A. Symmetric Spaces: a Short Overview

Riemannian symmetric spaces have been extensively studied by mathematicians, and there are many ways to characterize them. They can be described as simply connected Riemannian manifolds, for which the curvature is covariantly constant, or Riemannian manifolds, for which the geodesic reflection in each point defines a global isometry of the space. A key consequence is that symmetric spaces are homogeneous manifolds, which means in particular that the neighbourhood of any point in the space looks the same, and moreover that they can be efficiently described by the theory of semisimple Lie groups.

To be more precise a *symmetric space* is a Riemannian manifold (M, g) such that for any point $p \in M$, the geodesic reflection at p is induced by a global isometry of M. A direct consequence is that the group of isometries $\mathrm{Isom}(M, g)$ acts transitively on M, i.e. given $p, q \in M$ there exists $g \in \mathrm{Isom}(M, g)$ such that $g(p) = q$. Thus symmetric spaces are homogeneous manifolds, which means in particular that the neighbourhood of any point in the space looks the same. This leads to an efficient description by the theory of semisimple Lie groups: $M = \mathsf{G}/\mathsf{K}$ where $\mathsf{G} = \mathrm{Isom}_0(\mathsf{M})$ and K, a compact Lie group, is the stabilizer of a point $p \in M$.

A.1. Classification

Every symmetric space (M, g) can be decomposed into an (almost) product $M = M_1 \times \cdots \times M_k$ of symmetric spaces. A symmetric space is *irreducible*, if it cannot be further decomposed into a Riemannian product $M = M_1 \times M_2$. We restrict our discussion to these fundamental building blocks, the irreducible symmetric spaces.

Irreducible symmetric spaces can be distinguished in two classes, the symmetric spaces of compact type, and the symmetric spaces of non-compact type, with an interesting **duality** between them. Apart from twelve exceptional examples, there are eleven infinite families of pairs of symmetric spaces X of compact and non-compact type, which we summarize in Table 6. We refer the reader to Helgason (1978) for more details and a list of the exceptional examples.

Remark. *Observe that, due to isomorphisms in low dimensions, the first cases of each of the above series is a hyperbolic space (of the suitable dimension). Using this one can construct many natural hyperbolic spaces as totally geodesic submanifolds of the symmetric spaces above. We listed them in Table 7 for the reader's convenience.*

Rank: An important invariant of a symmetric space M is its rank, which is the maximal dimension of an (isometrically embedded) Euclidean submanifold. In a rank r non-compact symmetric space, such submanifolds are isometric to $\mathbb{R}^n$, and called *maximal flats*. In a compact symmetric space, they are compact Euclidean manifolds such as tori.

Some of the rich symmetry of symmetric spaces is visible in the distribution of flats. As homogeneous spaces, each point of a symmetric space M must lie in *some* maximal flat, but in fact for every pair p, q of points in M, one may find some maximal flat containing them. The ability to move any pair of points into a fixed maximal flat by symmetries renders many quantities (such as the metric distances described below) computationally feasible.

A.2. Duality

Compactness provides a useful dichotomy for irreducible symmetric spaces. Symmetric spaces of compact type are compact and of non-negative sectional curvature. The basic example being the sphere S^n. Symmetric spaces of non-compact type are non-compact, in fact they are homeomorphic to $\mathbb{R}^n$ and of non-positive sectional curvature. The basic example being the hyperbolic spaces $\mathbb{H}^n$.

There is a duality between the symmetric spaces of non-compact type and those of compact type, pairing every non-compact symmetric space with its compact 'partner' or dual.

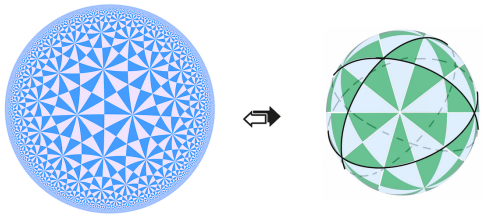

Figure 6. The duality between the hyperbolic plane and sphere is the basic example of the duality between symmetric spaces of compact and noncompact type.

Duality for symmetric spaces generalizes the relationship between spheres and hyperbolic spaces, as well as between classical and hyperbolic trigonometric functions. In the reference Table 6, we provide for each family of symmetric spaces an explicit realization of both the noncompact symmetric space and its compact dual as coset spaces G/K.

A.3. Vector-Valued Distance

The familiar geometric invariant of pairs of points is simply the distance between them. For rank n symmetric spaces, this one dimensional invariant is superseded by an n-dimensional invariant: the *vector valued distance*.

Abstractly, one computes this invariant as follows: for a symmetric space M with $\mathrm{Isom}_0(M) = G$, choose a distinguished basepoint $m \in M$, and let $K < G$ be the subgroup of symmetries fixing m. Additionally choose a distinguished maximal flat $F \subset M$ containing m, and an identification of this flat with $\mathbb{R}^n$. Given any pair of points

Type	Non-compact	Compact	$\mathrm{rk}_{\mathbb{R}}$	dim
AI	$\mathrm{SL}(n,\mathbb{R})/\mathrm{SO}(n,\mathbb{R})$	$\mathrm{SU}(n)/\mathrm{SO}(n)$	$n-1$	$\frac{(n-1)(n+2)}{2}$
A	$\mathrm{SL}(n,\mathbb{C})/\mathrm{SU}(2)$	$(\mathrm{SU}(n)\times\mathrm{SU}(n))/\mathrm{SU}(n)$	$n-1$	$(n+1)(n-1)$
BDI	$\mathrm{SO}(p,q)/\mathrm{SO}(p)\times\mathrm{SO}(q)$	$\mathrm{SO}(p+q)/\mathrm{SO}(p)\times\mathrm{SO}(q)$	$\min\{p,q\}$	pq
AIII	$\mathrm{SU}(p,q)/\mathrm{SU}(p)\times\mathrm{SU}(q)$	$\mathrm{SU}(p+q)/\mathrm{SU}(p)\times\mathrm{SU}(q)$	$\min\{p,q\}$	$2pq$
CI	$\mathrm{Sp}(2n,\mathbb{R})/\mathrm{U}(n)$	$\mathrm{Sp}(2n)/\mathrm{U}(n)$	n	$2n(n+1)$
DIII	$\mathrm{SO}^*(2n)/\mathrm{U}(n)$	$\mathrm{SO}(2n)/\mathrm{U}(n)$	$\lfloor\frac{n}{2}\rfloor$	$n(n-1)$
CII	$\mathrm{Sp}(p,q)/\mathrm{Sp}(p)\times\mathrm{Sp}(q)$	$\mathrm{Sp}(p+q)/\mathrm{Sp}(p)\times\mathrm{Sp}(q)$	$\min\{p,q\}$	$4pq$
AII	$\mathrm{SL}(n,\mathbb{H})/\mathrm{Sp}(n)$	$\mathrm{SU}(2n)/\mathrm{Sp}(n)$	$n-1$	$(n-1)(2n+1)$
D	$\mathrm{SO}(2n,\mathbb{C})/\mathrm{SO}(2n)$	$(\mathrm{SO}(2n)\times\mathrm{SO}(2n))/\mathrm{SO}(2n)$	n	$n(2n-1)$
B	$\mathrm{SO}(2n+1,\mathbb{C})/\mathrm{SO}(2n+1)$	$(\mathrm{SO}(2n+1)\times\mathrm{SO}(2n+1))/\mathrm{SO}(2n+1)$	n	$n(2n+1)$
C	$\mathrm{Sp}(n,\mathbb{C})/\mathrm{Sp}(n)$	$(\mathrm{Sp}(n)\times\mathrm{Sp}(n))/\mathrm{Sp}(n)$	n	

Table 6. The classical symmetric spaces. Row CI represents the Siegel spaces and their compact duals.

Type	Parameters	Symmetric space	
AI	$n=2$	$\mathrm{SL}(2,\mathbb{R})/\mathrm{SO}(2,\mathbb{R})$	$\mathbb{H}^2$
A	$n=2$	$\mathrm{SL}(2,\mathbb{C})/\mathrm{SU}(2)$	$\mathbb{H}^3$
BDI	$p=1$	$\mathrm{SO}(1,q)/\mathrm{SO}(q)$	$\mathbb{H}^q$
AIII	$p=1,q=1$	$\mathrm{SU}(1,1)/\mathrm{SU}(1)\times\mathrm{SU}(1)$	$\mathbb{H}^2$
CI	$n=1$	$\mathrm{Sp}(2,\mathbb{R})/\mathrm{U}(1)$	$\mathbb{H}^2$
DIII	$n=2$	$\mathrm{SO}^*(4)/\mathrm{U}(1)$	$\mathbb{H}^2$
CII	$p=q=1$	$\mathrm{Sp}(1,1)/\mathrm{Sp}(1)\times\mathrm{Sp}(1)$	$\mathbb{H}^4$
AII	$n=1$	$\mathrm{SL}(2,\mathbb{H})/\mathrm{Sp}(1)$	$\mathbb{H}^5$
D	$n=1$	$\mathrm{SO}(2,\mathbb{C})/\mathrm{SO}(2)$	$\mathbb{R}^*$
B	$n=1$	$\mathrm{SO}(3,\mathbb{C})/\mathrm{SO}(3)$	$\mathbb{H}^3$
C	$n=1$	$\mathrm{Sp}(1,\mathbb{C})/\mathrm{Sp}(1)$	$\mathbb{H}^3$

Table 7. Hyperbolic spaces for low parameters

$p,q\in M$, one may find an isometry $g\in G$ moving p to m, and q to some other point $g(q)=v\in F$ in the distinguished flat. Under the identification of F with $\mathbb{R}^n$, the difference vector $v-m$ is a vector-valued invariant of the original two points, and determines the vector valued distance. (In practice we may arrange so that m is identified with $\mathbf{0}$, so this difference is simply v).

