Computationally Tractable Riemannian Manifolds for Graph Embeddings

Calin Cruceru[1], **Gary Bécigneul**[1,2], **Octavian-Eugen Ganea**[1,2]
[1] Department of Computer Science, ETH Zürich, Switzerland
[2] Computer Science and Artificial Intelligence Lab, MIT, USA
ccruceru@inf.ethz.ch, {garyb,oct}@mit.edu

Abstract

Representing graphs as sets of node embeddings in certain curved Riemannian manifolds has recently gained momentum in machine learning due to their desirable geometric inductive biases (e.g., hierarchical structures benefit from hyperbolic geometry). However, going beyond embedding spaces of constant sectional curvature, while potentially more representationally powerful, proves to be challenging as one can easily lose the appeal of computationally tractable tools such as geodesic distances or Riemannian gradients. Here, we explore two computationally efficient matrix manifolds, showcasing how to learn and optimize graph embeddings in these Riemannian spaces. Empirically, we demonstrate consistent improvements over Euclidean geometry while often outperforming hyperbolic and elliptical embeddings based on various metrics that capture different graph properties. Our results serve as new evidence for the benefits of non-Euclidean embeddings in machine learning pipelines.

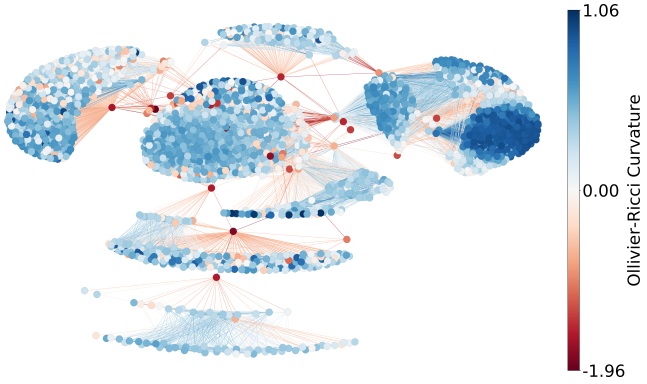


Figure 1: A dense social network from Facebook (Leskovec and Mcauley 2012) used in our experiments. It shows the Ollivier-Ricci curvatures of edges and their averages for nodes. More such drawings are included in Appendix H.

1 Introduction

Before representation learning started gravitating around deep representations (Bengio et al. 2009) in the last decade, a line of research that sparked interest in the early 2000s was based on the so called manifold hypothesis (Bengio, Courville, and Vincent 2013). According to it, real-world data given in their raw format (e.g., pixels of images) lie on a low-dimensional manifold embedded in the input space. At that time, most manifold learning algorithms were based on locally linear approximations to points on the sought manifold – LLE (Roweis and Saul 2000), Isomap (Tenenbaum, De Silva, and Langford 2000) – or on spectral methods – MDS (Hofmann and Buhmann 1995), graph Laplacian eigenmaps (Belkin and Niyogi 2002).

Back to recent years, two trends are apparent: (i) the use of graph-structured data and their direct processing by machine learning algorithms (Bruna et al. 2014; Henaff, Bruna, and LeCun 2015; Grover and Leskovec 2016), and (ii) the resurgence of the manifold hypothesis, but with a different flavor – being explicit about the assumed manifold and, perhaps, the inductive bias that it entails: hyperbolic spaces (Nickel and Kiela 2017; Ganea, Becigneul, and Hofmann 2018), spherical spaces (Wilson et al. 2014), and Cartesian products of them (Gu et al. 2018). While for the first two the choice can be a priori justified – e.g., complex networks are intimately related to hyperbolic geometry (Krioukov et al. 2010) – the last one is motivated through the presumed flexibility coming from its varying curvature. Our work takes that hypothesis further by exploring the representation properties of several *irreducible spaces*[1] of non-constant sectional curvature. We use, in particular, Riemannian manifolds where points are represented as specific types of matrices and which are at the sweet spot between semantic richness and tractability.

With no additional qualifiers, *graph embedding* is a vaguely specified intermediary step used as part of systems solving a wide range of graph analytics problems such as link prediction (Trouillon et al. 2016; Zhang and Chen 2018) and node classification (Li, Zhu, and Zhang 2016; Wang et al. 2017). What they all have in common is the representation of certain parts of a graph as points in a continuous space. As a particular instance of that general task, here we embed nodes of graphs with structural information only (i.e., undirected and without node or edge labels), as the one shown in Figure 1 in novel curved spaces, by leveraging the closed-form expressions of the corresponding Riemannian distance between

[1] Not expressible as Cartesian products of other manifolds, be they model spaces, as in (Gu et al. 2018), or yet others.

embedding points; the resulting geodesic distances enter a differentiable objective function which "compares" them to the ground-truth metric given through the node-to-node graph distances. We focus on the representation capabilities of the considered matrix manifolds relative to the previously studied spaces by monitoring graph reconstruction metrics.

Our **main contributions** include **(i)** the introduction of two families of matrix manifolds for graph embedding purposes: the non-positively curved spaces of symmetric positive definite (SPD) matrices, and the compact, non-negatively curved Grassmann manifolds; **(ii)** reviving Stochastic Neighbor Embedding (SNE) (Hinton and Roweis 2003) in the context of Riemannian optimization as a way to unify, on the one hand, the loss functions based on the reconstruction likelihood of local graph neighborhoods and, on the other hand, the global, all-pairs stress functions used for global metric recovery; **(iii)** a generalization of the usual ranking-based metric to quantify reconstruction fidelity beyond immediate neighbors; **(iv)** a comprehensive experimental comparison of the introduced manifolds against the baselines in terms of their graph reconstruction capabilities, focusing on the impact of curvature.

