Symmetric Spaces for Graph Embeddings: A Finsler-Riemannian Approach

Federico L´opez ¹  Beatrice Pozzetti ²  Steve Trettel ³  Michael Strube ¹  Anna Wienhard ²

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., 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...
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 $\ell_1$ or the $\ell_\infty$ 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 to endow Riemannian symmetric spaces with Finsler metrics, which have been applied in compressed sensing (Donoho & Tsaig, 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 $\ell_1$, $\ell_2$ (Euclidean), and $\ell_\infty$ metrics on the plane. Below: Distance minimizing geodesics are not necessarily unique in Finsler geometry. The two paths shown have the same (minimal) $\ell_1$ length.](image-url)
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 \( \ell^1 \) (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 \( \ell^\infty \) 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 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 \( M = M_1 \times \cdots \times M_k \) 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 \( \ell^1 \) 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 1 Computing Distances**

1. **Input from Model:** Choice of basepoint \( m \), maximal flat \( F \), identification \( \phi: F \to \mathbb{R}^n \), 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 \( vDist(p, q) = v \).
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 \( C \cdot M_1 \cdot \cdots \cdot M_k \), the VVD is the vector \( (vDist(p_1, q_1), \ldots, vDist(p_k, q_k)) \) of FVDs 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 \( X \in \{R, F\} \).
We implement the general aspects of the theory of RSS (Bonnabel, 2011; Bécigneul & Ganea, 2019). See Toolkit 2 models generalizing the Poincaré disk and the upper half 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 \( \ell^1 \) metric on the Euclidean subspaces. As already remarked, the \( \ell^1 \) 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 \( F_{\infty} \) (the latter induces the \( \ell^\infty \) 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.
4. Implementation

A complex number \( z = x + iy \) 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 \( S_n \) or \( B_n \) 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 \( O(n^3) \). 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 \( \epsilon \) and a point \( Z \in \text{Sym}(n, \mathbb{C}) \), we compute a point \( Z^\epsilon \) close to the original point lying in the \( \epsilon \)-interior of the model. For \( S_n \), starting from \( Z = X + iY \) we orthogonally diagonalize \( Y = K^tDK \), and then modify \( D = \text{diag}(d_t) \) by setting each diagonal entry to \( \max\{d_t, \epsilon\} \). An analogous projection is defined on the bounded domain \( B_n \), see Appendix B.8.

**Algorithm 1 Computing Distances**

1: Given two points \( Z_1, Z_2 \in S_n :=
2: Define \( Z_3 = \sqrt{\det(Z_1)}^{-1} (Z_2 - \Re(Z_1)) \sqrt{\det(Z_1)}^{-1} \in S_n \),
3: Define \( W = (Z_3 - i\text{Id})(Z_3 + i\text{Id})^{-1} \in B_n \),
4: Use the Takagi factorization to write \( W = RK^* \) 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 \( v_1 \geq v_2 \geq \cdots \geq 0 \). The **Vector-valued Distance** is \( \text{vDist}(Z_1, Z_2) = (v_1, v_2, \ldots, v_n) \).
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 \( l^\epsilon \)-metric** is \( d^\epsilon(Z_1, Z_2) := \sum_{i=1}^{n} v_i \).
9: The **Finsler distance inducing the \( l^\infty \)-metric** is \( d^{\infty}(Z_1, Z_2) := \max\{v_i\} = v_1 \).

**Algorithm 2 Computing Riemannian Gradient**

1: Given \( f : S_n \to \mathbb{R} \) and \( Z = X + iY \in 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 \( \text{grad}_R(f) = Y \cdot \text{grad}_E(f) \cdot Y \).
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 id. 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 (E) and hyperbolic (H) spaces (we compare to the Poincaré model (Nickel & Kiela, 2017) since the Bounded Domain model is a generalization of it), Cartesian products thereof (E × H and H × 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 HypSPDₙ 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 Sₙ and Bₙ have n(n + 1)/2 parameters, thus we compare to baselines of the same dimensionality.¹ All implementations are taken from Geoopt (Kochurov et al., 2020).

Metrics: Following previous work (Sala et al., 2018; Gu 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 × Grids of a tree and 2D grids, and Grid × 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 F₁ 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 F₁ 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
Table 3. Results for different datasets in high-dimensional spaces. Best result is **bold**, second best is *underlined*.
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 $v\text{Dist}(Z_i, Z_j) = (v_1, v_2)$ (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 TREE $\odot$ GRIDS the edge coloring distinguishes the hyperbolic parts of the graph (blue edges) and the Euclidean parts (yellow edges). For the GRID $\odot$ 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 $F_\infty$ 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 $T = \{(u, v)\}$, we follow a metric learning approach (Vinh Tran et al., 2020) and learn embeddings by optimizing the following hinge loss function:

$$L = \sum_{(u,v) \in T} \sum_{(u,w) \notin T} [m + d_K(u, v)^2 - d_K(u, w)^2]_+$$  \hspace{1cm} (3)

where $K$ is the target space, $w$ is an item the user has not interacted with, $u, v, w \in K$, $m > 0$ is the hinge margin and $[z]_+ = \max(0, z)$. To generate recommendations, for each user $u$ we rank the items $v_i$ according to their distance to $u$. 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 (Cantarini 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
model marginally outperforms the baselines, in the other two cases the $F_3$ 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 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 ALPHES 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.

---

**Table 4.** Results for recommender system datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ML-1M</th>
<th>ML-100k</th>
<th>LASTFM</th>
<th>Movielens</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR@10</td>
<td>HR@10</td>
<td>HR@10</td>
<td>HR@10</td>
<td></td>
</tr>
<tr>
<td>SGD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E^2$</td>
<td>46.5±0.6</td>
<td>53.4±1.0</td>
<td>28.7</td>
<td>55.2±0.3</td>
</tr>
<tr>
<td>$E^3$</td>
<td>46.0±0.5</td>
<td>54.3±1.0</td>
<td>28.2</td>
<td>54.8±0.5</td>
</tr>
<tr>
<td>$E^4 \times E^2$</td>
<td>52.0±0.7</td>
<td>53.1±1.3</td>
<td>27.9</td>
<td>45.5±0.9</td>
</tr>
<tr>
<td>$SPD_{2n}$</td>
<td>45.8±1.0</td>
<td>53.3±1.4</td>
<td>28.0</td>
<td>55.4±0.2</td>
</tr>
<tr>
<td>$S^n_{10}$</td>
<td>53.8±0.3</td>
<td>27.7</td>
<td>55.7±0.9</td>
<td>28.6</td>
</tr>
<tr>
<td>$S^n_{20}$</td>
<td>53.8±0.3</td>
<td>27.7</td>
<td>55.7±0.9</td>
<td>28.6</td>
</tr>
<tr>
<td>$S^n_{30}$</td>
<td>52.9±0.6</td>
<td>27.2</td>
<td>55.6±1.3</td>
<td>29.4</td>
</tr>
</tbody>
</table>

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


Falkenberg, A. Method to calculate the inverse of a complex matrix using real matrix inversion. 2007.