In rank 1, the flat F identifies with $\mathbb{R}^1$, and this difference vector $v-m$ with a number. This number encodes all geometric information about the pair (p,q) invariant under the symmetries of M. Indeed, the distance from p to q is simply its absolute value!

In rank n, "taking the absolute value" has an n-dimensional generalization, via a finite a finite group of symmetries of called the Weyl group. This group $W<K$ acts on the flat F, and abstractly, *the vector valued distance* $\mathrm{vDist}(\mathrm{p},\mathrm{q})$ *from* p *to* q *is this difference vector up to the action of the Weyl group*. This vector valued distance $\mathrm{vDist}(\mathrm{p},\mathrm{q})$ is the complete invariant for pairs of points in M - it contains all geometric information about the pair which is invariant under all symmetries. In particular, given the vector valued distance $\mathrm{vDist}(\mathrm{p},\mathrm{q})$, the (Riemannian) distance from p to q is trivial to compute - it is given by the length of $\mathrm{vDist}(\mathrm{p},\mathrm{q})$ in $\mathbb{R}^n$.

The identification of F with $\mathbb{R}^n$ makes this more explicit. Here the Weyl group acts as a group of linear transformations, which divide $\mathbb{R}^n$ into a collection of conical fundamental domains for the action, known as Weyl chambers. Choosing a fixed Weyl chamber C, we may use these symmetries to move our originally found difference vector $v-m$ into C. The vector valued distance is this resulting vector $\mathrm{vDist}(\mathrm{p},\mathrm{q})\in\mathrm{C}$.

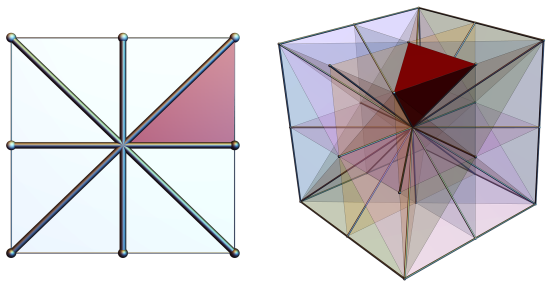

Figure 7. A choice of Weyl chamber the Siegel spaces of rank n is given by $C=\{(v_i)\in\mathbb{R}^n \mid v_1\geq v_2\geq\cdots\geq v_n\geq 0\}$. In rank 1, this is the nonnegative reals. Illustrated here are ranks $n=2,3$.

For example, in rank n Siegel space, the Weyl group acts on $\mathbb{R}^2$ by the reflection symmetries of a cube, and a choice of Weyl chamber amounts to a choice of linear ordering of the vector components with respect to zero. One choice is shown in Figure 7. In rank 2, this chamber is used to display the vector valued distances associated to edges and nodes

of an embedded graph in Figures 13-20. Note that once a Weyl chamber is picked it may be possible to find the vector valued distance corresponding to a vector in $\mathbb{R}^n$ without explicit use of the Weyl group: for the Siegel spaces this is by sorting the vector components in nondecreasing order.

Computing Distance: The process for computing the vector valued distance is summarized below. It is explicitly carried out for the Siegel spaces and their compact duals in Appendix B.

Let M, G, K, F, m be as in the previous section. Choose an identification $\phi\colon F \to \mathbb{R}^n$ which sends the basepoint m to $\mathbf{0}$, and a Weyl chamber $C \subset \mathbb{R}^n$ for the Weyl group W. For any pair of points $p, q \in M$;

1. **Move p to the basepoint:**
   Compute $g \in G$ such that $g(p) = m$.
2. **Move q into the flat:**
   Compute $k \in K$ such that $k(g(q)) \in F$. Now both $g(p) = m$ and $k(g(q))$ lie in the distinguished flat F.
3. **Identify the flat with $\mathbb{R}^n$:**
   Compute $u = \phi(k(g(q))) \in \mathbb{R}^n$. The points $\mathbf{0}$ and u represent p, q after being moved into the flat, respectively.
4. **Return the Vector Valued Distance:**
   Compute $v \in C$ such that $v = Au$ for some element $A \in W$. This is the vector valued distance $\mathrm{vDist}(\mathrm{p}, \mathrm{q})$

The **Riemannian distance** is computed directly from the vector valued distance as its Euclidean norm, $\mathrm{dist}(p, q) = \|\mathrm{vDist}(p, q)\|$.

A.4. Finsler Metrics

A Riemannian metric on a manifold M is defined by a smooth choice of inner product on the tangent bundle. Finsler metrics generalize this by requiring only a smoothly varying choice of norm $\|\cdot\|_F$. The length of a curve γ is defined via integration of this norm along the path

$$\mathrm{Length}_F(\gamma) = \int_I \|\gamma'\|_F dt,$$

and the distance between points by the infimum of this over all rectifiable curves joining them

$$d_F(p, q) = \inf\{\mathrm{Length}_F(\gamma) \mid \gamma(0) = p,\ \gamma(1) = q\}$$

The geometry of symmetric spaces allows the computation of Finsler distances, like much else, to take place in a chosen maximal flat. On such flat spaces, the ability to identify all tangent spaces allow particularly simple Finsler metrics to be defined by choosing a single norm on $\mathbb{R}^n$. We quickly review this theory below.

Finsler Metrics on $\mathbb{R}^n$: Any norm on $\mathbb{R}^n$ defines a Finsler metric. As norms on a vector space are uniquely determined by their unit spheres, the data of a Finsler metric is given by a convex polytope S containing $\mathbf{0}$. An important example in this work is the ℓ^1 Finsler metric on $\mathbb{R}^n$, given by the norm $\|(x_i)\|_{\ell^1} = \sum_i |x_i|$. Its unit sphere is the boundary of the dual to the n-dimensional cube (in $\mathbb{R}^2$, this is again a square, but oriented at 45° with respect to the coordinate axes).

Given such an P, the Finsler norm $\|v\|_F$ of a vector $v \in \mathbb{R}^n$ is the unique positive ℓ such that $\frac{1}{\ell} v \in \partial P$. Figure 8 below shows the spheres of radius 1 and 2 with respect to the ℓ^1 metric on the plane.

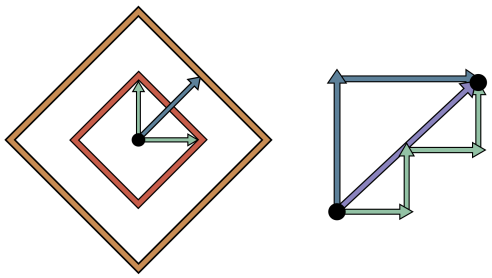

Figure 8. Left: Vectors of length 1 and 2 with respect to the ℓ^1 norm on $\mathbb{R}^2$. Right: three geodesics of length 4 in the ℓ^1 metric (to same scale as left image).

While affine lines are geodesics in Finsler geometry, they need not be the unique geodesics between a pair of points. Consider again Figure 8: the vector sum of the two unit vectors in is exactly the diagonal vector, which lies on the ℓ^1 sphere of radius 2. That is, in ℓ^1 geometry traveling along the diagonal, or along the union of a vertical and horizontal side of a square both are distance minimizing paths of length 2. The ℓ^1 metric is often called the 'taxicab' metric for this reason: much as in a city with a grid layout of streets, there are many shortest paths between a generic pair of points, as you may break your path into different choices of horizontal and vertical segments without changing its length. See Figure 2 in the main text for another example of this.

Finsler Metrics on Symmetric Spaces: To define a Finsler metric on a symmetric space M, it suffices to define it on a chosen maximal flat, and evaluate on arbitrary pairs of points with the help of the vector valued distance. To induce a well defined Finsler metric M, a norm on this designated flat need only be invariant under the Weyl group W. Said geometrically, the unit sphere of the norm $\|\cdot\|_F$ needs to contain it as a subgroup of its symmetries. Given such a norm, the Finsler distance between two points is simply the

Finsler norm of their vector valued distance

$$d_F(p,q) = \|\mathrm{vDist}(p,q)\|_F.$$

Consequentially once the vector valued distance is known, any selection of Riemannian or Finsler distances may be computed at marginal additional cost.

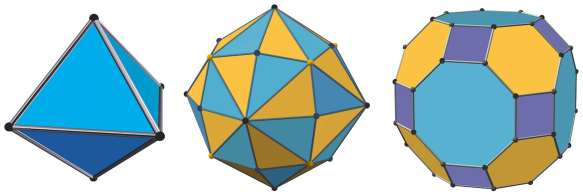

Figure 9. The unit spheres of several Finsler metrics on $\mathbb{R}^3$ invariant under the Weyl group of the rank 3 Siegel space. The octahedron induces the ℓ^1 metric.

A.5. Local Geometry for Riemannian Optimization

Different Riemannian optimization methods require various input from the local geometry - here we describe a computation of the Riemannian gradient, parallel transport and the exponential map for general irreducible symmetric spaces.

Riemannian Gradient Given a function $f\colon M \to \mathbb{R}$, the *differential* of f is a 1-form which measures how infinitesimal changes in the input affects (infinitesimally) the output. More precisely at each point $p \in M$, df is a linear functional on T_pM sending a vector v to the directional derivative $df_p(v)$ of f in direction v.

In Euclidean space, this data is conveniently expressed as a vector: *the gradient* ∇f defined such that $(\nabla f(p)) \cdot v = df_p(v)$. This extends directly to any Riemannian manifold, where the dot product is replaced with the Riemannian metric. That is, the *Riemannian gradient* of a function $f\colon M \to \mathbb{R}$ is the vector field $grad_R(f)$ on M such that

$$g_p(\mathrm{grad}_R(f), v) = df_p(v)$$

for every $p \in M$, $v \in T_pM$. Given a particular model (and thus, a particular coordinate system and metric tensor) one may use this implicit definition to give a formula for grad_R. See Appendix B.6 for an explicit example, deriving the Riemannian gradient for Siegel space from its metric tensor.

