LEARNING GRAPH REPRESENTATIONS IN NORMED SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Theoretical results from discrete geometry suggest that normed spaces can abstractly embed finite metric spaces with surprisingly low theoretical bounds on distortion in low dimensions. In this paper, inspired by this theoretical insight, we propose normed spaces as a more flexible and computationally efficient alternative to several popular Riemannian manifolds for learning graph embeddings. Our normed space embeddings significantly outperform several popular manifolds on a large range of synthetic and real-world graph reconstruction benchmark datasets while requiring significantly fewer computational resources. We also empirically verify the superiority of normed space embeddings on growing families of graphs associated with negative, zero, and positive curvature, further reinforcing the flexibility of normed spaces in capturing diverse graph structures as graph sizes increase. Lastly, we demonstrate the utility of normed space embeddings on two applied graph embedding tasks, namely, link prediction and recommender systems. Our work highlights the potential of normed spaces for geometric graph representation learning, raises new research questions, and offers a valuable tool for experimental mathematics in the field of finite metric space embeddings.

1 INTRODUCTION

Graph representation learning aims to embed real-world graph data into ambient spaces while sufficiently preserving the geometric and statistical graph structures for subsequent downstream tasks and analysis. Graph data in many domains exhibit non-Euclidean features, making Euclidean embedding spaces an unfit choice. Motivated by the manifold hypothesis (see, e.g., Bengio et al. (2013)), recent research work has proposed embedding graphs into Riemannian manifolds (Chamberlain et al., 2017; Defferrard et al., 2020; Grattarola et al., 2020; Gu et al., 2019; Tifrea et al., 2019). These manifolds introduce inductive biases, such as symmetry and curvature, that can match the underlying graph properties, thereby enhancing the quality of the embeddings. For instance, Chamberlain et al. (2017) and Defferrard et al. (2020) proposed embedding graphs into hyperbolic and spherical spaces, with the choice determined by the graph structures. More recently, López et al. (López et al., 2021; López et al., 2021) proposed Riemannian symmetric spaces as a framework that unifies many Riemannian manifolds previously considered for representation learning. They also highlighted the Siegel and SPD symmetric spaces, whose geometries combine the sought-for inductive biases of many manifolds. However, operations in these non-Euclidean spaces are computationally demanding and technically challenging, making them impractical for embedding large graphs.

In this work, we propose normed spaces, particularly ℓ_1^d and ℓ_∞^d, as a more flexible, more computationally efficient, and less technically challenging alternative to several popular Riemannian manifolds for learning graph embeddings. Our proposal is motivated by theoretical results from discrete geometry, which suggest that normed spaces can abstractly embed finite metric spaces with surprisingly low theoretical bounds on distortion in low dimensions. This is evident in the work of Bourgain (1985); Johnson & Lindenstrauss (1984) and Johnson et al. (1987).

We evaluate the representational capacity of normed spaces on synthetic and real-world benchmark graph datasets through a graph reconstruction task. Our empirical results corroborate the theoretical motivation; as observed in our experiments, diverse classes of graphs with varying structures can be embedded in low-dimensional normed spaces with low average distortion. Second, we find that normed spaces consistently outperform Euclidean spaces, hyperbolic spaces, Cartesian products of these spaces, Siegel spaces, and spaces of SPD matrices across test setups. Further empirical analysis shows that the embedding capacity of normed spaces remains robust across varying graph curvatures and with increasing graph sizes. Moreover, the computational resource requirements for normed spaces grow much slower than other Riemannian manifold alternatives as the graph size increases. Lastly, we showcase the versatility of normed spaces in two applied graph embedding tasks, namely, link prediction and recommender systems, with the ℓ_1 normed space surpassing the baseline spaces.

As the field increasingly shifts towards technically challenging geometric methods, our work underscores the untapped potential of simpler geometric techniques. As demonstrated by our experiments, normed spaces set a compelling baseline for future work in geometric representation learning.

