluster of Excellence), and by the Klaus Tschira Foundation, Heidelberg, Germany.

References

- Bachmann, G., Bécigneul, G., and Ganea, O.-E. Constant curvature graph convolutional networks. In *37th International Conference on Machine Learning (ICML)*, 2020.
- Bécigneul, G. and Ganea, O.-E. Riemannian adaptive optimization methods. In *7th International Conference on Learning Representations, ICLR*, New Orleans, LA, USA, May 2019. URL <https://openreview.net/forum?id=r1eiqi09K7>.
- Boland, J. and Newberger, F. Minimal entropy rigidity for Finsler manifolds of negative flag curvature. *Ergodic Theory and Dynamical Systems*, 21(1):13–23, 2001. doi: 10.1017/S0143385701001055.
- Bonnabel, S. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58, 11 2011. doi: 10.1109/TAC.2013.2254619.
- Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34 (4):18–42, 2017.
- Cantador, I., Brusilovsky, P., and Kuflik, T. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys 2011, New York, NY, USA, 2011. ACM.
- Cayley, A. Sur quelques propriétés des déterminants gauches. *Journal für die reine und angewandte Mathematik*, 32:119–123, 1846. URL <http://www.digizeitschriften.de/dms/img/?PID=GDZPPN002145308>.
- Chamberlain, B., Deisenroth, M., and Clough, J. Neural embeddings of graphs in hyperbolic space. In *Proceedings of the 13th International Workshop on Mining and Learning with Graphs (MLG)*, 2017.
- Chami, I., Ying, Z., Ré, C., and Leskovec, J. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems 32*, pp. 4869–4880. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/0415740eaa4d9decabc8da001d3fd805f-Paper.pdf>.
- Chami, I., Gu, A., Chatziafratis, V., and Ré, C. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- Cruceru, C., Bécigneul, G., and Ganea, O.-E. Computationally tractable Riemannian manifolds for graph embeddings. In *37th International Conference on Machine Learning (ICML)*, 2020.
- Defferrard, M., Milani, M., Gusset, F., and Perraudin, N. DeepSphere: A graph-based spherical CNN. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=B1e30lStPB>.
- Donoho, D. L. and Tsaig, Y. Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse. *IEEE Trans. Information Theory*, 54(11):4789–4812, 2008.
- Dua, D. and Graff, C. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- Falkenberg, A. Method to calculate the inverse of a complex matrix using real matrix inversion. 2007.
- Ganea, O.-E., Bécigneul, G., and Hofmann, T. Hyperbolic entailment cones for learning hierarchical embeddings. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1646–1655, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/ganea18a.html>.
- Goh, K.-I., Cusick, M. E., Valle, D., Childs, B., Vidal, M., and Barabási, A.-L. The human disease network. *Proceedings of the National Academy of Sciences*, 104 (21):8685–8690, 2007.
- Grattarola, D., Zambon, D., Livi, L., and Alippi, C. Change detection in graph streams by learning graph embeddings on constant-curvature manifolds. *IEEE Trans. Neural Networks Learn. Syst.*, 31(6):1856–1869, 2020. doi: 10.1109/TNNLS.2019.2927301. URL <https://doi.org/10.1109/TNNLS.2019.2927301>.
- Gu, A., Sala, F., Gunel, B., and Ré, C. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=HJxeWnCcF7>.
- Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using NetworkX. In Varoquaux, G., Vaught, T., and Millman, J. (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.
- Harper, F. M. and Konstan, J. A. The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015. ISSN 2160-6455.