Related Work. Our work is inspired by the emerging field of geometric deep learning (GDL) through its use of geometry (Bronstein et al. 2017). That being said, our motivation and approach are different. In GDL, deep networks transform data in a geometry-aware way, usually as part of larger discriminative or generative models: e.g., graph neural networks (Bruna et al. 2014; Henaff, Bruna, and LeCun 2015), hyperbolic neural networks (Ganea, Bécigneul, and Hofmann 2018; Mathieu et al. 2019), hyperspherical neural networks (Defferrard, Bresson, and Vandergheynst 2016; Xu and Durrett 2018), and others (Bachmann, Bécigneul, and Ganea 2019; Skopek, Ganea, and Bécigneul 2020). We, on the other hand, embed graph nodes in a simpler, transductive setting, employing Riemannian optimization (Bonnabel 2013; Becigneul and Ganea 2019) to directly obtain the corresponding embeddings. The broader aim to which we contribute is that of understanding the role played by the space curvature in graph representation learning. In this sense, works such as those of Sarkar (2011) and Krioukov et al. (2010), who formally describe the connections between certain types of graphs (i.e., trees and complex networks, respectively) and hyperbolic geometry, inspire us: ultimately, we seek similar results for new classes of graphs and embedding spaces. This work, mostly an empirical one, is a first step in that direction for two families of matrix manifolds. It is similar in spirit to Gu et al. (2018) who empirically show that Cartesian products of model spaces can provide a good inductive bias in some cases. A non-goal of this work is comparing against all prior embedding methods that operate in the domain of vectors (e.g., Grover and Leskovec 2016; Hamilton, Ying, and Leskovec 2017; Donnat et al. 2018). Finally, the manifolds themselves are not new in the machine learning literature: recent computer vision applications take into account the intrinsic geometry of SPD matrices (Dong et al. 2017; Huang and Van Gool 2017) and Grassmann subspaces (Huang, Wu, and Van Gool 2018; Zhang et al. 2018) when building discriminative models. It is the study of the implications of their curvature for graph representations that is novel.

2 Preliminaries & Background

Notation. Let $G = (X, E, w)$ be an undirected graph, with X the set of nodes, E the set of edges, and $w : E \to \mathbb{R}_+$ the edge-weighting function. Let $m = |X|$. We denote by $d_G(x_i, x_j)$ the shortest path distance between nodes $x_i, x_j \in X$, induced by w. The node embeddings are[2] $Y = \{y_i\}_{i\in[m]} \subset \mathcal{M}$ and the geodesic distance function is $d_{\mathcal{M}}(y_i, y_j)$, with $\mathcal{M}$, the embedding space, a Riemannian manifold. $\mathcal{N}(x_i)$ denotes the set of neighbors of node x_i.

Riemannian Geometry. A comprehensive account of the fundamental concepts from Riemannian geometry is included in Appendix A. Informally, an n-dimensional manifold $\mathcal{M}$ is a space that locally resembles $\mathbb{R}^n$. Each point $x \in \mathcal{M}$ has attached a tangent space $T_x\mathcal{M}$ – a vector space that can be thought of as a first-order local approximation of $\mathcal{M}$ around x. The Riemannian metric $\langle\cdot,\cdot\rangle_x$ is a collection of inner products on these tangent spaces that vary smoothly with x. It makes possible measuring geodesic distances, angles, and curvatures. The different notions of *curvature* quantify the ways in which a surface is locally curved around a point. The exponential map is a function $\exp_x : T_x\mathcal{M} \to \mathcal{M}$ that can be seen as "folding" or projecting the tangent space onto the manifold. Its inverse is called the logarithm map, $\log_x(\cdot)$.

Learning Framework. The embeddings are learned in a simple framework in which a loss function $\mathcal{L}$ depending on the embedding points solely via the Riemannian distances between them is minimized using stochastic Riemannian optimization. In this respect, the following general property is useful (Lee 2006): for any point x on a Riemannian manifold $\mathcal{M}$ and any y in a neighborhood of x, we have[3]

$$\nabla^R_x \, d^2(x, y) = -2 \log_x(y). \tag{1}$$

Hence, as long as $\mathcal{L}$ is differentiable with respect to the (squared) distances, it will also be differentiable with respect to the embedding points. The specifics of $\mathcal{L}$ are deferred to Section 4.

Model Spaces & Cartesian Products. The model spaces of Riemannian geometry are manifolds with constant sectional curvature K: (i) Euclidean space ($K = 0$), (ii) hyperbolic space ($K < 0$), and (iii) elliptical space ($K > 0$). We summarize the Riemannian geometric tools of the last two in Appendix B. They are used as baselines in our experiments. We also recall that given a set of manifolds $\{\mathcal{M}_i\}_{i=1}^k$, the product manifold $\mathcal{M} = \times_{i=1}^k \mathcal{M}_i$ has non-constant sectional curvature and can be used for graph embedding purposes as long as each factor has efficient closed-form formulas for the quantities of interest (Gu et al. 2018).

[2] We use $i \in [m]$ as a short-hand for $i \in \{1, 2, \ldots, m\}$.

[3] ∇^R_x denotes the Riemannian gradient at x. See Appendix A.

Measuring Curvature around Embeddings. Curvature properties are central to our work since they set apart the matrix manifolds discussed in Section 3. Recall that any manifold locally resembles Euclidean space. Hence, several ways of quantifying *the actual* space curvature between embeddings have been proposed; see Appendix C for an overview. One which we find more convenient for analysis and presentation purposes, because it yields bounded and easily interpretable values, is based on sums of angles in geodesic triangles formed by triples $x, y, z \in \mathcal{M}$,

$$k_\theta(x, y, z) = \theta_{x,y} + \theta_{x,z} + \theta_{y,z}, \quad \text{with} \tag{2}$$
$$\theta_{x_1,x_2} = \cos^{-1} \frac{\langle u_1, u_2 \rangle_{x_3}}{\|u_1\|_{x_3} \|u_2\|_{x_3}}, \; u_{\{1,2\}} = \log_{x_3}(x_{\{1,2\}}).$$

It takes values in the intervals $[0, \pi]$ and $[\pi, 3\pi]$, in hyperbolic and elliptical spaces, respectively. In practice, we look at empirical distributions of $\overline{k}_\theta = (k_\theta - \pi)/2\pi$, with values in $[-0.5, 0]$ and $[0, 1]$, respectively, obtained by sampling triples (x, y, z) from an embedding set $\{y_i\}_{i=1}^k$.