2 Related Work

Graph embeddings are mappings of discrete graphs into continuous spaces, commonly used as substitutes for the graphs in machine learning pipelines. There are numerous approaches for producing graph embeddings, and we highlight some representative examples: (1) Matrix factorization methods (Belkin & Niyogi, 2001; Cai et al., 2010; Tang & Liu, 2011) which decompose adjacency or Laplacian matrices into smaller matrices, providing robust mathematical vector representations of nodes; (2) Graph neural networks (GNNs) (Kipf & Welling, 2017; Veličković et al., 2018; Chami et al., 2019a) which use message-passing to aggregate node information, effectively capturing local and global statistical graph structures; (3) Autoencoder approaches (Kipf & Welling, 2016; Salha et al., 2019) which involve a two-step process of encoding and decoding to generate graph embeddings; (4) Random walk approaches (Perozzi et al., 2014; Grover & Leskovec, 2016; Kriege, 2022) which simulate random walks on the graph, capturing node proximity in the embedding space through co-occurrence probabilities; and (5) Geometric approaches (Gu et al., 2019; López et al., 2021) which leverage the geometric inductive bias of embedding spaces to align with the inherent graph structures, typically aiming to learn approximate isometric embeddings of the graphs in the embedding spaces. We note that these categories are not mutually exclusive. For instance, matrix factorization can be seen as a linear autoencoder approach, and geometric approaches can be combined with graph neural networks. Here we follow previous work (Gu et al., 2019; López et al., 2021; López et al., 2021; Giovanni et al., 2022a) and use a geometric approach to produce graph embeddings in normed spaces.

Recently, there has been a growing interest in geometric deep learning, especially in the use of Riemannian manifolds for graph embeddings. Those manifolds include hyperbolic spaces (Chamberlain et al., 2017; Ganea et al., 2018; Nickel & Kiela, 2018; López et al., 2019), spherical spaces (Meng et al., 2019; Defferrard et al., 2020), combinations thereof (Bachmann et al., 2020; Grattarola et al., 2020; Law & Stam, 2020), Cartesian products of spaces (Gu et al., 2019; Tifrea et al., 2019), Grassmannian manifolds (Huang et al., 2018), spaces of symmetric positive definite matrices (SPD) (Huang & Gool, 2017; Cruceru et al., 2020), and Siegel spaces (López et al., 2021). All these spaces are special cases of Riemannian symmetric spaces, also known as *homogeneous spaces*. Non-homogeneous spaces, such as Giovanni et al. (2022a), have been explored for embedding heterogeneous graphs. Other examples of mathematical spaces include Hilbert spaces (Sriperumbudur et al., 2010; Herath et al., 2017), Lie groups (Falorsi et al., 2018) (such as the torus (Ebisu & Ichise, 2018)), non-abelian groups (Yang et al., 2020) and pseudo-Riemannian manifolds of constant nonzero curvature (Law & Stam, 2020). These spaces introduce inductive biases that align with critical graph features. For instance, hyperbolic spaces, known for embedding infinite trees with arbitrarily low distortion (Sarkar, 2012), are particularly suitable for hierarchical data. Though the computations involved in working with these spaces are tractable, they incur non-trivial computational costs and often pose technical challenges. In contrast, normed spaces, which we focus on in this work, avoid these complications.

3 Theoretical Inspiration

In discrete geometry, abstract embeddings of finite metric spaces into normed spaces, which are characterized by low theoretical distortion bounds, have long been studied. Here we review some of the existence results that motivated our work. These results provide a rationale for using normed spaces to embed various graph types, much like hyperbolic spaces are often matched with hierarchical graph structures. While these results offer a strong motivation for our experiments, we emphasize that these theoretical insights do not immediately translate to or predict our empirical results.

It is a well-known fact that any n-pointed metric space can be isometrically embedded into a ℓ_∞^n. For many classes of graphs, the dimension can be substantially lowered: the complete graph K_n embeds

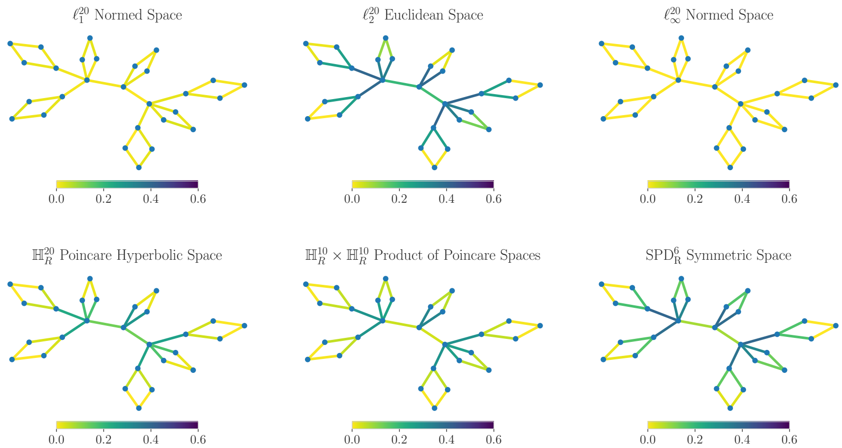

Figure 1: Embedding distortion across spaces on a small synthetic graph, with color indicating distortion levels (the absolute difference between graph edge and norm distances). The graph embeds well in the ℓ_1 and ℓ_∞ normed spaces but endures distortion in other spaces.