Parallel Transport

While the lack of curvature in Euclidean space allows all tangent spaces to be identified, in general symmetric spaces the result of transporting a vector from one tangent space to another is a nontrivial, path dependent operation. This *parallel transport* assigns to a path γ in M from p to q an isomorphism $P_\gamma\colon T_pM \to T_qM$ interpreted as taking a vector $v \in T_pM$ at p to $P_\gamma(v) \in T_qM$ by "moving without turning" along γ.

The computation of parallel transport along geodesics in a symmetric space is possible directly from the isometry group. To fix notation, for each $m \in M$ let $\sigma_m \in G$ be the geodesic reflection fixing m. Let γ be a geodesic in M through p at $t = 0$. As t varies, the isometries $\tau_t = \sigma_{\gamma(t/2)} \circ \sigma_p$, called *transvections*, form the 1-parameter subgroup of translations along γ. If $p, q \in M$ are two points at distance L apart along the the geodesic γ, the transvection τ_L takes p to q, and its derivative $(d\tau_L)_p = P_\gamma\colon T_pM \to T_qM$ performs the parallel transport for γ.

The Exponential Map & Lie Algebra The exponential map for a Riemannian manifold M is the map $\exp\colon TM \to M$ such that if $v \in T_p(X)$, $\exp(v)$ is the point in M reached by traveling distance $\|v\|$ along the geodesic on M through p with initial direction parallel to v.

When M is a symmetric space with symmetry group G, this may be computed using the Lie group exponential $\exp\colon \mathfrak{g} \to G$ (the matrix exponential, when G is a matrix Lie group). Choose a point $p \in M$ and let σ_p be the geodesic reflection in p. Then σ_p defines an involution $G \to G$ by $g \mapsto \sigma_p \circ g \circ \sigma_p$ (where composition is as isometries of M), and the eigenspaces of the differential of this involtuion give a decomposition $\mathfrak{g} = \mathfrak{k} \oplus \mathfrak{p}$ into the $+1$ eigenspace $\mathfrak{k}$ and the -1 eigenspace $\mathfrak{p}$. Here $\mathfrak{k}$ is the Lie algebra of the stabilizer $K = \mathfrak{stab}(p) < G$, and so $\mathfrak{p}$ identifies with T_pM under the differential of the quotient $G \to G/K \cong M$.

Let $\phi\colon T_pM \to \mathfrak{p}$ be the inverse of this identification. Then for a vector $v \in T_pM$, we may find the point $q = \exp_p(v) \in M$ as follows:

1. Compute $\phi(v) = A \in \mathfrak{p}$. This is the tangent vector v, viewed as a matrix in the Lie algebra to G.
2. Compute $g = \exp(A)$, where $\exp$ is the matrix exponential.
3. Use the action of G on M by isometries to compute $q = g(p)$.

B. Explicit Formulas for Siegel Spaces

This section gives the calculations mathematics required to implement two models of Siegel space (the bounded domain model and upper half space) as well as a model of its compact dual.

B.1. Linear Algebra Conventions

A few clarifications from linear algebra can be useful:

1. The inverse of a matrix X^{-1}, the product of two matrices XY, the square X^2 of a square matrix are understood with respect to the matrix operations. Unless all matrices are diagonal these are different than doing the same operation to each entry of the matrix.

2. If $Z = X + iY$ is a complex matrix,
   - Z^t denotes the transpose matrix, i.e. $(Z^t)_{ij} = Z_{ji}$,
   - $\overline{Z} = X - iY$ denotes the complex conjugate
   - X^* denotes its transpose conjugate, i.e. $X^* = \overline{X^t}$.

3. A complex square matrix Z is *Hermitian* if $Z^* = Z$. In this case its eigenvalues are real and positive. It is *unitary* if $Z^* = Z^{-1}$. In this case its eigenvalues are complex numbers of absolute value 1 (i.e. points in the unit circle).

4. If X is a real symmetric, or complex Hermitian matrix, $X \gg 0$ means that X is positive definite, equivalently all its eigenvalues are bigger than zero.

B.2. Takagi Factorization

Given a complex symmetric matrix A, the Takagi factorization is an algorithm that computes a real diagonal matrix D and a complex unitary matrix K such that

$$A = \overline{K} D K^*.$$

This will be useful to work with the bounded domain model. It is done in a few steps

1. Find Z_1 unitary, D real diagonal such that

$$A^* A = Z_1^* D^2 Z_1$$

2. Find Z_2 orthogonal, B complex diagonal such that

$$\overline{Z}_1 A Z_1^* = Z_2 B Z_2^t$$

   This is possible since the real and imaginary parts of $\overline{Z}_1 A Z_1^*$ are symmetric and commute, and are therefore diagonalizable in the same orthogonal basis.

3. Set Z_3 be the diagonal matrix with entries

$$(Z_3)_{ii} = \left(\sqrt{\frac{b_i}{|b_i|}} \right)^{-1}$$

   where $b_i = (B)_{ii}$

4. Set $K = Z_1^* Z_2 Z_3$, D as in Step 1. It then holds

$$A = \overline{K} D K^*.$$

B.3. Siegel Space and its Models

We consider two models for the symmetric space, the bounded domain

$$\mathcal{B}_n := \{Z \in \mathrm{Sym}(n, \mathbb{C}) | \ \mathrm{Id} - Z^* Z \gg 0\}$$

and the upper half space

$$\mathcal{S}_n := \{X + iY \in \mathrm{Sym}(n, \mathbb{C}) | \ Y \gg 0\}.$$

An explicit isomorphism between the two domains is given by the Cayley transform

$$\begin{array}{rccc} c: & \mathcal{B}_n & \to & \mathcal{S}_n \\ & Z & \mapsto & i(Z + \mathrm{Id})(Z - \mathrm{Id})^{-1} \end{array}$$

whose inverse $c^{-1} = t$ is given by

$$\begin{array}{rccc} t: & \mathcal{S}_n & \to & \mathcal{B}_n \\ & X & \mapsto & (X - i\mathrm{Id})(X + i\mathrm{Id})^{-1} \end{array}$$

When needed, a choice of **basepoint** for these models is $i\mathrm{Id} \in \mathcal{S}_n$ for upper half space and the zero matrix $\mathbf{0} \in \mathcal{B}_n$ for the bounded domain. A convenient choice of **maximal flats** containing these basepoints are the subspaces $\{iD \mid D = \mathrm{diag}(d_i),\ d_i > 0\} \subset \mathcal{S}_n$ and $\{D = \mathrm{diag}(d_i) \mid d_i \in (-1, 1)\} \subset \mathcal{B}_n$.

The group of symmetries of the Siegel space $\mathcal{S}_n$ is $\mathrm{Sp}(2n, \mathbb{R})$, the subgroup of $\mathrm{SL}(2n, \mathbb{R})$ preserving a symplectic form: a non-degenerate antisymmetric bilinear form on $\mathbb{R}^{2n}$. In this text we will choose the symplectic form represented, with respect to the standard basis, by the matrix $\left(\begin{smallmatrix} 0 & \mathrm{Id}_n \\ -\mathrm{Id}_n & 0 \end{smallmatrix}\right)$ so that the symplectic group is given by the matrices that have the block expression

$$\left\{ \begin{pmatrix} A & B \\ C & D \end{pmatrix} \middle| \begin{array}{l} A^t D - C^t B = \mathrm{Id} \\ A^t C = C^t A \\ B^t D = D^t B \end{array} \right\}$$

where A, B, C, D are real $n \times n$ matrices.

The symplectic group $\mathrm{Sp}(2n, \mathbb{R})$ acts on $\mathcal{S}_n$ by non-commutative fractional linear transformations

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot Z = (AZ + B)(CZ + D)^{-1}.$$

The action of $\mathrm{Sp}(2n, \mathbb{R})$ on $\mathcal{B}_n$ can be obtained through the Cayley transform.

B.4. Computing the Vector-Valued Distance

The Riemannian metric, as well as any desired Finsler distance, are computable directly from the vector-valued distance as explained in Appendix A.3. Following those steps, we give an explicit implementation for the upper half space

model below, and subsequently use the Cayley transform to extend this to the bounded domain model.

Given as input two points $Z_1, Z_2 \in \mathcal{S}_n$ we perform the following computations:

1) Move Z_1 to the basepoint: Compute the image of Z_2 under the transformation taking Z_1 to iI, defining

$$Z_3 := \sqrt{\Im Z_1}^{-1}(Z_2 - \Re Z_1)\sqrt{\Im Z_1}^{-1} \in \mathcal{S}_n$$

2) Move Z_2 into the chosen flat: Define

$$W = t(Z_3) \in \mathcal{B},$$

and use the Takagi factorization to write

$$W = \overline{K}DK^*$$

for some real diagonal matrix D with eigenvalues between 0 and 1, and some unitary matrix K. *Note: to make computations easier, we are leveraging the geometry of both models here, so in fact $i(I+D)(I-D)^{-1}$ is the matrix lying in the standard flat containing iI.*

3) Identify the flat with $\mathbb{R}^n$: Define the vector $v = (v_i) \in \mathbb{R}^n$ with

$$v_i = \log \frac{1+d_i}{1-d_i},$$

for d_i the i^{th} diagonal entry of the matrix D from the last step.