3 Matrix Manifolds for Graph Embeddings

We propose two families of matrix manifolds that lend themselves to computationally tractable Riemannian optimization in our graph embedding framework.[4] They cover negative and positive curvature ranges, respectively, resembling the relationship between hyperbolic and hyperspherical spaces. Their properties are summarized in Table 1. Details and proofs are included in Appendix D.

Non-positive Curvature: SPD Manifold

The space of $n \times n$ real symmetric positive-definite matrices, $\mathcal{S}^{++}(n) := \{A \in \mathcal{S}(n) : \langle x, Ax \rangle > 0 \text{ for all } x \neq 0\}$, is an $\frac{n(n+1)}{2}$-dimensional differentiable manifold – an embedded submanifold of $\mathcal{S}(n)$, the space of $n \times n$ symmetric matrices. Its tangent space can be identified with $\mathcal{S}(n)$.

Riemannian Structure. The most common Riemannian metric endowed to $\mathcal{S}^{++}(n)$ is $\langle P, Q \rangle_A = \operatorname{Tr} A^{-1} P A^{-1} Q$. Also called the *canonical* metric, it is motivated as being invariant to congruence transformations $\Gamma_X(A) = X^\top A X$, with X an $n \times n$ invertible matrix (Pennec, Fillard, and Ayache 2006). The induced distance function is[5]

$$d(A, B) = \sqrt{\sum_{i=1}^{n} \log^2 \big(\lambda_i(A^{-1}B)\big)}. \tag{7}$$

It is equivalent to the more compact expression (5). It can be interpreted as measuring how well A and B can be simultaneously reduced to the identity matrix (Chossat and Faugeras 2009).

[4]Counterexamples include the low-rank and the (compact) Stiefel manifolds, which lack closed-form distance functions.

[5]We use $\lambda_i(X)$ to denote the ith eigenvalue of X in some arbitrary but fixed order (or when the order is not important).

Properties. The canonical SPD manifold has non-positive sectional curvature everywhere (Bhatia 2009). It is also a high-rank symmetric space (Lang 2012). The high-rank property tells us that there are *at least planes* of the tangent space on which the sectional curvature vanishes. Contrast this with the hyperbolic space which is also a symmetric space but where the only (intrinsic) flats are the geodesics. At the same time, and still in contrast, the sectional curvatures of the SPD manifold at each point are not bounded from below. Moreover, only one degree of freedom can be factored out of the manifold $\mathcal{S}^{++}(n)$: it is isometric to $\mathcal{S}_*^{++}(n) \times \mathbb{R}$, with $\mathcal{S}_*^{++}(n) := \{A \in \mathcal{S}^{++}(n) : \det(A) = 1\}$, an irreducible manifold (Dolcetti and Pertici 2018). Hence, $\mathcal{S}^{++}$ achieves a mix of flat and negatively-curved areas that cannot be obtained via other Riemannian Cartesian products.

Alternative Metric. A popular function that is commonly used in lieu of the squared canonical distance is the *symmetric Stein divergence*,

$$S(A, B) := \log\det\Big(\frac{A+B}{2}\Big) - \frac{1}{2}\log\det(AB). \tag{8}$$

It has been thoroughly studied in (Sra 2012; Sra and Hosseini 2015) who prove that $\sqrt{S}$ is a metric and that $S(A, B)$ shares many properties of the Riemannian distance function (5) such as congruence and inversion invariances as well as geodesic convexity in each argument. It is particularly appealing for backpropagation-based training due to its computationally efficient gradients (see below). Hence, we experiment with it too when matching graph metrics. We note that, although a valid squared-metric, S does not respect the geometry of the SPD manifold (in the sense of identity (1), for instance), but that is not a problem as our approach to embedding only necessitates a measure of dissimilarity between data points that is differentiable.

Computational Aspects. We compute gradients via automatic differentiation (Paszke et al. 2017). Nonetheless, notice that if $A = UDU^\top$ is the eigendecomposition of a symmetric matrix with distinct eigenvalues and $\mathcal{L}$ is some loss function that depends on A only via D, then $\frac{\partial \mathcal{L}}{\partial A} = U \frac{\partial \mathcal{L}}{\partial D} U^\top$ (Giles 2008). Computing geodesic distances requires the eigenvalues of $A^{-1}B$, though, which may not be symmetric. We overcome that by using the matrix $A^{-1/2} B A^{-1/2}$ instead which is SPD and has the same spectrum. Moreover, for the 2×2 and 3×3 cases, we use closed-form eigenvalue formulas to speed up our implementation.[6] For the Stein divergence, the gradients can be computed in closed form as $\nabla_A S(A, B) = \frac{1}{2}(A + B)^{-1} - \frac{1}{2}A^{-1}$. We additionally note that many of the required matrix operations can be efficiently computed via Cholesky decompositions (Appendix D).

Non-negative Curvature: Grassmann Manifold

The orthogonal group $O(n)$ is the set of $n \times n$ real orthogonal matrices. It is a special case of the compact Stiefel manifold

[6]This could be done in theory for $n \leqslant 4$ – a consequence of the Abel-Ruffini theorem from algebra. However, for $n = 4$ the formulas are outperformed by numerical eigenvalue algorithms.