isometrically in $l_1^{\lceil \log_2(n) \rceil}$, every tree T with n vertices embeds isometrically in $\ell_\infty^{\mathcal{O}(\log n)}$, and every tree T with ℓ leaves embeds isometrically in $\ell_\infty^{\mathcal{O}(\log \ell)}$ (Linial et al., 1995).

Bourgain showed that similar dimension bounds can be obtained for finite metric spaces in general by relaxing the requirement that the embedding is isometric. A map $f : X \to Y$ between two metric spaces (X, d_X) and (Y, d_Y) is called a *D-embedding* for a real number $D \geq 1$ if there exists a number $r > 0$ such that for all $x_1, x_2 \in X$,

$$r \cdot d_X(x_1, x_2) \leq d_Y(f(x_1), f(x_2)) \leq D \cdot r \cdot d_X(x_1, x_2).$$

The infimum of the numbers D such that f is a D-embedding is called the *disortion* of f. Every n-point metric space (X, d) can be embedded in an $\mathcal{O}(\log n)$-dimensional Euclidean space with an $\mathcal{O}(\log n)$ distortion (Bourgain, 1985).

Johnson and Lindenstrauss obtained stronger control on the distortion at the cost of increasing the embedding dimension. Any set of n points in a Euclidean space can be mapped to $\mathbb{R}^t$ where $t = \mathcal{O}(\frac{\log n}{\epsilon^2})$ with distortion at most $1 + \epsilon$ in the distances. Such a mapping may be found in random polynomial time (Johnson & Lindenstrauss, 1984).

Similar embedding theorems were obtained by Linial et al. in other ℓ_p-spaces. In random polynomial-time (X, d) may be embedded in $\ell_p^{\mathcal{O}(\log n)}$ (for any $1 \leq p \leq 2$), with distortion $\mathcal{O}(\log n)$ (Linial et al., 1995) or into $\ell_p^{\mathcal{O}(\log^2 n)}$ (for any $p > 2$), with distortion $\mathcal{O}(\log n)$ (Linial et al., 1995).

When the class of graphs is restricted, stronger embedding theorems are known. Krauthgamer et al. obtain embeddings with bounded distortion for graphs when certain minors are excluded; we mention the special case of planar graphs. Let X be an n-point edge-weighted planar graph, equipped with the shortest path metric. Then X embeds into $\ell_\infty^{\mathcal{O}(\log n)}$ with $\mathcal{O}(1)$ distortion (Krauthgamer et al., 2004).

Furthermore, there are results on the limitations of embedding graphs into ℓ_p-spaces. For example, Linial et al. show that their embedding result for $1 \leq p \leq 2$ is sharp by considering expander graphs. Every embedding of an n-vertex constant-degree expander into an ℓ_p space, $2 \geq p \geq 1$, of any dimension, has distortion $\Omega(\log n)$. The metric space of such a graph cannot be embedded with constant distortion in any normed space of dimension $\mathcal{O}(\log n)$ (Linial et al., 1995).

These theoretical results illustrate the principle that large classes of finite metric spaces can in theory be abstractly embedded with low theoretical bounds on distortion in low dimensional normed spaces. Furthermore, the distortion and dimension can be substantially improved when the class of metric spaces is restricted. This leaves open many practical questions about the embeddability of real-world data into normed spaces and translating these theoretical results into predictions about the empirical results from experiments.

4 EXPERIMENTS

We evaluate the graph embedding capacity of normed spaces alongside other popular Riemannian manifolds via a graph reconstruction task on various synthetic and real-world graphs (§4.1). We analyze further (a) the space capacity and computational costs for varying graph sizes and curvatures; (b) space dimension; and in Appendix D, we extend our analysis to (c) expander graphs and (d) the asymmetry of the loss function. Separately, we evaluate normed spaces on two tasks: link prediction (§4.2) and recommender systems (§4.3). For link prediction, we investigate the impact of normed spaces on **four popular graph neural networks**.