4) Return the Vector Valued Distance: Sort the absolute values of the entries of v to be in nonincreasing order, and set $\mathrm{vDist}(Z_1, Z_2)$ equal to the resulting list.

$$\mathrm{vDist} = (|v_{i_1}|, |v_{i_2}|, \ldots, |v_{i_n}|)$$

$$|v_{i_1}| \geq |v_{i_2}| \geq \cdots \geq |v_{i_n}|$$

Bounded domain: In this case, given $W_1, W_2 \in \mathcal{B}$ we consider the pair $Z_1, Z_2 \in \mathcal{S}_n$ obtained applying the Cayley transform $Z_i = t(W_i)$. Then we can apply the previous algorithm, indeed

$$\mathrm{vDist}(W_1, W_2) = \mathrm{vDist}(Z_1, Z_2).$$

B.5. Riemannian & Finsler Distances

The Riemannian distance between two points X, Y in the Siegel space (either the upper half space or bounded domain model) is induced by the Euclidean metric on its maximal flats. This is calculable directly from the vector valued distance $\mathrm{vDist}(X, Y) = (v_1, v_2, \ldots, v_n)$ as

$$d^R(X, Y) = \sqrt{\sum_{i=1}^{n} v_i^2}.$$

The Weyl group for the rank n Siegel space is the symmetry group of the n cube. Thus, any Finsler metric on $\mathbb{R}^n$ whose unit sphere has these symmetries has these symmetries induces a Finsler metric on Siegel space. The class of such finsler metrics includes many well-known examples such as the ℓ^p metrics

$$\|(v_1, \ldots, v_n)\|_{\ell^p} = \left(\sum_i |v_i|^p\right)^{\frac{1}{p}},$$

which is one of the reasons the Siegel space is an attractive avenue for experimentation.

Of particular interest are the ℓ^1 and ℓ^∞ Finsler metrics. The distance functions induced on the Siegel space by them are given below

$$d^{F_1}(X, Y) = \sum_{i=1}^{n} v_i \qquad d^{F_\infty}(X, Y) = v_1.$$

Where X, Y are points in Siegel space (again, either in the upper half space or bounded domain models), and the v_i are the component of the vector valued distance $\mathrm{vDist}(X, Y) = (v_1, v_2, \ldots, v_n)$.

There are explicit bounds between the distances, for example

$$\frac{1}{\sqrt{n}} d^{F_1}(X, Y) \leq d^R(X, Y) \leq d^{F_1}(X, Y) \tag{4}$$

Furthermore, we have

$$\begin{aligned} d^{F_1}(X, Y) = \log\det(\sqrt{R(X, Y)} + \mathrm{Id}) - \\ \log\det(\mathrm{Id} - \sqrt{R(X, Y)}) \end{aligned} \tag{5}$$

which, in turn, allows to estimate the Riemannian distance using (4).

B.6. Riemannian Gradient

We consider on $\mathrm{Sym}(n, \mathbb{C})$ the Euclidean metric given by

$$\|V\|_E^2 = \mathrm{tr}(V\overline{V}),$$

here tr denotes the trace, and, as above, $V\overline{V}$ denotes the matrix product of the matrix V and its conjugate.

Siegel upperhalf space: The Riemannian metric at a point $Z \in \mathcal{S}_n$, where $Z = X + iY$ is given by (Siegel, 1943)

$$\|V\|_R^2 = \mathrm{tr}(Y^{-1}VY^{-1}\overline{V}).$$

As a result we deduce that

$$\mathrm{grad}(f(Z)) = Y \cdot \mathrm{grad}_{\mathrm{E}}(f(Z)) \cdot Y$$

Bounded domain: In this case we have

$$\mathrm{grad}(f(Z)) = A \cdot \mathrm{grad}_{\mathrm{E}}(f(Z)) \cdot A$$

where $A = \mathrm{Id} - \overline{Z}Z$

B.7. Embedding Initialization

Different embeddings methods initialize the points close to a fixed basepoint. In this manner, no a priori bias is introduced in the model, since all the embeddings start with similar values.

We use the **basepoints** specified previously: $i\mathrm{Id}$ for Siegel upper half space and $\mathbf{0}$ for the bounded domain model.

In order to produce a random point we generate a random symmetric matrix with small entries and add it to our basepoint. As soon as all entries of the perturbation are smaller than $1/n$ the resulting matrix necessarily belongs to the model. In our experiments, we generate random symmetric matrices with entries taken from a uniform distribution $\mathcal{U}(-0.001, 0.001)$.

B.8. Projecting Back to the Models

The goal of this section is to explain two algorithms that, given ϵ and a point $Z \in \mathrm{Sym}(n, \mathbb{C})$, return a point $Z_\epsilon^{\mathcal{S}}$ (resp. $Z_\epsilon^{\mathcal{B}}$), a point close to the original point lying in the ϵ-interior of the model. This is the equivalent of the projection applied in Nickel & Kiela (2017) to constrain the embeddings to remain within the Poincaré ball, but adapted to the structure of the model. Observe that the projections are not conjugated through the Cayley transform.

Siegel upperhalf space: In the case of the Siegel upperhalf space $\mathcal{S}_n$ given a point $Z = X + iY \in \mathrm{Sym}(n, \mathbb{C})$

1. Find a real n-dimensional diagonal matrix D and an orthogonal matrix K such that
$$Y = K^t D K$$
2. Compute the diagonal matrix D_ϵ with the property that
$$(D_\epsilon)_{ii} = \begin{cases} D_{ii} & \text{if } D_{ii} > \epsilon \\ \epsilon & \text{otherwise} \end{cases}$$
3. The projection is given by
$$Z_\epsilon^{\mathcal{S}} := X + iK^t D_\epsilon K$$

Bounded Domain: In the case of the bounded domain $\mathcal{B}$ given a point $Z = X + iY \in \mathrm{Sym}(n, \mathbb{C})$

1. Use the Takagi factorization to find a real n-dimensional diagonal matrix D and an unitary matrix K such that
$$Y = \overline{K} D K^*$$
2. Compute the diagonal matrix $D_\epsilon^{\mathcal{B}}$ with the property that
$$(D_\epsilon^{\mathcal{B}})_{ii} = \begin{cases} D_{ii} & \text{if } D_{ii} < 1 - \epsilon \\ 1 - \epsilon & \text{otherwise} \end{cases}$$
3. The projection is given by
$$Z_\epsilon^{\mathcal{B}} := \overline{K} D_\epsilon^{\mathcal{B}} K^*$$

B.9. Crossratio and Distance

Given two points X, Y in Siegel space, there is an alternative means of calculating the vector valued distance (and thus any Riemannian or Finsler distance one wishes) via an invariant known as the cross ratio.

Siegel upperhalf space: Given two points $X, Y \in \mathcal{S}_n$ their crossratio is given by the complex $n \times n$-matrix

$$R_{\mathcal{S}}(X, Y) = (X - Y)(X - \overline{Y})^{-1}(\overline{X} - \overline{Y})(\overline{X} - Y)^{-1}.$$

It was established by Siegel (Siegel, 1943) that if $r_1, \ldots, r_n$ denote the eigenvalues of R (which are necessarily real greater than or equal to 1) and we denote by v_i the numbers

$$v_i = \log \frac{1 + \sqrt{r_i}}{1 - \sqrt{r_i}}$$

then the v_i are the components of the vector-valued distandce $\mathrm{vDist}(X, Y)$. Thus, the Riemannian distance is

$$d^R(X, Y) = \sqrt{\sum_{i=1}^{n} v_i^2}.$$

The Finsler distances d^{F_1} and d^{F_∞} are likewise given by the same formulas as previously.

In general it is computationally difficult to compute the eigenvalues, or the squareroot, of a general complex matrix. However, we can use the determinant $\det_R$ of the matrix $R(X, Y)$ to give a lower bound on the distance:

$$\log \frac{1 + \sqrt{\det_R}}{1 - \sqrt{\det_R}} \leq d^R(X, Y).$$

Bounded domain: The same study applies to pairs of points $X, Y \in \mathcal{B}$, but their crossratio should be replaced by the expression

$$\begin{aligned} R_{\mathcal{B}}(X, Y) = (X - Y)(X - \overline{Y^{-1}})^{-1} \\ (\overline{X^{-1}} - \overline{Y^{-1}})(\overline{X^{-1}} - Y)^{-1} \end{aligned} \tag{6}$$

B.10. The Compact Dual of the Siegel Space

The compact dual to the (non-positively) curved Siegel space is a compact non-negatively curved symmetric space; in rank 1 this is just the 2-sphere. Many computations in the compact dual are analogous to those for the Siegel spaces, and are presented below.

MODEL

Abstractly, the compact dual is the space of complex structures on quaternionic n-dimensional space compatible with a fixed inner product. It is convenient to work with a coordinate chart, or *affine path* covering all but a measure zero subset of this space. We denote this patch by $\mathcal{D}_n$, which consists of all $n \times n$ complex symmetric matrices:

$$\mathcal{D}_n = \mathrm{Sym}(n; \mathbb{C})$$

With this choice of model, tangent vectors to $\mathcal{D}_n$ are also represented by complex symmetric matrices. More precisely, for each $W \in \mathcal{D}_n$ we may identify the tangent space $T_W\mathcal{D}_n$ with $\mathrm{Sym}(n, \mathbb{C})$.

Basepoint: The basepoint of $\mathcal{D}_n$ is the zero matrix $\mathbf{0}$.

Maximal Flat: A useful choice of maximal flat is the subspace of real diagonal matrices.

Projection: The model $\mathcal{D}_n$ is a linear subspace of the space of $n \times n$ complex matrices. Orthogonal projection onto this subspace is given by symmetrization,

$$W \mapsto \frac{1}{2}(W + W^t).$$

Isometries: The symmetries of the compact dual are given by the compact symplectic group $\mathrm{Sp}(n)$. With respect to the model $\mathcal{D}_n$, we may realize this as the intersection of the complex symplectic group $\mathrm{Sp}(2n, \mathbb{C})$ and the unitary group $U(2n, \mathbb{C})$

$$\left\{ \begin{pmatrix} A & B \\ C & D \end{pmatrix} \;\middle|\; \begin{array}{l} A^tD - C^tB = \mathrm{Id} \\ A^tC = C^tA \\ B^tD = D^tB \\ A^*A + C^*C = \mathrm{Id} \\ B^*B + D^*D = \mathrm{Id} \\ A^*B + C^*D = 0 \end{array} \right\}$$

where A, B, C, D are *complex* $n \times n$ matrices. The first four conditions are analogs of those defining $\mathrm{Sp}(2n; \mathbb{R})$, and the final three come from the defining property that a unitary matrix M satisfies $M^*M = \mathrm{Id}$.