Property	Expression	SPD $\mathcal{S}^{++}(n)$	Grassmann $Gr(k,n)$
Dimension	$\dim(\mathcal{M})$	$n(n+1)/2$	$k(n-k)$
Tangent space	$T_A\,\mathcal{M}$	$\{A \in \mathbb{R}^{n\times n} : A = A^\top\}$	$\{P \in \mathbb{R}^{n\times k} : A^\top P = 0\}$
Projection	$\pi_A(P')$	$(P' + P'^\top)/2$	$(\mathrm{Id}_n - AA^\top)P'$
Riem. metric	$\langle P, Q\rangle_A$	$\mathrm{Tr}\, A^{-1}PA^{-1}Q$ (3)	$\mathrm{Tr}\, P^\top Q$ (4)
Riem. gradient	∇^R_A	$A\,\pi_A(\nabla^E_A)\,A$	$\pi_A(\nabla^E_A)$
Geodesic	$\gamma_{A;P}(t)$	$A\exp(tA^{-1}P)$	$[AV \quad U]\,[\cos(t\Sigma) \quad \sin(t\Sigma)]\,V^\top$ with $P = U\Sigma V^\top$
Retraction	$R_A(P)$	$A + P + \frac{1}{2}PA^{-1}P$	$UV^\top$ with $A + P = U\Sigma V$
Log map	$\log_A(B)$	$A\log(A^{-1}B)$	$U\Sigma V^\top$ with $\begin{bmatrix} A^\top B \\ (\mathrm{Id}_n - AA^\top)B \end{bmatrix} = \begin{bmatrix} V\cos(\Sigma)V^\top \\ U\sin(\Sigma)V^\top \end{bmatrix}$
Riem. distance	$d(A,B)$	$\big\|\log(A^{-1}B)\big\|_F$ (5)	$\sqrt{\sum_{i=1}^k \theta_i^2}$ with $A^\top B = U\,\mathrm{diag}\big(\cos(\theta_i)\big)V^\top$ (6)

Table 1: Summary of Riemannian geometric tools for the SPD (Bhatia 2009) and Grassmann (Edelman, Arias, and Smith 1998; Zhang et al. 2018) manifolds. Notation: A, B – manifold points; P, Q – tangent space points; P' – ambient space point; $\exp(A)$ / $\log(A)$ – matrix exponential / logarithm.

$V(k,n) := \{A \in \mathbb{R}^{n\times k} : A^\top A = \mathrm{Id}_k\}$, i.e., the set of $n \times k$ "tall-skinny" matrices with orthonormal columns, for $k \leqslant n$. The Grassmannian is defined as the space of k-dimensional linear subspaces of $\mathbb{R}^n$. It is related to the Stiefel manifold in that every orthonormal k-frame in $\mathbb{R}^n$ spans a k-dimensional subspace of the n-dimensional Euclidean space. Similarly, every such subspace admits infinitely many orthonormal bases. This suggests the identification of the Grassmann manifold $Gr(k,n)$ with the quotient space $V(k,n)/O(k)$ (more about quotient manifolds in Appendix A). In other words, an $n \times k$ orthonormal matrix $A \in V(k,n)$ represents the equivalence class $[A] = \{AQ_k : Q_k \in O(k)\} \cong \mathrm{span}(A)$, which is a single point on $Gr(k,n)$.

Riemannian Structure. The canonical Riemannian metric of $Gr(k,n)$ is simply the Frobenius inner product (4). We refer to (Edelman, Arias, and Smith 1998) for details on how it arises from its quotient geometry. The closed form formula for the Riemannian distance, shown in (6), depends on the set $\{\theta_i\}_{i=1}^k$ of so-called principal angles between two subspaces. They can be interpreted as the minimal angles between all possible bases of the two subspaces (Zhang et al. 2018).

Properties. The Grassmann manifold $Gr(k,n)$ is a compact, non-negatively curved manifold. As shown in (Wong 1968), its sectional curvatures at $A \in Gr(k,n)$ satisfy $K_A(P,Q) = 1$ (for $k = 1,\ n > 2$) and $0 \leqslant K_A(P,Q) \leqslant 2$ (for $k > 1,\ n > k$), for all $P, Q \in T_A\,Gr(k,n)$. Contrast this with the constant positive curvature of the sphere which can be made arbitrarily large by making $R \to 0$.

Computational Aspects. Computing a geodesic distance requires the SVD decomposition of an $k \times k$, matrix which can be significantly smaller than the manifold dimension $k(n-k)$. For $k = 2$, we use closed-form solutions for singular values. See Appendix D for details. Otherwise, we employ standard numerical algorithms. For the gradients, a result analogous to the one for eigenvalues from earlier (Giles 2008) makes automatic differentiation straight-forward.

4 Decoupling Learning and Evaluation

Recall that our goal is to preserve the graph structure given through its node-to-node shortest paths by minimizing a loss which encourages similar *relative*[7] geodesic distances between node embeddings. Recent related work broadly uses local or global loss functions that focus on either close neighborhood information or all-pairs interactions, respectively. The methods that fall under the former emphasize correct placement of immediate neighbors, such as the one used in (Nickel and Kiela 2017) for unweighted graphs:

$$\mathcal{L}_{\text{neigh}}(Y) = -\sum_{(i,j)\in E} \log \frac{\exp\big(-d_{\mathcal{M}}(y_i,y_j)\big)}{\sum_{k\in\mathcal{N}(i)} \exp\big(-d_{\mathcal{M}}(y_i,y_k)\big)}. \tag{9}$$

Those that fall under the latter, on the other hand, compare distances directly via loss functions inspired by generalized MDS (Bronstein, Bronstein, and Kimmel 2006), e.g.,[8]

$$\mathcal{L}_{\text{stress}}(Y) \quad = \sum_{i<j} \Big(d_G(x_i,x_j) - d_{\mathcal{M}}(y_i,y_j)\Big)^2,$$

$$\text{or } \mathcal{L}_{\text{distortion}}(Y) = \sum_{i<j} \left|\frac{d^2_{\mathcal{M}}(y_i,y_j)}{d^2_G(x_i,x_j)} - 1\right|. \tag{10}$$

[7] An embedding satisfying $d_{\mathcal{M}}(y_i,y_j) = \alpha\, d_G(x_i,x_j)$ (for all $i,j \in [m]$), for $\alpha > 0$, should be perfect.

[8] Note that $\mathcal{L}_{\text{stress}}$ focuses mostly on distant nodes while $\mathcal{L}_{\text{distortion}}$ yields larger values when close ones are misrepresented. The latter is one of several objectives used in (Gu et al. 2018) (as per their code and private correspondence).