4.1 BENCHMARK: GRAPH RECONSTRUCTION

Shortest Path Metric Embeddings. A *metric embedding* is a mapping $f : X \to Y$ between two metric spaces (X, d_X) and (Y, d_Y). Ideally, one would desire metric embeddings to be distance preserving. In practice, accepting some distortion can be necessary. In this case, the overall quality of an embedding can be evaluated by *fidelity measures* such as the *average distortion* D_{avg} and the *mean average precision* mAP. (Refer to Appendix C.1 for the definitions.) A special case of metric embedding is the *shortest path metric embedding*, also known as *low-distortion* or *approximate isometric* embedding, where X is the node set $\mathcal{V}$ of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $d_\mathcal{G}$ corresponds to the shortest path distance within $\mathcal{G}$. These embeddings represent or reconstruct the original graph G in the chosen embedding space Y, ideally preserving the desirable geometric features of the graph.

Learning Framework. To compute these embeddings, we optimize a distance-based loss function inspired by generalized MDS (Bronstein et al., 2006) and which was used earlier in, e.g., Gu et al. (2019); López et al. (2021); Giovanni et al. (2022a). Given graph distances $d_\mathcal{G}(u, v)$ between all pairs $u, v \in \mathcal{V}$ of nodes connected by a path in $\mathcal{G}$, which we denote $u \sim v$, the loss is defined as:

$$\mathcal{L}(f) = \sum_{u \sim v} \left| \left(\frac{d_Y(f(u), f(v))}{d_\mathcal{G}(u, v)} \right)^2 - 1 \right|, \tag{1}$$

where $d_Y(f(u), f(v))$ is the distance between the corresponding node embeddings in the target embedding space. In this context, the model parameters are a finite collection of points $f(u)$ in Y, each indexed by a specific node u of $\mathcal{G}$. These parameters, i.e., the coordinates of the points, are optimized by minimizing the loss function through gradient descent. This loss function treats the distortion of different path lengths uniformly during training. We note that the loss function exhibits an asymmetry: node pairs that are closer than they should contribute at most 1 to the loss, while those farther apart can have an unbounded contribution. Interestingly, we find that this asymmetry does not have any negative impact on the empirical results of node embeddings (see Appendix D). We provide more context for the loss function in Appendix B.

Motivation. In geometric machine learning, graph reconstruction tasks have achieved a "de facto" benchmark status for empirically quantifying the representational capacity of geometric spaces for preserving graph structures given through their local close neighborhood information, global all-node interactions, or an intermediate of both (Nickel & Kiela, 2017; 2018; Gu et al., 2019; Cruceru et al., 2020; López et al., 2021; López et al., 2021). This fidelity to structure is crucial for downstream tasks such as link prediction and recommender systems, where knowing the relationships between nodes or users is key. Other applications include embedding large taxonomies. For instance, Nickel & Kiela (2017) and Nickel & Kiela (2018) proposed embedding WordNet while maintaining its local graph structure (semantic relationships between words), and applied these embeddings to downstream NLP tasks. We note that though we employ a global loss function, the resulting normed space embeddings preserve both local and global structure notably well.

Experimental Setup. Following the work of Gu et al. (2019), we train graph embeddings by minimizing the previously mentioned distance-based loss function. We follow (López et al., 2021), and we do not apply any scaling to either the input graph distances or the distances calculated in the space, unlike earlier work (Gu et al., 2019; Cruceru et al., 2020). We report the average results across five runs in terms of (a) *average distortion* D_{avg} and (b) *mean average precision* (mAP). We provide the training details and the data statistics in Appendix C.1.