This group acts on $\mathcal{D}_n$ by non-commutative fractional linear transformations

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot W = (AW + B)(CW + D)^{-1}.$$

Riemannian Metric & Gradient: The Riemannian metric at a point $W \in \mathcal{D}_n$ is given by

$$\langle U, V \rangle_W = (\mathrm{Id} + \overline{W}W)^{-1}U(\mathrm{Id} + \overline{W}W)^{-1}\overline{V},$$

where U, V are tangent vectors at W.

The gradient of a function on the compact dual can be written in terms of its Euclidean gradient, via a formula very similar to that for the Bounded Domain model of the Siegel space. In this case we have

$$\mathrm{grad}(f(W)) = A \cdot \mathrm{grad}_{\mathrm{E}}(f(W)) \cdot A$$

where (the only difference from the bounded domain version being that the $-$ sign in the definition of A has been replaced with a $+$).

VECTOR VALUED DISTANCE

We again give an explicit implementation of the abstract procedure described in Appendix A.3, to calculate the vector valued distance associated to an arbitrary pair $W_1, W_2 \in \mathcal{D}_n$ as follows:

Move $W1$ to the basepoint:

1. Use the Takagi factorization to write
$$W_1 = \overline{U}PU^*$$
for a unitary matrix U and real diagonal matrix P.
2. From P, we build the diagonal matrix $A = (\mathrm{Id} + P^2)^{-1/2}$. That is, the diagonal entries of A are $a_i = \frac{1}{\sqrt{1+p_i^2}}$ for p_i the diagonal entries of P.
3. From A, U we build the following elements $M, R \in \mathrm{Sp}(n)$ of the compact symplectic group:
$$M = \begin{pmatrix} A & -AP \\ AP & A \end{pmatrix} \qquad R = \begin{pmatrix} U^t & 0 \\ 0 & U^* \end{pmatrix}$$

We now use the transformation $M \cdot R$ to move the pair (W_1, W_2) to a pair $(\mathbf{0}, Z)$. Because W_1 ends at the basepoint by construction, we focus on W_2.

4. Compute $X = R.W_2$, that is $X = U^tW_2U$.
5. Compute $Z = M.X$, that is $Z = (AX - AP)(APX - A)^{-1}$. Alternatively, this simplifies to the conjugation $Z = AYA^{-1}$ by A of the matrix $Y = (X - P)(PX - \mathrm{Id})^{-1}$

Move Z into the chosen flat: Use the Takagi factorization to write

$$Z = \overline{K}DK^*$$

for a unitary matrix K and real diagonal matrix D.

Identify the Flat with $\mathbb{R}^n$: Produce from D the n-vector

$$v = (\arctan(d_1), \ldots \arctan(d_n))$$

Where $(d_1, \ldots d_n)$ are the diagonal entries of D.

Return the Vector Valued Distance: Order the the entries of v in nondecreasing order. This is the vector valued distance.

$$\text{vDist} = (v_{i_1}, v_{i_2}, \ldots, v_{i_n})$$
$$v_{i_1} \geq v_{i_2} \geq \cdots \geq v_{i_n} \geq 0$$

RIEMANNIAN AND FINSLER DISTANCES:

The Riemannian distance between two points X, Y in the compact dual is calculable directly from the vector valued distance $\text{vDist}(X, Y) = (v_1, v_2, \ldots, v_n)$ as

$$d^R(X, Y) = \sqrt{\sum_{i=1}^{n} v_i^2}.$$

The Weyl group for the compact dual is the same as for Seigel space, the symmetries of the n-cube. Thus the same collection of Finsler metrics induce distance functions on the compact dual, and their formulas in terms of the vector valued distance are unchanged.

$$d^{F_1}(X, Y) = \sum_{i=1}^{n} v_i \qquad d^{F_\infty}(X, Y) = v_1.$$

B.11. Interpolation between Siegel Space and its Compact Dual

The Siegel space and its compact dual are part of a 1-parameter family of spaces indexed by a real parameter $k \in \mathbb{R}$. When $n = 1$ the symmetric spaces are two-dimensional, and this k is interpreted as their (constant) curvature. That is, this family represents an interpolation between the hyperbolic plane ($k = -1$) and the sphere ($k = 1$) through Euclidean space ($k = 0$) as schematically represented in Figure 10. Below we describe the generalization of this to all n, by giving the model, symmetries, and distance functions in terms of the parameter $k \in \mathbb{R}$.

Model: Our models are most similar to the Bounded domain model of Siegel space, and so we use notation to match. For each $k \in \mathbb{R}$ we define the subset $\mathcal{B}_n^k$ of $\text{Sym}(n, \mathbb{C})$ as follows:

$$\mathcal{B}_n^k = \begin{cases} \{W \mid \text{Id} + kW^*W >> 0\} & k < 0 \\ \text{Sym}(n, \mathbb{C}) & k \geq 0 \end{cases}$$

The **basepoint** for $\mathcal{B}_n^k$ is the zero matrix $\mathbf{0}$ for all k. **Projection back to the model** is analogous to what is done for the bounded domain when $k < 0$, and is just symmetrization for $k \geq 0$.

Isometries: Denote by G^k the isometry group of $\mathcal{B}_n^k$. A uniform description of G^k can be given in close analogy to the description of the symmetries of the compact dual.

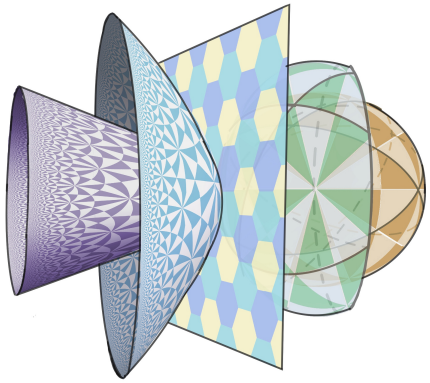

Figure 10. A 1-parameter family of spaces interpolating between the Siegel space and its compact dual, here illustrated in rank 1 ($\mathbb{H}^2$ transitioning S^2 through the Euclidean plane with $k = 0$).

For each $k \in \mathbb{R}$, $G^k = \text{Sp}(2n, \mathbb{C}) \cap \mathcal{U}^k$ where $\mathcal{U}^k$ is a generalization of the usual unitary group

$$\mathcal{U}^k = \left\{ M \mid M^* \left(\begin{smallmatrix} k\text{Id} & 0 \\ 0 & \text{Id} \end{smallmatrix}\right) M = \left(\begin{smallmatrix} k\text{Id} & 0 \\ 0 & \text{Id} \end{smallmatrix}\right) \right\}$$

Riemannian Geometry: The Riemannian metric at a point $W \in \mathcal{B}_n^k$ is given by the formula

$$\langle U, V \rangle_W^k = \text{tr}\left(A^{-1} U A^{-1} \overline{V}\right)$$

Where $A = \text{Id} + k\overline{W}W$. As before, this allows us to compute the **Riemannian Gradient** in terms of the Euclidean gradient on $\mathcal{B}_n^k$:

$$\text{grad}(f(W)) = A \cdot \text{grad}_{\text{E}}(f(W)) \cdot A$$

From the Riemannian metric we may explicitly compute the distance function from the basepoint $\mathbf{0}$ to a real diagonal matrix $D \in \mathcal{B}_n^k$:

$$\text{dist}^k(\mathbf{0}, D) = \begin{cases} \sqrt{\sum_i \frac{\text{arctanh}\left(d_i \sqrt{|k|}\right)^2}{\sqrt{|k|}}} & k < 0 \\ \sqrt{\sum_i d_i^2} & k = 0 \\ \sqrt{\sum_i \frac{\arctan\left(d_i \sqrt{k}\right)^2}{\sqrt{k}}} & k > 0 \end{cases}$$

Distance: The seven step procedure for calculating distance in the compact dual can be modified to give a procedure for the distance in $\mathcal{B}_n^k$. To calculate the Riemannian distance, Step 7 must be replaced with the distance formula above. The only other changes involve the construction of the matrix called M

- In step 2, the computation of P is unchanged but A is replaced with $A = (\text{Id} + kP^2)^{-1/2}$.
- In step 3, the matrix M is replaced with

$$M = \begin{pmatrix} A & -AP \\ \text{sgn}(k)AP & A \end{pmatrix}$$

$(\lvert V\rvert, \lvert E\rvert)$	2D GRID $(36, 60)$		TREE $(40, 39)$	
	D_{avg}	mAP	D_{avg}	mAP
$\mathcal{S}_3^R$	12.29	100.00	4.27	95.00
$\mathcal{S}_3^{F_\infty}$	0.21	100.00	**2.01**	100.00
$\mathcal{S}_3^{F_1}$	0.02	100.00	2.10	100.00
$\mathcal{B}_3^R$	12.26	100.00	4.14	94.17
$\mathcal{B}_3^{F_\infty}$	0.29	100.00	2.04	100.00
$\mathcal{B}_3^{F_1}$	**0.01**	100.00	2.06	99.17
$\mathcal{D}_3^R$	47.59	54.35	69.65	29.05
$\mathcal{D}_3^{F_\infty}$	63.85	18.94	75.33	15.18
$\mathcal{D}_3^{F_1}$	28.68	82.96	38.84	55.28
$\mathcal{S}_4^R$	12.27	100.00	4.20	98.33
$\mathcal{S}_4^{F_\infty}$	0.49	100.00	1.72	100.00
$\mathcal{S}_4^{F_1}$	**0.01**	100.00	1.58	100.00
$\mathcal{B}_4^R$	12.24	100.00	4.10	100.00
$\mathcal{B}_4^{F_\infty}$	0.17	100.00	**1.18**	100.00
$\mathcal{B}_4^{F_1}$	**0.01**	100.00	1.48	100.00
$\mathcal{D}_4^R$	41.82	78.20	65.95	31.76
$\mathcal{D}_4^{F_\infty}$	53.31	79.34	74.19	19.16
$\mathcal{D}_4^{F_1}$	13.38	100.00	23.64	71.94

Table 8. Comparison of the compact dual model to the upper half space and bounded domain model on two synthetic datasets.