The two types of objectives yield embeddings with different properties. It is thus not surprising that each one of them has been coupled in prior work with a preferred metric quantifying reconstruction fidelity. The likelihood-based one is evaluated via the popular rank-based mean average precision (mAP), while the global, stress-like ones yield best scores when measured by the average distortion (AD) of the reference metric. See, e.g., (De Sa et al. 2018) for their definitions.

To decouple learning and evaluation, as well as to get both fairer and more informative comparisons between embeddings spaces, we propose to optimize another loss function that allows *explicitly* moving in a continuous way on the representation scale ranging from "local neighborhoods patching," as encouraged by (9), to the global topology matching, as measured by those from (10). Similarly, we propose a more fine-grained ranking metric that makes the trade-off clearer.

RSNE – Unifying Two Disparate Regimes. We advocate training embeddings via a version of the celebrated Stochastic Neighbor Embedding (SNE) (Hinton and Roweis 2003) adapted to the Riemannian setting. SNE works by attaching to each node a distribution defined over all other nodes and based on the distance to them. This is done for both the input graph distances, yielding the ground truth distribution, and for the embedding distances, yielding the model distribution. That is, with $j \neq i$, we have

$$p_{ij} := p(x_j \mid x_i) = \frac{\exp\big(-d_G^2(x_i, x_j)/T\big)}{Z_{p_i}}$$

$$\text{and } \; q_{ij} := q(y_j \mid y_i) = \frac{\exp\big(-d_{\mathcal{M}}^2(y_i, y_j)\big)}{Z_{q_i}}, \qquad (11)$$

where Z_{p_i} and Z_{q_i} are the normalizing constants and T is the input scale parameter. The original SNE formulation uses $\mathcal{M} = \mathbb{R}^n$. In this case, the probabilities are proportional to an isotropic Gaussian $\mathcal{N}(y_j \mid y_i, T)$. As defined above, it is our (natural) generalization to Riemannian manifolds – RSNE. The embeddings are then learned by minimizing the sum of Kullback-Leibler (KL) divergences between $p_i := p(\cdot \mid x_i)$ and $q_i := q(\cdot \mid y_i)$: $\mathcal{L}_{\text{SNE}}(Y) := \sum_{i=1}^{m} \mathrm{D}_{\mathrm{KL}}\left[p_i \parallel q_i\right]$. For $T \to 0$, it is easy to show that it recovers the *local neighborhood* regime from (9). For a large T, the SNE objective tends towards placing equal emphasis on the relative distances between all pairs of points, thus behaving similar to the MDS-like loss functions (10) (Hinton and Roweis 2003, Section 6). What we have gained is that the temperature parameter acts as a knob for controlling the optimization goal.

F1@k – Generalizing Ranking Fidelity. We generalize the ranking fidelity metric in the spirit of mAP@k (e.g., (Gu et al. 2018)) for nodes that are k hops away from a source node, with $k > 1$. Recall that the motivation stems, for one, from the limitation of mean average precision to immediate neighbors, and, at the other side of the spectrum, from the sensitivity to absolute values of non-ranking metrics such as the average distortion. For an unweighted[9] graph G, we denote by $L_G(u; k)$ the set of nodes that are exactly k hops away from a source node u (i.e., "on layer k"), and by $\mathcal{B}_G(v; u)$ the set of nodes that are closer to node u than another node v. Then, for an embedding $f : G \to \mathcal{M}$, the *precision* and *recall* of a node v in the shortest-path tree rooted at u, with $u \neq v$, are given by

$$P(v; u) := \frac{|\mathcal{B}_G(v; u) \cap \mathcal{B}_{\mathcal{M}}(f(v); f(u))|}{|\mathcal{B}_{\mathcal{M}}(f(v); f(u))|}$$

$$\text{and } \; R(v; u) := \frac{|\mathcal{B}_G(v; u) \cap \mathcal{B}_{\mathcal{M}}(f(v); f(u))|}{|\mathcal{B}_G(v; u)|}. \qquad (12)$$

They follow the conventional definitions. For instance, the numerator is the number of true positives: the nodes that appear before v in the shortest-path tree rooted at u and, at the same time, are embedded closer to u than v is. The definition of the F1 score of (u, v), denoted by $F_1(v; u)$, follows naturally as the harmonic mean of precision and recall. Then, the F1@k metric is obtained by averaging the F1 scores of all nodes that are on layer $k \geqslant 1$, across all shortest-path trees. That is, with $c(k) = \sum_{u \in G} |L_G(u; k)|$,

$$F_1(k) := \frac{1}{c(k)} \sum_{u \in G} \sum_{v \in L_G(u;k)} F_1(v; u). \qquad (13)$$

This draws a curve $\{(k, F_1(k))\}_{k \in [d(G)]}$, where $d(G)$ is the diameter of the graph.

5 Experiments

We restrict our experiments here to evaluating the graph reconstruction capabilities of the proposed matrix manifolds relative to the constant curvature baseline spaces. A thorough analysis via properties of nearest-neighbor graphs constructed from points picked randomly from the manifolds, inspired by (Krioukov et al. 2010), is included in Appendix E. It shows that the two matrix manifolds often lead to distinctive network structures.

Training Details & Evaluation. We start by computing all-pairs shortest-paths in all input graphs, performing max-scaling, and serializing them to disk. Then, for each manifold, we optimize a set of embeddings for several combinations of optimization settings and loss functions, including both the newly proposed Riemannian SNE, for several values of T, and the ones used in prior work (Section 4). Finally, because we are ultimately interested in the representation power of each embedding space, we report the best F1@1, area under the F1@k curve (AUC), and average distortion (AD), across those repetitions. This is in line with the experimental framework from prior works (Nickel and Kiela 2017; Gu et al. 2018) with the added benefit of treating the objective function as a nuisance. Training times are about two times longer due to the more complex operations involved in computing pairwise distances. More details are included in Appendix F.

Synthetic Graphs. We begin by showcasing the RSNE objective and the F1@k metric for several generated grap (Figure 2). On the $10 \times 10 \times 10$ grid and the 1000-nodes cycle

[9] We have mostly used unweighted graphs in our experiments, so here we restrict the treatment as such.