$(\|V\|, \|E\|)$	4D GRID (625, 2000)		TREE (364, 363)		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
$\mathbb{R}^{20}_{\ell_1}$	1.08 ±0.00	**100.00**	1.62±0.02	73.56	1.22±0.01	**100.00**	1.22±0.01	71.91	1.75±0.02	60.13
$\mathbb{R}^{20}_{\ell_2}$	11.24±0.00	**100.00**	3.92±0.04	42.30	9.78±0.00	96.03	3.86±0.02	34.21	4.28±0.04	27.50
$\mathbb{R}^{20}_{\ell_\infty}$	0.13±0.00	**100.00**	**0.15±0.01**	**100.00**	**0.58±0.01**	**100.00**	**0.09±0.01**	**100.00**	**0.23±0.02**	**99.39**
$\mathbb{H}^{20}_{R}$	25.23±0.05	63.74	0.54±0.02	**100.00**	20.59±0.11	75.67	14.56±0.27	44.14	14.62±0.13	30.28
SPD^{6}_{R}	11.24±0.00	**100.00**	1.79±0.02	55.92	8.83±0.01	98.49	1.56±0.02	62.31	1.83±0.00	72.17
$\mathcal{S}^{4}_{R}$	11.27±0.01	**100.00**	1.35±0.02	78.53	8.68±0.02	98.03	1.45±0.09	72.49	1.54±0.08	76.66
$\mathcal{S}^{4}_{F_\infty}$	5.92±0.06	99.61	1.23±0.28	99.56	3.31±0.06	99.95	10.88±0.19	63.52	10.48±0.21	72.53
$\mathcal{S}^{4}_{F_1}$	**0.01±0.00**	**100.00**	0.76±0.02	91.57	1.08±0.16	**100.00**	1.03±0.00	78.71	0.84±0.06	80.52
$\mathcal{B}^{4}_{R}$	11.28±0.01	**100.00**	1.27±0.05	74.77	8.74±0.09	98.12	2.88±0.32	72.55	2.76±0.11	96.29
$\mathcal{B}^{4}_{F_\infty}$	7.32±0.16	97.92	1.51±0.13	99.73	4.26±0.26	99.70	6.55±1.77	73.80	7.15±0.85	90.51
$\mathcal{B}^{4}_{F_1}$	0.39±0.02	**100.00**	0.77±0.02	94.64	1.28±0.16	**100.00**	1.09±0.03	76.55	0.99±0.01	81.82
$\mathbb{R}^{10}_{\ell_1} \times \mathbb{R}^{10}_{\ell_\infty}$	0.16±0.00	**100.00**	0.63±0.02	99.73	0.62±0.00	**100.00**	0.54±0.01	99.84	0.60±0.01	94.81
$\mathbb{R}^{10}_{\ell_1} \times \mathbb{H}^{10}_{R}$	0.55±0.00	**100.00**	1.13±0.01	99.73	0.62±0.01	**100.00**	1.76±0.02	50.74	1.65±0.01	89.47
$\mathbb{R}^{10}_{\ell_2} \times \mathbb{H}^{10}_{R}$	11.24±0.00	**100.00**	1.19±0.04	**100.00**	9.30±0.04	98.03	2.15±0.05	58.23	2.03±0.01	97.88
$\mathbb{R}^{10}_{\ell_\infty} \times \mathbb{H}^{10}_{R}$	0.14±0.00	**100.00**	0.22±0.02	96.96	1.91±0.01	99.13	0.15±0.01	99.96	0.57±0.01	90.34
$\mathbb{H}^{10}_{R} \times \mathbb{H}^{10}_{R}$	18.74±0.01	78.47	0.65±0.02	**100.00**	8.61±0.03	97.63	1.08±0.06	77.20	2.80±0.65	84.88

Table 1: Results on the five synthetic graphs. Lower D_{avg} is better. Higher mAP is better.