Where $\text{sgn}(k) = 0$ if $k = 0$. Note the computation of $M.X$ in Step 5 also changes, as M has changed. Now $M.X = (AX - AP)(\text{sgn}(k)APX - A)^{-1}$.

B.12. Experiments on the Compact Dual

We perform experiments on small synthetic datasets to compare the performance of the dual model to the upper half Siegel space and the Bounded domain model. Results are reported in Table 8. We can observe the reduced representation capabilities of the compact dual model, even on small datasets.

C. Experimental Setup

C.1. Implementation of Complex Operations

All models and experiments were implemented in PyTorch (Paszke et al., 2019) with distributed data parallelism, for high performance on clusters of CPUs/GPUs.

Given a complex matrix $Z \in \mathbb{C}^{n\times n}$, we model real and imaginary components $Z = X + iY$ with $X, Y \in \mathbb{R}^{n\times n}$ separate matrices with real entries. We followed standard complex math to implement basic arithmetic matrix operations. For complex matrix inversion we implemented the procedure detailed in Falkenberg (2007).

Hardware: All experiments were run on Intel Cascade Lake CPUs, with microprocessors Intel Xeon Gold 6230 (20 Cores, 40 Threads, 2.1 GHz, 28MB Cache, 125W TDP). Although the code supports GPUs, we did not utilize them due to higher availability of CPU's.

C.2. Optimization

As stated before, the models under consideration are Riemannian manifolds, therefore they can be optimized via stochastic Riemannian optimization methods such as RSGD (Bonnabel, 2011) (we adapt the Geoopt implementation (Kochurov et al., 2020)). Given a function $f(\theta)$ defined over the set of embeddings (parameters) θ and let ∇_R denote the Riemannian gradient of $f(\theta)$, the parameter update according to RSGD is of the form:

$$\theta_{t+1} = \mathcal{R}_{\theta_t}(-\eta_t \nabla_R f(\theta_t))$$

where $\mathcal{R}_{\theta_t}$ denotes the retraction onto space at θ and η_t denotes the learning rate at time t. Hence, to apply this type of optimization we require the Riemannian gradient (described in Appendix B.6) and a suitable retraction.

Retraction: Following Nickel & Kiela (2017) we experiment with a simple retraction:

$$\mathcal{R}_{\theta_t}(v) = \theta + v$$

C.3. Graph Reconstruction

Loss Function: To compute the embeddings, we optimize the distance-based loss function proposed in Gu et al. (2019). Given graph distances $\{d_G(X_i, X_j)\}_{ij}$ between all pairs of connected nodes, the loss is defined as:

$$\mathcal{L}(x) = \sum_{1\le i\le j\le n} \left| \left(\frac{d_{\mathcal{P}}(x_i, x_j)}{d_G(X_i, X_j)} \right)^2 - 1 \right|$$

where $d_{\mathcal{P}}(x_i, x_j)$ is the distance between the corresponding node representations in the embeddings space. This formulation of the loss function captures the average distortion. We regard as future work experimenting with different loss functions, similar to the ones proposed on Cruceru et al. (2020).

Evaluation Metrics: To measure the quality of the learned embeddings we follow the same fidelity metrics applied in Gu et al. (2019), which are distortion and precision. The distortion of a pair of connected nodes a, b in the graph G, where $f(a), f(b)$ are their respective embeddings in the space $\mathcal{P}$ is given by:

$$\text{distortion}(a, b) = \frac{|d_{\mathcal{P}}(f(a), f(b)) - d_G(a, b)|}{d_G(a, b)}$$

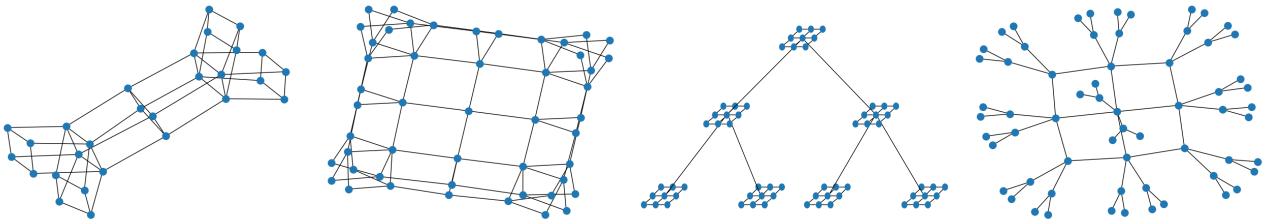

Figure 11. a) Cartesian product of tree and 2D grid. b) Cartesian product of tree and tree. c) Rooted product of tree and 2D grids. d) Rooted product of 2D grid and trees.

Graph	Nodes	Edges	Triples	Grid Layout	Tree Valency	Tree Height
4D GRID	625	2000	195,000	$(5)^4$		
TREE	364	363	66,066		3	5
TREE × GRID	496	1,224	122,760	4×4	2	3
TREE × TREE	225	420	25,200		2	3
TREE ⋄ GRIDS	775	1,270	299,925	5×5	2	4
GRID ⋄ TREES	775	790	299,925	5×5	2	4

Table 9. Synthetic graph stats

Graph	Nodes	Edges	Triples
USCA312	312	48,516	48,516
bio-diseasome	516	1,188	132,870
csphd	1,025	1,043	524,800
road-euroroad	1,039	1,305	539,241
facebook	4,039	88,234	8,154,741

Table 10. Real-world graph stats

The average distortion D_{avg} is the average over all pairs of points. Distortion is a global metric that considers the explicit value of all distances.

The other metric that we consider is the mean average precision (mAP). It is a ranking-based measure for local neighborhoods that does not track explicit distances. Let $G = (V, E)$ be a graph and node $a \in V$ have neighborhood $\mathcal{N}_a = \{b_1, ..., b_{\deg(a)}\}$, where $\deg(a)$ is the degree of a. In the embedding f, define R_{a,b_i} to be the smallest ball around $f(a)$ that contains b_i (that is, R_{a,b_i} is the smallest set of nearest points required to retrieve the i-th neighbor of a in f). Then:

$$\text{mAP}(f) = \frac{1}{|V|} \sum_{a \in V} \frac{1}{\deg(\text{a})} \sum_{i=1}^{|\mathcal{N}_a|} \frac{|\mathcal{N}_a \cap R_{a,b_i}|}{|R_{a,b_i}|}$$

Data: We employ NetworkX (Hagberg et al., 2008) to generate the synthetic datasets, and their Cartesian and rooted products. The statistics of the synthetic datasets reported in this work are presented in Table 9, and a diagram of the graphs can be seen in Figure 11.

The real-world datasets were downloaded from the Network Repository (Rossi & Ahmed, 2015). Stats are presented in Table 10.

By triples we mean the 3-tuple $(\mathbf{u}, \mathbf{v}, d(\mathbf{u}, \mathbf{v}))$, where **u, v** represent connected nodes in the graph, and d(**u,v**) is the shortest distance between them.

Setup Details: For all models and datasets we run the same grid search and optimize the distortion loss, applying RSGD. We report the average of 5 runs in all cases. The implementation of all baselines are taken from Geoopt (Kochurov et al., 2020). We train for 3000 epochs, reducing the learning rate by a factor of 5 if the model does not improve the performance after 50 epochs, and early stopping based on the average distortion if the model does not improve after 150 epochs. We use the burn-in strategy (Nickel & Kiela, 2017; Cruceru et al., 2020) training with a 10 times smaller learning rate for the first 10 epochs. We experiment with learning rates from $\{0.05, 0.01, 0.005, 0.001\}$, batch sizes from $\{512, 1024, 2048\}$ and max gradient norm from $\{10, 50, 250\}$.

Experimental Observations: We noticed that for some combinations of hyper-parameters and datasets, the learning process for the Bounded domain model becomes unstable. Points eventually fall outside of the space, and need to be projected in every epoch. We did not observe this behavior on the Siegel model. We consider that these findings are in line with the ones reported on Nickel & Kiela (2018), where they observe that the Lorentz model, since it is unbounded, is more stable for gradient-based optimization than the Poincaré one.

C.4. Recommender Systems

Setup Details: For all models and datasets we run the same grid search and optimize the Hinge loss from Equation 3, applying RSGD. We report the average of 5 runs in all cases. We train for 500 epochs, reducing the learning rate by a factor of 5 if the model does not improve

Dataset	Users	Items	Interactions	Density (%)
ml-1m	6,040	3,706	1,000,209	4.47
ml-100k	943	1,682	100,000	6.30
last.fm	1,892	17,632	92,834	0.28
meetup-nyc	46,895	16,612	277,863	0.04

Table 11. Recommender system dataset stats

Dataset	Nodes	Classes	Triples
IRIS	150	3	11,175
ZOO	101	7	5,050
GLASS	214	6	22,790

Table 12. Machine learning datasets used for node classification.

the performance after 50 epochs, and early stopping based on the dev set if the model does not improve after 150 epochs. We use the burn-in strategy (Nickel & Kiela, 2017; Cruceru et al., 2020) training with a 10 times smaller learning rate for the first 10 epochs. We experiment with learning rates from $\{0.1, 0.05, 0.01, 0.005, 0.001\}$, batch sizes from $\{1024, 2048\}$ and max gradient norm from $\{10, 50, 250\}$.

Data: We provide a brief description of the datasets used in the recommender systems experiments.

- ml-1m and ml-100k: refers to the MovieLens datasets (Harper & Konstan, 2015).[4]
- last.fm: Dataset of artist listening records from 1892 users (Cantador et al., 2011).[5]
- meetup: dataset crawled from Meetup.com, where the goal is to recommend events to users (Pham et al., 2015). The dataset consists of the data from two regions, New York City (NYC) and state of California (CA), we only report results for NYC.

Stats for the datasets are presented in Table 11.