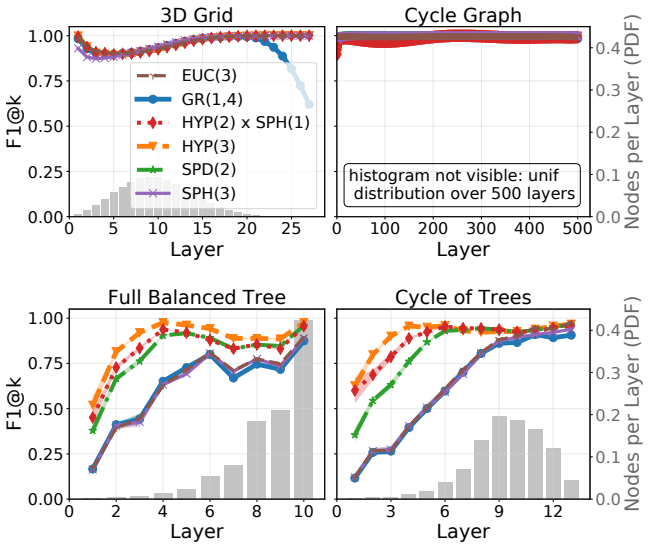


Figure 2: F1@k curves (left y-axis) and PMFs of node pairs per hop count (right y-axis) for several synthetic graphs. The objective was RSNE at high temperature T.

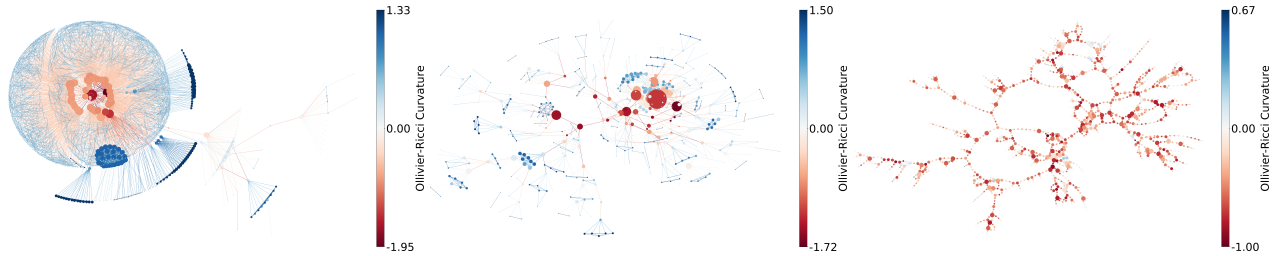
all manifolds perform well. This is because every Riemannian manifold generalizes Euclidean space and Euclidean geometry suffices for grids and cycles (e.g., a cycle looks locally like a line). The more discriminative ones are the two other graphs – a full balanced tree (branching factor $r = 4$ and depth $h = 5$) and a cycle of 10 trees ($r = 3$ and $h = 4$). The best performing embeddings involve a hyperbolic component while the SPD ones rank between those and the non-negatively curved ones (which are indistinguishable). The results confirm our expectations: (more) negative curvature is useful when embedding trees. Finally, notice that the high-temperature RSNE regime used here encourages the recovery of the global structure (high AUC F1@k) more than the local neighborhoods (low individual F1@k values for small k).

Non-positive Curvature. We compare the Euclidean, hyperbolic, and SPD spaces on several real datasets in Table 2. We include visualizations of them in Figures 1 and 3. For the SPD manifold, we experiment with both the canonical distance function and the (related) S-divergence as model metrics. When performing Riemannian optimization, we use the same canonical Riemannian tools (as per Table 1). More details about the graphs and an analysis of their geometric properties are attached in Appendix G. Extended results are included in Appendix H. First of all, we see that the (partial) negative curvature of the SPD and hyperbolic manifolds is beneficial: they outperform the flat Euclidean embeddings in almost all scenarios. This can be explained by the apparent scale-free nature of the input graphs (Krioukov et al. 2010). Second, we see that especially when using the S-divergence, which we attribute to the better-behaved optimization task thanks to its geodesic convexity and stable gradients (see Section 3), the SPD embeddings achieve significant improvements on the average distortion metric and are competitive and sometimes better on the ranking metrics.

Graph	Dim	Manifold	F1@1	AUC	Avg. Dist.
facebook	3	Euc	70.28	95.27	0.193
		Hyp	71.08	95.46	0.173
		SPD	71.09	95.26	0.170
		Stein	**75.91**	95.59	**0.114**
	6	Euc	79.60	96.41	0.090
		Hyp	81.83	96.53	0.089
		SPD	79.52	96.37	0.090
		Stein	**83.95**	96.74	**0.061**
web-edu	3	Euc	29.18	87.14	0.245
		Hyp	**55.60**	**92.10**	0.245
		SPD	29.02	88.54	0.246
		Stein	48.28	90.87	**0.084**
	6	Euc	49.31	91.19	0.143
		Hyp	**66.23**	95.78	0.143
		SPD	42.16	91.90	0.142
		Stein	62.81	**96.51**	**0.043**
bio-diseasome	3	Euc	83.78	91.21	0.145
		Hyp	**86.21**	**95.72**	0.137
		SPD	83.99	91.32	0.140
		Stein	**86.70**	94.54	**0.105**
	6	Euc	93.48	95.84	0.073
		Hyp	**96.50**	**98.42**	0.071
		SPD	93.83	95.93	0.072
		Stein	94.86	97.64	**0.066**
power	3	Euc	49.34	87.84	0.119
		Hyp	**60.18**	**91.28**	**0.068**
		SPD	52.48	90.17	0.121
		Stein	54.06	90.16	0.076
	6	Euc	63.62	92.09	0.061
		Hyp	**75.02**	**94.34**	0.060
		SPD	67.69	91.76	0.062
		Stein	70.70	93.32	**0.049**

Table 2: The results for "$\mathcal{S}^{++}$ vs. $\mathbb{H}$". The better results are in bold. The "Stein manifold" is SPD trained with the Stein divergence (see text). The F1@1 and AUC metrics are multiplied by 100.