Baseline Comparison. We compare the performance of normed metric spaces with many other spaces for graph embedding. These spaces fall under three classes: (a) Normed spaces: $\mathbb{R}^{20}_{\ell_1}$, $\mathbb{R}^{20}_{\ell_2}$ and $\mathbb{R}^{20}_{\ell_\infty}$; (b) Riemannian symmetric spaces (Cruceru et al., 2020; López et al., 2021; López et al., 2021), incl. the space of SPD matrices: SPD^{6}_{R}, Siegel upper half spaces: $\mathcal{S}^{4}_{R}$, $\mathcal{S}^{4}_{F_1}$, $\mathcal{S}^{4}_{F_\infty}$, bounded symmetric spaces: $\mathcal{B}^{4}_{R}$, $\mathcal{B}^{4}_{F_1}$, $\mathcal{B}^{4}_{F_\infty}$, hyperbolic spaces (Nickel & Kiela, 2017): $\mathbb{H}^{20}_{R}$ (Poincaré model), and product spaces (Gu et al., 2019): $\mathbb{H}^{10}_{R} \times \mathbb{H}^{10}_{R}$; (c) Cartesian product spaces involving normed spaces: $\mathbb{R}^{10}_{\ell_1} \times \mathbb{R}^{10}_{\ell_\infty}$, $\mathbb{R}^{10}_{\ell_1} \times \mathbb{H}^{10}_{R}$, $\mathbb{R}^{10}_{\ell_2} \times \mathbb{H}^{10}_{R}$ and $\mathbb{R}^{10}_{\ell_\infty} \times \mathbb{H}^{10}_{R}$. The notation for all metrics follows a standardized format: the superscript represents the space dimension, and the subscript denotes the specific distance metric used (e.g., R for Riemannian and F for Finsler). Following López et al. (2021), we ensure uniformity across metric spaces by using the same number of free parameters, specifically a dimension of 20; and more importantly, we are concerned with the capacity of normed space at such low dimensions where non-Euclidean spaces have demonstrated success for embedding graphs (Chami et al., 2019b; Gu et al., 2019). We also investigate the capacities of spaces with growing dimensions, observing that other spaces necessitate much higher dimensions to match the capacity of the ℓ_∞ normed space (see Tab. 3). Note that $\mathcal{S}^n_{\cdot}$ and $\mathcal{B}^n_{\cdot}$ have $n(n+1)$ dimensions, and $\text{SPD}^n_{\cdot}$ has a dimension of $n(n+1)/2$. We elaborate on these metric spaces in Appendix A.

Synthetic Graphs. Following the work of López et al. (2021), we compare the representational capacity of various geometric spaces on several synthetic graphs, including grids, trees, and their Cartesian and rooted products. Further, we extend our analysis to three expander graphs, which can be considered theoretical worst-case scenarios for normed spaces, and thus are challenging setups for graph reconstruction. Tab. 1 reports the results on synthetic graphs.

Overall, the ℓ_∞ normed space largely outperforms all other metric spaces considered on the graph configurations we examine. Notably, it excels over manifolds typically paired with specific graph topologies. For instance, the ℓ_∞ space significantly outperforms hyperbolic spaces and surpasses Cartesian products of hyperbolic spaces on embedding tree graphs. Further, the ℓ_∞ space outperforms sophisticated symmetric spaces such as $\mathcal{S}^{4}_{F_1}$ and $\mathcal{B}^{4}_{F_1}$ on the graphs with mixed Euclidean and hyperbolic structures (TREE ⋄ GRIDS and GRIDS ⋄ TREE), although these symmetric spaces have compound geometries that combine Euclidean and hyperbolic subspaces. We also observe competitive performance from the ℓ_1 space, which outperforms the ℓ_2, hyperbolic, and symmetric spaces equipped with Riemannian and Finsler infinity metrics. Interestingly, combining ℓ_∞ and hyperbolic spaces using the Cartesian product does not bring added benefits and is less effective than using the ℓ_∞ space alone. Further, combining ℓ_1 and ℓ_∞ spaces yields intermediate performance between the individual ℓ_1 and ℓ_∞ spaces, due to the substantial performance gap between these two spaces. These findings underline the high capacity of the ℓ_1 and ℓ_∞ spaces, aligning with our theoretical motivations.

In Appendix D, though the expander graphs embed with substantial distortion, the ℓ_1 and ℓ_∞ spaces endure much lower distortion in D_{avg} than other spaces and also yield favorable mAP scores.