To generate evaluation splits, the penultimate and last item the user has interacted with are withheld as dev and test set respectively.

C.5. Node Classification

Setup Details: In these experiments, for all datasets we use the cosine distance on the datapoints' features to compute a complete input distance graph. We employ the available features and normalize them so that each attribute has mean zero and standard deviation one. Once we have a graph, we embed it in the exact same way than in the graph reconstruction task. Finally, we use the learned node embeddings as features to feed a logistic regression classifier

Matrix Mapping: 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 upper half-space model of dimension n, we apply the following mapping. From each complex matrix embedding $Z = X + iY$ we stack the result of the following operations in matrix form as:

$$M = \begin{pmatrix} Y + XY^{-1}X & XY^{-1} \\ Y^{-1}X & Y^{-1} \end{pmatrix}$$

where $M \in \mathbb{R}^{2n \times 2n}$. This mapping is the natural realisation of HypSPD_n as a totally geodesic submanifold of SPD_{2n}. Since $M \in \text{SPD}_{2n}$, finally we apply the LogEig map as proposed by Huang & Gool (2017), which yields a representation in a flat space. This operations results in new matrix of the form:

$$\text{LogEig}(M) = \begin{pmatrix} U & V \\ V & -U \end{pmatrix}$$

where $U, V \in Sym(n)$. The final step is to take the upper triangular from U and V, and concatenate them as a vector of $n(n+1)$ dimensions.

This procedure is implemented for the Upper half-space. In the case of the Bounded domain model, we first map the points to the upper half-space with the Cayley transform.

Datasets: All datasets were downloaded from the UCI Machine Learning Repository (Dua & Graff, 2017). [6] Statistics about the datasets used are presented in Table 12.

C.6. Learning the Weights for Finsler Distances

Both Finsler metrics F_1 and F_∞ exhibit outstanding results in our experiments. However, there are significant differences in their relative performances depending on the target dataset. F_1 and F_∞ are two metrics among many variants in the family of Finsler metrics. Thus, we propose an alternative of our Finslerian models, by learning weights for the summation of Algo 1, step 5, according to: $d^{F_W}(Z_1, Z_2) : \sum_{i=1}^{n} [\alpha_i \log(1+d_i/1-d_i)]$ where $\alpha_i \in \mathbb{R}$ are model parameters. The intuition behind this variant is to impose an inductive bias through the family of Finsler metrics, while allowing the model to learn from the data which particular metric is more suitable in each case. The model has flexibility to represent F_1 or F_∞, and also explore different variations, such as Finsler metrics of minimum entropy

[4] https://grouplens.org/datasets/movielens/

[5] https://grouplens.org/datasets/hetrec-2011/

[6] https://archive.ics.uci.edu/ml/datasets.php

	4D GRID		TREE × TREE		TREE ⋄ GRIDS	
	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathcal{S}_4^{F_\infty}$	5.92	99.61	3.31	99.95	10.88	63.52
$\mathcal{S}_4^{F_1}$	0.01	**100.00**	**1.08**	**100.00**	1.03	78.71
$\mathcal{S}_4^{F_W}$	**0.00**	**100.00**	1.09	**100.00**	**0.91**	**93.37**

Table 13. Comparison with learning weights for Finsler metrics.

(Boland & Newberger, 2001). We report results in Table 13. We observe that the model recovers the F_1 metric in the cases where it is the most convenient, whereas for TREE ⋄ GRIDS, it finds a more optimal solution.

D. Distance Algorithm Complexity

D.1. Theoretical Complexity

In this section we discuss the computational theoretical complexity of the different operations involved in the development of this work. We employ Big O notation[7]. Since in all cases operations are not nested, but are applied sequentially, the costs can be added resulting in a polynomial expression. Thus, by applying the properties of the notation, we disregard lower-order terms of the polynomial.

Real Matrix Operations: For $n \times n$ matrices with real entries, the associated complexity of each operation is as follows:[8]

- Addition and subtraction: $\mathcal{O}(n^2)$
- Multiplication: $\mathcal{O}(n^{2.4})$
- Inversion: $\mathcal{O}(n^{2.4})$
- Diagonalization: $\mathcal{O}(n^3)$

Complex Matrix Operations: A complex symmetric matrix $Z \in \mathrm{Sym}(n, \mathbb{C})$ can be written as $Z = X + iY$, where $X = \Re(Z), Y = \Im(Z) \in \mathrm{Sym}(n, \mathbb{R})$ are symmetric matrices with real entries. We implement the elemental operations for these matrices with the following associated costs:

- Multiplication: $\mathcal{O}(n^{2.4})$. It involves 4 real matrix multiplications, plus additions and subtractions.
- Square root: $\mathcal{O}(n^3)$. It involves a diagonalization and 2 matrix multiplications.[9]
- Inverse: $\mathcal{O}(n^{2.4})$. It involves real matrix inversions and multiplications (Falkenberg, 2007).

Takagi Factorization: This factorization involves complex and real multiplications ($\mathcal{O}(n^{2.4})$), and diagonalizations ($\mathcal{O}(n^3)$). It also involves the diagonalization of a $2nx2n$ matrix, which implies:

$$\mathcal{O}((2n)^3) = \mathcal{O}(8n^3) \simeq \mathcal{O}(n^3) \tag{7}$$

Therefore, the final boundary for its cost is $\mathcal{O}(n^3)$.

Cayley Transform: This operation along with its inverse are composed of matrix inversions and multiplications, thus the cost is $\mathcal{O}(n^{2.4})$.

Distance Algorithm: The full computation of the distance algorithm in the **upperhalf space** involves matrix square root, multiplications, inverses, and the application of the Cayley transform and the Takagi factorization. Since they are applied sequentially, without affecting the dimensionality of the matrices, we can take the highest value as the asymptotic cost of the algorithm, which is $\mathcal{O}(n^3)$.

For the **bounded domain**, the matrices are mapped into the upperhalf space by an additional application of the inverse Cayley transform, and then the same distance algorithm is applied. Therefore, in this space the complexity also converges to $\mathcal{O}(n^3)$.

D.2. Empirical Complexity

To empirically measure the time involved in the distance calculation we generate a batch of 1024 pairs of points ($n \times n$ matrices). We perform the time evaluation for different values of n. Results can be observed in Figure 12.

We observe that as we increase the dimensionality, the relation tends to be polynomial, in line with the theoretical cost stated in §4.

E. Vector-valued Distance: a Tool for the Analysis of Embeddings

The vector-valued distance can also be used to develop other tools to analyze the embedding. More specifically we use it not only to create a **continuous edge coloring** as in Section 6, but also a **vector distance plot**, and a **continuous node coloring with respect to a root**.

For the **vectorial distance plot** we sample pairs of connected vertices of the graph $\{z_i, z_j\}$ and plot the result of $\mathrm{vDist}(Z_i, Z_j) = (v_1, v_2)$ (see Algorithm 1, step 6). In Figure 13-20 we show the plots of (v_1, v_2) for the embeddings of different dataset embedded into the Upper Half models with respect to Riemannian, F_1 and F_∞ metrics. In the F_1

[7] https://en.wikipedia.org/wiki/Big_O_notation

[8] https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations

[9] https://en.wikipedia.org/wiki/Square_root_of_a_matrix

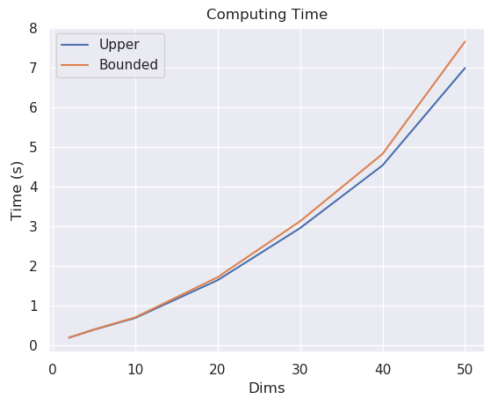

Figure 12. Time of distance calculation for a batch of 1024 pairs of points for different matrix dimensions.

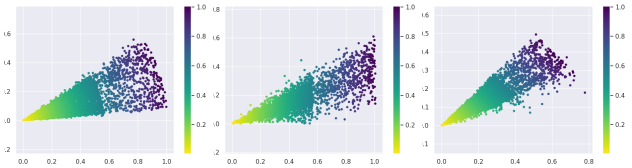

Figure 13. Plot of (v_1, v_2) of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F1}$ (center), and $\mathcal{S}_2^{F\infty}$ (right) for vertex pairs sampled from USCA312. Color indicates ground-truth distance.

Figure 14. Plot of (v_1, v_2) of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F1}$ (center), and $\mathcal{S}_2^{F\infty}$ (right) for vertex pairs sampled from BIO-DISEASOME. Color indicates ground-truth distance.

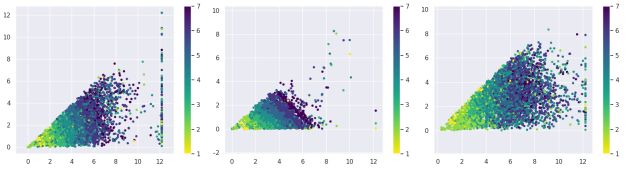

Figure 15. Plot of (v_1, v_2) of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F1}$ (center), and $\mathcal{S}_2^{F\infty}$ (right) for vertex pairs sampled from CSPHD. Color indicates ground-truth distance.

case, the addition of both d-values sums up to the distance, whereas for the F_∞, the largest v (v_1) corresponds to the distance. The plots match the ℓ^1 and ℓ^∞ metrics from Figure 2, verifying the intuition about the distances.

The vectorial distance plots give a first qualitative visualization of the embedding. For dissimilar data sets, the edge plots look quite different. They can accumulate near the boundary of the cone (the diagonal or the horizontal), or be evenly distributed. The can also be refined by sampling only specific edges of the graph.