How Do the Embeddings Curve? Since any manifold locally resembles Euclidean space, it is a priori unclear to what extent its theoretical curvature is leveraged by the embeddings. To shed light on that, we employ the analysis technique based on sum-of-angles in geodesic triangles from Section 2. We recognize in Figure 4 a remarkably consistent pattern: the better performing embeddings (as per Table 2) yield more negatively-curved triangles. Notice, for instance, the collapsed box plot corresponding to the "web-edu" hyperbolic embedding **(a)**, i.e., almost all triangles sampled have sum-of-angles close to 0 . This is explained by its obvious tree-like structure (Figure 3a). Similarly, the SPD-Stein embedding of "facebook" outperforms the hyperbolic one in terms of F1@1 and that reflects in the slightly more stretched box plot **(b)**. Moreover, the pattern applies to the best average-



(a) "web-edu" (b) "bio-diseasome" (c) "road-minnesota"

Figure 3: The graphs embedded in Tables 2 and 3. The graph "facebook" is shown in Figure 1.

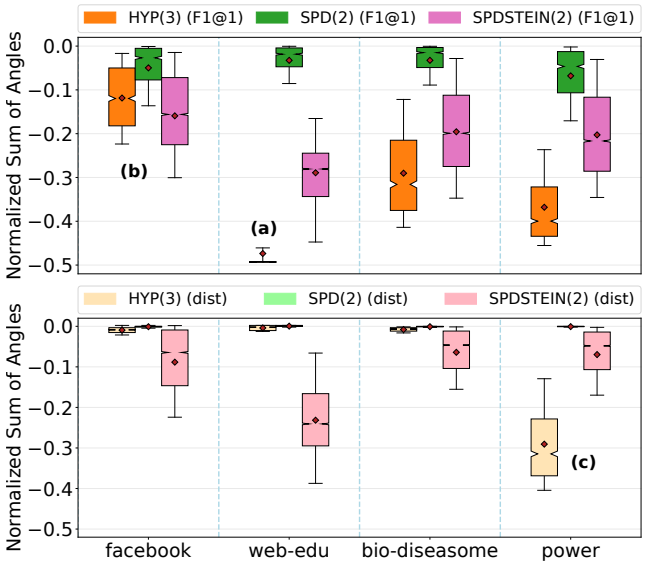


Figure 4: Distributions of (normalized) sum-of-angles in geodesic triangles formed by the learned embeddings that yield the best F1@1 metrics (up) and the best average distortion metrics (down), for all datasets from Table 2, for $n = 3$. 10000 triples are sampled.

distortion embeddings, where the SPD-Stein embeddings are the only ones that make non-negligible use of negative curvature and, hence, perform better – the only exception is the "power" graph **(c)**, for which indeed Table 2 confirms that the hyperbolic embeddings are slightly better.

Compact Embeddings. We embed several graphs with traits associated with positive curvature in Grassmann manifolds and compare them to spherical embeddings. Table 3 shows that the former yields non-negligibly lower average distortion on the "cat-cortex" dissimilarity dataset (Scannell, Blakemore, and Young 1995) and that the two are on-par on the "road-minnesota" graph (displayed in Figure 3c – notice its particular structure, characterized by cycles and low node degrees). As a general pattern, though, we find learning compact embeddings to be optimization-unfriendly (i.e., the results are quite sensitive to the optimization settings).

Dim	Manifold	F1@1 (road-minnesota)	Avg. Dist. (cat-cortex)
2	Euc	79.01	0.288
	Hyp	79.46	0.264
	Sphere	**82.19**	0.255
	$Gr(1,3)$	78.91	**0.234**
3	Euc	89.58	0.200
	Hyp	89.60	0.197
	Sphere	89.55	0.195
	$Gr(1,4)$	90.02	**0.168**
4	Euc	93.66	0.150
	Hyp	93.39	0.153
	Sphere	93.65	0.156
	$Gr(1,5)$	93.89	0.139
	$Gr(2,4)$	94.01	**0.129**

Table 3: Some results for "Gr vs. $\mathbb{S}$". We show the two metrics on two datasets that are the most discriminative. The full results, following the Table 2 format, are in Appendix H.

6 Conclusion & Future Work

We proposed to use the SPD and Grassmann manifolds for learning representations of graphs and showed that they are competitive against previously considered constant-curvature spaces on the graph reconstruction task, consistently and significantly outperforming them in some cases. Our results suggest that their geometry can accommodate certain graphs with better precision and less distortion than other embedding spaces. We thoroughly described their properties, emphasizing those that set them apart, and worked out the practically challenging aspects. Moreover, we advocate the Riemannian SNE objective for learning embeddings as a way to unify two different families of loss functions used in recent related works. It allows practitioners to explicitly tune the desired optimization goal by adjusting the temperature parameter. Finally, we defined the F1@k metric as a more general way of quantifying ranking fidelity.

Our work is related to some fundamental research questions. How does the curvature of a manifold influence the types of metrics that it can represent? How would a faithful embedding influence downstream tasks, such as node classification or link prediction? These are some of the questions we are excited about and plan to pursue in future work.

Acknowledgements

We would like to thank Andreas Bloch for suggesting the curvature quantification approach based on the sum of angles in geodesic triangles. We are grateful to Prof. Thomas Hofmann for making this collaboration possible. We thank the anonymous reviewers for helping us improve this work.

Gary Bécigneul is funded by the Max Planck ETH Center for Learning Systems.

Broader Impact

Our research deals with a deeply technical question: how can we leverage geometry to better represent graphs? It is also part of a nascent area of machine learning that tries to improve existing methods by bringing forward geometry tools and theories which have been known in mathematics for a (relatively) long time. That being said, should it later materialize into practically useful applications, its impact can be significant. That is due to the ubiquity of graphs as models of data and the possibility for our research to improve graph-based models. To give several examples, our research can have a broader impact in network analysis (in particular, social networks), working with biological data (e.g., proteins, molecules), or learning from knowledge graphs. The impact of our work can be beneficial, for instance, when having more faithful graph models can lead to improved recommender systems or drug discovery. However, we emphasize that current models might suffer from various other modes of failure and, thus, are not capable of replacing human expertise and intervention.