$(\|V\|, \|E\|)$	USCA312 (312, 48516)	BIO-DISEASOME (516, 1188)		CSPHD (1025, 1043)		EUROROAD (1039, 1305)		FACEBOOK (4039, 88234)	
	D_{avg}	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathbb{R}^{20}_{\ell_1}$	0.29±0.01	1.62±0.01	89.14	1.59±0.02	52.34	1.73±0.01	93.61	2.38±0.02	31.22
$\mathbb{R}^{20}_{\ell_2}$	**0.18±0.01**	3.83±0.01	76.31	4.04±0.01	47.37	4.50±0.00	87.70	3.16±0.01	32.21
$\mathbb{R}^{20}_{\ell_\infty}$	0.95±0.02	**0.53±0.01**	**98.24**	**0.42±0.01**	**99.28**	**1.06±0.01**	**99.48**	**0.71±0.02**	42.21
$\mathbb{H}^{20}_R$	2.39±0.02	6.83±0.08	91.26	22.42±0.23	60.24	43.56±0.44	54.25	3.72±0.00	44.85
SPD^6_R	0.21±0.02	2.54±0.00	82.66	2.92±0.11	57.88	19.54±0.99	92.38	2.92±0.05	33.73
$\mathcal{S}^4_R$	0.28±0.03	2.40±0.02	87.01	4.30±0.18	59.95	29.21±0.91	84.92	3.07±0.04	30.98
$\mathcal{S}^4_{F_\infty}$	0.57±0.08	2.78±0.49	93.95	27.27±1.00	59.45	46.82±1.02	72.03	1.90±0.11	**45.58**
$\mathcal{S}^4_{F_1}$	**0.18±0.02**	1.55±0.04	90.42	1.50±0.03	64.11	3.79±0.07	94.63	2.37±0.07	35.23
$\mathcal{B}^4_R$	0.24±0.07	2.69±0.10	89.11	28.65±3.39	62.66	53.45±2.65	48.75	3.58±0.10	30.35
$\mathcal{B}^4_{F_\infty}$	0.21±0.04	4.58±0.63	90.36	26.32±6.16	54.94	52.69±2.28	48.75	2.18±0.18	39.15
$\mathcal{B}^4_{F_1}$	**0.18±0.07**	1.54±0.02	90.41	2.96±0.91	67.58	21.98±0.62	91.63	5.05±0.03	39.87
$\mathbb{R}^{10}_{\ell_1} \times \mathbb{R}^{10}_{\ell_\infty}$	0.47±0.01	1.56±0.01	98.22	1.38±0.02	89.18	1.65±0.02	98.34	2.16±0.02	39.90
$\mathbb{R}^{10}_{\ell_1} \times \mathbb{H}^{10}_R$	0.72±0.01	1.99±0.01	93.78	1.83±0.02	78.10	2.26±0.02	96.19	2.77±0.02	33.79
$\mathbb{R}^{10}_{\ell_2} \times \mathbb{H}^{10}_R$	**0.18±0.00**	2.52±0.02	91.99	3.06±0.02	73.25	4.24±0.02	89.93	2.80±0.01	34.26
$\mathbb{R}^{10}_{\ell_\infty} \times \mathbb{H}^{10}_R$	0.42±0.02	1.42±0.02	96.51	1.16±0.01	76.91	1.77±0.01	97.38	1.41±0.02	35.03
$\mathbb{H}^{10}_R \times \mathbb{H}^{10}_R$	0.47±0.18	2.57±0.05	95.00	7.02±1.07	79.22	23.30±1.62	75.07	2.51±0.00	36.39

Table 2: Results on the five real-world graphs.

In sum, these results affirm that the ℓ_1 and ℓ_∞ spaces are well-suited for embedding graphs, showing robust performance when their geometry closely, or even poorly, aligns with the graph structures.

Real-World Graph Networks. We evaluate the representational capacity of metric spaces on five popular real-world graph networks. These include (a) USCA312 (Hahsler & Hornik, 2007) and EUROROAD (Šubelj & Bajec, 2011), representing North American city networks and European road systems respectively; (b) BIO-DISEASOME (Goh et al., 2007), a biological graph representing the relationships between human disorder and diseases and their genetic origins; (c) CSPHD (Nooy et al., 2011), a graph of Ph.D. advisor-advisee relationships in computer science and (d) FACEBOOK (McAuley & Leskovec, 2012), a dense social network from Facebook.

In Tab. 2, the ℓ_1 and ℓ_∞ spaces generally outperform all other metric spaces on real-world graphs, consistent with the synthetic graph results. However, for USCA312—a weighted graph of North American cities where edge lengths match actual spherical distances—the inherent spherical geometry limits effective embedding into the ℓ_1 and ℓ_∞ spaces at lower dimensions.

Large Graph Representational Capacity. We assess the capacity of the metric spaces for embedding graphs of increasing size, focusing on trees (negative curvature), grids (zero curvature), and fullerenes (positive curvature). See the illustrations of these graphs in Appendix E. For trees, we fix the valency at 3 and vary the tree height from 1 to 7; for 4D grids, we vary the grid dimension from 2 to 7; for fullerenes, we vary the number of carbon atoms from 20 to 240. We report the average results across three runs.