- doi: 10.1145/2827872. URL <https://doi.org/10.1145/2827872>.
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW '17*, pp. 173–182, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349130. doi: 10.1145/3038912.3052569. URL <https://doi.org/10.1145/3038912.3052569>.
- Helgason, S. *Differential geometry, Lie groups, and symmetric spaces*. Academic Press New York, 1978. ISBN 0123384605.
- Hsieh, C.-K., Yang, L., Cui, Y., Lin, T.-Y., Belongie, S., and Estrin, D. Collaborative metric learning. In *Proceedings of the 26th International Conference on World Wide Web, WWW '17*, pp. 193–201, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349130. doi: 10.1145/3038912.3052639. URL <https://doi.org/10.1145/3038912.3052639>.
- Huang, Z. and Gool, L. V. A Riemannian network for SPD matrix learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17*, pp. 2036–2042. AAAI Press, 2017.
- Huang, Z., Wu, J., and Gool, L. V. Building deep networks on Grassmann manifolds. In McIlraith, S. A. and Weinberger, K. Q. (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3279–3286. AAAI Press, 2018. URL <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16846>.
- Kochurov, M., Karimov, R., and Kozlukov, S. Geopt: Riemannian optimization in PyTorch. *ArXiv*, abs/2005.02819, 2020.
- Krioukov, D., Papadopoulos, F., Vahdat, A., and Boguñá, M. On Curvature and Temperature of Complex Networks. *Physical Review E*, 80(035101), Sep 2009.
- Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., and Boguñá, M. Hyperbolic geometry of complex networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 82:036106, 09 2010. doi: 10.1103/PhysRevE.82.036106.
- Law, M. T. and Stam, J. Ultrahyperbolic representation learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. SphereFace: Deep hypersphere embedding for face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6738–6746, 2017. doi: 10.1109/CVPR.2017.713.
- López, F. and Strube, M. A fully hyperbolic neural model for hierarchical multi-class classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 460–475, Online, November 2020. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.findings-emnlp.42>.
- López, F., Heinzerling, B., and Strube, M. Fine-grained entity typing in hyperbolic space. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepLanLP-2019)*, pp. 169–180, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4319. URL <https://www.aclweb.org/anthology/W19-4319>.
- McAuley, J. and Leskovec, J. Learning to discover social circles in ego networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12*, pp. 539–547, Red Hook, NY, USA, 2012. Curran Associates Inc.
- Meng, Y., Huang, J., Wang, G., Zhang, C., Zhuang, H., Kaplan, L., and Han, J. Spherical text embedding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 8208–8217. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/043ab21fc5a1607b381ac3896176dac6-Paper.pdf>.
- Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6341–6350. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/59dfa2df42d9e3d41f5b02bfc32229dd-Paper.pdf>.

- Nickel, M. and Kiela, D. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3779–3788, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/nickel18a.html>.
- Nielsen, F. and Sun, K. *Clustering in Hilbert’s Projective Geometry: The Case Studies of the Probability Simplex and the Elliptope of Correlation Matrices*, pp. 297–331. Springer International Publishing, Cham, 2019. ISBN 978-3-030-02520-5. doi: 10.1007/978-3-030-02520-5_11. URL https://doi.org/10.1007/978-3-030-02520-5_11.
- Nooy, W. D., Mrvar, A., and Batagelj, V. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, USA, 2011. ISBN 0521174805.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.
- Pennec, X. Manifold-Valued Image Processing with SDP Matrices. In *Riemannian Geometric Statistics in Medical Image Analysis*, pp. 75–134. Academic Press, 2020.
- Pham, T. N., Li, X., Cong, G., and Zhang, Z. A general graph-based model for recommendation in event-based social networks. In *2015 IEEE 31st International Conference on Data Engineering*, pp. 567–578, 2015. doi: 10.1109/ICDE.2015.7113315.
- Ratliff, N. D., Wyk, K. V., Xie, M., Li, A., and Rana, M. A. Generalized nonlinear and Finsler geometry for robotics. *CoRR*, abs/2010.14745, 2020. URL <https://arxiv.org/abs/2010.14745>.
- Rossi, R. A. and Ahmed, N. K. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL <http://networkrepository.com>.
- Rubin-Delanchy, P. Manifold structure in graph embeddings, 2020. URL <https://arxiv.org/abs/2006.05168>.
- Sala, F., De Sa, C., Gu, A., and Re, C. Representation tradeoffs for hyperbolic embeddings. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4460–4469, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/sala18a.html>.
- Siegel, C. L. Symplectic geometry. *American Journal of Mathematics*, 65(1):1–86, 1943. ISSN 00029327, 10806377. URL <http://www.jstor.org/stable/2371774>.
- Skopek, O., Ganea, O.-E., and Becigneul, G. Mixed-curvature variational autoencoders. In *8th International Conference on Learning Representations (ICLR)*, April 2020. URL <https://openreview.net/pdf?id=S1g6xeSKDS>.
- Takagi, T. On an algebraic problem related to an analytic theorem of carathéodory and fejér and on an allied theorem of Landau. *Japanese journal of mathematics :transactions and abstracts*, 1:83–93, 1924. doi: 10.4099/jjm1924.1.0_83.
- Tifrea, A., Bécigneul, G., and Ganea, O.-E. Poincare GloVe: Hyperbolic word embeddings. In *7th International Conference on Learning Representations, ICLR*, New Orleans, LA, USA, May 2019. URL <https://openreview.net/forum?id=Ske5r3AqK7>.
- Vinh Tran, L., Tay, Y., Zhang, S., Cong, G., and Li, X. HyperML: A boosting metric learning approach in hyperbolic space for recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM ’20*, pp. 609–617, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371850. URL <https://doi.org/10.1145/3336191.3371850>.
- Wilson, R. C., Hancock, E. R., Pekalska, E., and Duin, R. P. W. Spherical and hyperbolic embeddings of data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2255–2269, 2014.
- Xu, J. and Durrett, G. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.