References

Bachmann, G.; Bécigneul, G.; and Ganea, O.-E. 2019. Constant Curvature Graph Convolutional Networks. *arXiv preprint arXiv:1911.05076* .

Becigneul, G.; and Ganea, O.-E. 2019. Riemannian Adaptive Optimization Methods. In *International Conference on Learning Representations*.

Belkin, M.; and Niyogi, P. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, 585–591.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8): 1798–1828.

Bengio, Y.; et al. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning* 2(1): 1–127.

Bhatia, R. 2009. *Positive definite matrices*, volume 24. Princeton University Press.

Bonnabel, S. 2013. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control* 58(9): 2217–2229.

Bronstein, A. M.; Bronstein, M. M.; and Kimmel, R. 2006. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences* 103(5): 1168–1172.

Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; and Vandergheynst, P. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34(4): 18–42.

Bruna, J.; Zaremba, W.; Szlam, A.; and Lecun, Y. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*.

Chossat, P.; and Faugeras, O. 2009. Hyperbolic planforms in relation to visual edges and textures perception. *PLoS computational biology* 5(12): e1000625.

De Sa, C.; Gu, A.; Ré, C.; and Sala, F. 2018. Representation tradeoffs for hyperbolic embeddings. *Proceedings of machine learning research* 80: 4460.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, 3844–3852.

Dolcetti, A.; and Pertici, D. 2018. Differential properties of spaces of symmetric real matrices. *arXiv preprint arXiv:1807.01113* .

Dong, Z.; Jia, S.; Zhang, C.; Pei, M.; and Wu, Y. 2017. Deep manifold learning of symmetric positive definite matrices with application to face recognition. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Donnat, C.; Zitnik, M.; Hallac, D.; and Leskovec, J. 2018. Learning Structural Node Embeddings via Diffusion Wavelets. In *International ACM Conference on Knowledge Discovery and Data Mining (KDD)*, volume 24.

Edelman, A.; Arias, T. A.; and Smith, S. T. 1998. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications* 20(2): 303–353.

Ganea, O.; Becigneul, G.; and Hofmann, T. 2018. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *International Conference on Machine Learning*, 1646–1655.

Ganea, O.; Bécigneul, G.; and Hofmann, T. 2018. Hyperbolic neural networks. In *Advances in neural information processing systems*, 5345–5355.

Giles, M. 2008. An extended collection of matrix derivative results for forward and reverse mode automatic differentiation. *Oxford University Computing Laboratory* .

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

Gu, A.; Sala, F.; Gunel, B.; and Ré, C. 2018. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 1024–1034.

Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* .

Hinton, G. E.; and Roweis, S. T. 2003. Stochastic neighbor embedding. In *Advances in neural information processing systems*, 857–864.

Hofmann, T.; and Buhmann, J. 1995. Multidimensional scaling and data clustering. In *Advances in neural information processing systems*, 459–466.

Huang, Z.; and Van Gool, L. 2017. A riemannian network for spd matrix learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Huang, Z.; Wu, J.; and Van Gool, L. 2018. Building deep networks on Grassmann manifolds. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Krioukov, D.; Papadopoulos, F.; Kitsak, M.; Vahdat, A.; and Boguná, M. 2010. Hyperbolic geometry of complex networks. *Physical Review E* 82(3): 036106.

Lang, S. 2012. *Fundamentals of differential geometry*, volume 191. Springer Science & Business Media.

Lee, J. M. 2006. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media.

Leskovec, J.; and Mcauley, J. J. 2012. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, 539–547.

Li, J.; Zhu, J.; and Zhang, B. 2016. Discriminative deep random walk for network classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1004–1013.

Mathieu, E.; Le Lan, C.; Maddison, C. J.; Tomioka, R.; and Teh, Y. W. 2019. Continuous Hierarchical Representations with Poincaré Variational Auto-Encoders. In *Advances in neural information processing systems*, 12544–12555.

Nickel, M.; and Kiela, D. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, 6338–6347.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. *NIPS 2017 Workshop Autodiff* .

Pennec, X.; Fillard, P.; and Ayache, N. 2006. A Riemannian framework for tensor computing. *International Journal of computer vision* 66(1): 41–66.

Roweis, S. T.; and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290(5500): 2323–2326.

Sarkar, R. 2011. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, 355–366. Springer.

Scannell, J. W.; Blakemore, C.; and Young, M. P. 1995. Analysis of connectivity in the cat cerebral cortex. *Journal of Neuroscience* 15(2): 1463–1483.

Skopek, O.; Ganea, O.-E.; and Bécigneul, G. 2020. Mixed-curvature Variational Autoencoders. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=S1g6xeSKDS.

Sra, S. 2012. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Advances in neural information processing systems*, 144–152.

Sra, S.; and Hosseini, R. 2015. Conic geometric optimization on the manifold of positive definite matrices. *SIAM Journal on Optimization* 25(1): 713–739.

Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500): 2319–2323.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML).

Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.; and Yang, S. 2017. Community preserving network embedding. In *Thirty-first AAAI conference on artificial intelligence*.

Wilson, R. C.; Hancock, E. R.; Pekalska, E.; and Duin, R. P. 2014. Spherical and hyperbolic embeddings of data. *IEEE transactions on pattern analysis and machine intelligence* 36(11): 2255–2269.

Wong, Y.-C. 1968. Sectional curvatures of Grassmann manifolds. *Proceedings of the National Academy of Sciences of the United States of America* 60(1): 75.

Xu, J.; and Durrett, G. 2018. Spherical Latent Spaces for Stable Variational Autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4503–4513.

Zhang, J.; Zhu, G.; Heath Jr, R. W.; and Huang, K. 2018. Grassmannian Learning: Embedding Geometry Awareness in Shallow and Deep Learning. *arXiv preprint arXiv:1808.02229* .

Zhang, M.; and Chen, Y. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, 5165–5175.