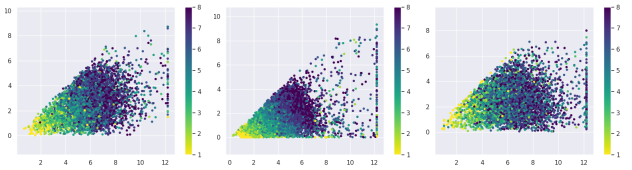

Figure 16. Plot of (v_1, v_2) of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F1}$ (center), and $\mathcal{S}_2^{F\infty}$ (right) for vertex pairs sampled from EUROROAD. Color indicates ground-truth distance.

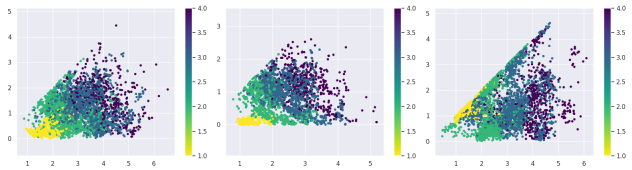

Figure 17. Plot of (v_1, v_2) of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F1}$ (center), and $\mathcal{S}_2^{F\infty}$ (right) for vertex pairs sampled from GRID4D. Color indicates ground-truth distance.

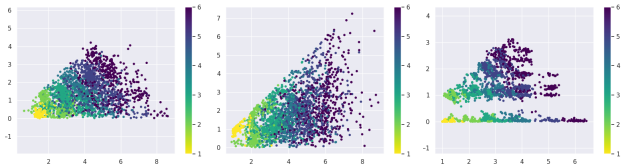

Figure 18. Plot of (v_1, v_2) of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F1}$ (center), and $\mathcal{S}_2^{F\infty}$ (right) for vertex pairs sampled from TREE $\times$ GRID. Color indicates ground-truth distance.

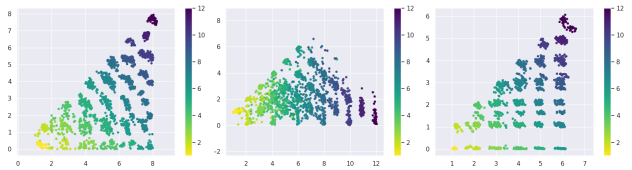

Figure 19. Plot of (v_1, v_2) of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F1}$ (center), and $\mathcal{S}_2^{F\infty}$ (right) for vertex pairs sampled from TREE $\times$ TREE. Color indicates ground-truth distance.

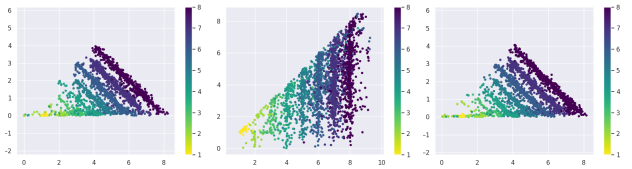

Figure 20. Plot of (v_1, v_2) of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F1}$ (center), and $\mathcal{S}_2^{F\infty}$ (right) for vertex pairs sampled from TREE. Color indicates ground-truth distance.

To construct a **continuous node coloring** we choose one vertex of our graph as the root r. For every other node z we take the vector-valued distance from r to z, and assign the ratio v_2/v_1 as a value for this node. We again represent the corresponding real number by a color shading, as in Figures 21. It can be thought as the accumulated angle over a path from the root r to the node z.

Evaluating the Quality of Embeddings: In the case of synthetic graphs, where we have full knowledge of the internal structure, we can use the edge and the node angles to compare the embeddings with respect to the Riemannian distance and the Finsler metrics F_1 and F_∞. Illustrating this with the two dimensional grid, we observe in Figure 21 that while in the Finsler metric all edges have the same angle, the embedding optimizing the Riemannian distance is more distorted and less geodesics. In the case of the Finsler distances one can also see more clearly that the symmetries of the graph are respected in the embedding. This shows that the Finsler embeddings are much better in representing structural features of the graphs than the Riemannian embeddings.

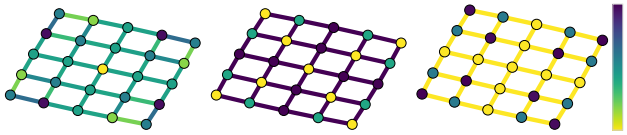

Figure 21. Analysis of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F_\infty}$ (center), and $\mathcal{S}_2^{F1}$ (right) for a 5×5 grid. Node colors indicate the angle of the vector-valued distance by taking the path from the central node. Edge colors indicate the angle for each edge.

More Edge Coloring: We plot the edge coloring for the three analyzed metric spaces, namely $\mathcal{S}_2^R, \mathcal{S}_2^{F_\infty}, \mathcal{S}_2^{F1}$ for the datasets analyzed in Figure 4, and for CSPHD in Figure 22-25. We can observe that in the Riemannian metric plots (left-hand side) there is no clear pattern that separates flat and hierarchical components in the graphs. The F_∞ and $F1$ metrics are the best at capturing the structural aspect of the datasets. They recognize very similar patterns, though they assign opposite angles to the vector-valued distance vectors, and this can be noticed from the fact that the colors assigned are in opposite sides of the spectrum (yellow means angles close to zero, blue means angles close to 45°).

To plot these visualizations and the ones of Figure 5, we adapted code released[10] by Cruceru et al. (2020).

[10] https://github.com/dalab/matrix-manifolds/blob/master/analysis/plot_ricci_curv.py

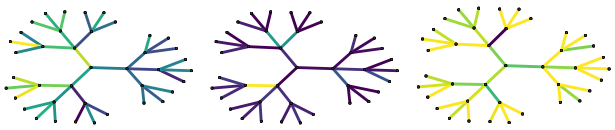

Figure 22. Edge coloring of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F_\infty}$ (center), and $\mathcal{S}_2^{F1}$ (right) for a tree.

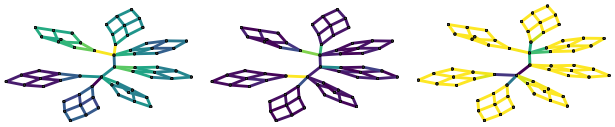

Figure 23. Edge coloring of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F_\infty}$ (center), and $\mathcal{S}_2^{F1}$ (right) for a TREE ⋄ GRIDS.

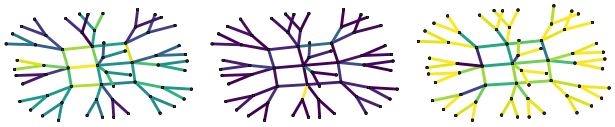

Figure 24. Edge coloring of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F_\infty}$ (center), and $\mathcal{S}_2^{F1}$ (right) for a GRID ⋄ TREES.

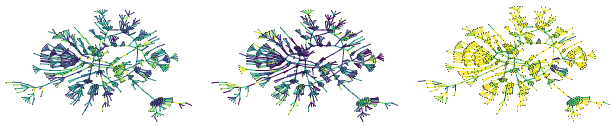

Figure 25. Edge coloring of $\mathcal{S}_2^R$ (left), $\mathcal{S}_2^{F_\infty}$ (center), and $\mathcal{S}_2^{F1}$ (right) for CSPHD.

F. More Results

Results for graph reconstruction in lower dimensions are presented in Table 14

$(\lvert V\rvert, \lvert E\rvert)$	4D GRID $(625, 2000)$		TREE $(364, 363)$		TREE × GRID $(496, 1224)$		TREE × TREE $(225, 420)$		TREE ⋄ GRIDS $(775, 1270)$		GRID ⋄ TREES $(775, 790)$	
	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathbb{E}^{12}$	11.24±0.00	**100.00**	5.71±0.01	32.72	9.80±0.00	83.25	9.79±0.00	95.97	5.11±0.05	22.24	5.48±0.03	21.84
$\mathbb{H}^{12}$	25.23±0.06	63.86	2.09±0.28	97.32	17.12±0.00	83.29	20.55±0.12	75.98	14.12±0.45	44.06	14.76±0.23	31.96
$\mathbb{E}^6 \times \mathbb{H}^6$	11.24±0.00	**100.00**	1.61±0.07	**100.00**	9.20±0.03	**100.00**	9.34±0.05	98.14	2.53±0.07	58.86	2.38±0.04	**97.56**
$\mathbb{H}^6 \times \mathbb{H}^6$	18.76±0.02	79.05	**0.92±0.04**	99.95	12.92±0.86	89.71	9.71±2.47	96.82	**1.32±0.08**	**72.62**	3.10±0.62	86.40
$\mathcal{S}_3^R$	13.26±0.01	99.54	1.69±0.03	71.64	9.26±0.01	99.57	8.80±0.21	97.47	1.82±0.07	64.52	2.27±0.18	79.10
$\mathcal{S}_3^{F_\infty}$	11.82±0.03	98.71	1.35±0.39	99.35	7.98±0.66	99.47	3.97±0.34	99.64	13.01±0.64	55.89	11.26±0.59	68.30
$\mathcal{S}_3^{F_1}$	**6.41±0.00**	**100.00**	1.07±0.04	74.98	**2.02±0.02**	**100.00**	**1.84±0.02**	**100.00**	1.43±0.01	65.90	**1.45±0.05**	81.25
$\mathcal{B}_3^R$	13.29±0.17	99.54	1.63±0.06	70.70	10.04±0.03	92.20	9.10±0.10	96.78	4.71±0.15	65.04	5.68±0.37	89.19
$\mathcal{B}_3^{F_\infty}$	12.45±0.18	97.70	2.59±0.34	98.94	10.33±0.47	93.58	4.74±0.00	96.66	11.33±0.10	65.07	10.39±0.15	79.43
$\mathcal{B}_3^{F_1}$	**6.41±0.00**	**100.00**	1.13±0.03	79.21	**2.02±0.00**	**100.00**	1.92±0.07	**100.00**	1.51±0.06	71.07	1.51±0.00	83.64

Table 14. Results for synthetic datasets. All models have same number of free parameters. Lower D_{avg} is better. Higher mAP is better.