Fig. 2 (left) reports the results on trees. Within the range of computationally feasible graphs, we see that as graph size grows, the capacity of all metric spaces, barring the ℓ_∞ space and its product with hyperbolic space, improves significantly before plateauing. In hyperbolic spaces, which are not scale-invariant, embedding trees with high fidelity to all path lengths could require scaling (see, e.g., Sarkar (2012), Sala et al. (2018)). This can be seen in the high embedding distortion of small trees, specifically those with a height less than 4. Further empirical analysis demonstrates that the optimization goal localizes unavoidable distortion to some paths of short combinatorial lengths and their contribution to the average loss becomes smaller with increased size since there are relatively fewer of them. In contrast, the ℓ_∞ space consistently exhibits a high capacity, largely unaffected by graph size, and significantly outperforms the hyperbolic space within the observed range.

Fig. 2 (center) reports the results on 4D grids with zero curvature. We find that the metric spaces whose geometry aligns poorly with grid structures, such as the ℓ_2 space, the hyperbolic space and their products, exhibit weak representational capacity. In contrast, the ℓ_1 and ℓ_∞ spaces preserve grid structures consistently well as the graph size increases. Fig. 2 (right) reports the results on fullerenes with positive curvature. Given that none of the spaces considered feature a positively curved geometry, they are generally ill-suited for embedding fullerenes. However, we see that the ℓ_1 and ℓ_∞ spaces and the product spaces accommodating either of these two spaces consistently outperform others even as the number of carbon atoms increases.

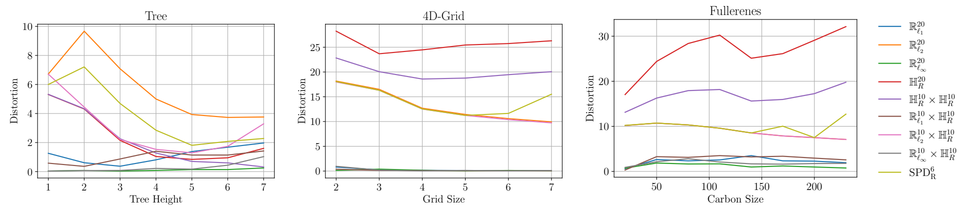

Figure 2: Metric space capacities with growing graph size and unnoticeable distortion variance.

Overall, these results show that the ℓ_1 and ℓ_∞ spaces consistently surpass other metric spaces in terms of representation capacity. They exhibit small performance fluctuation across various curvatures and maintain robust performance within graph configurations and size ranges we consider.

Training Efficiency. Fig. 3 compares the training time for different metric spaces on trees with growing size. The training times grow as $C(\text{space}) \times \mathcal{O}(\text{number of paths})$, where $C(\text{space})$ is a constant that depends on the embedding space. Among these spaces, SPD demands the highest amount of training efforts, even when dealing with small grids and trees. For other spaces, the training time differences become more noticeable with increasing graph size. The largest difference appears at a tree height of 6: The ℓ_1, ℓ_2, and ℓ_∞ normed spaces exhibit the highest efficiency, outperforming product, hyperbolic and SPD spaces in training time.

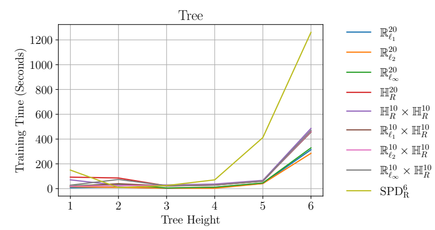

Figure 3: Training time vs. graph size.

These results are expected given than transcendental functions and eigendecompositions are computationally costly operations. Overall, normed spaces show high scalability with increasing graph size. Their training time grows much slower than product spaces and Riemannian alternatives. Similar trends are observed on grids (see Appendix D).

Space Dimension. Tab. 3 compares the results of different spaces across dimensions on the BIO-DISEASOME dataset. The surveyed theoretical results suggest that in sufficiently high dimensions, space capacities appear to approach theoretical limits, leading to the possibility that the performance of different spaces can become similar. However, we find that other spaces necessitate very high dimensions to match the capacity of the ℓ_∞ normed space. For instance, even after tripling space dimension, $\mathbb{H}_R^{66}$ and SPD_R^k still perform much worse than $\mathbb{R}_\infty^{20}$. $\mathbb{R}_{\ell_1}^{66}$ rivals $\mathbb{R}_\infty^{20}$ only in mAP. $\mathbb{R}_{\ell_1}^{33} \times \mathbb{R}_{\ell_\infty}^{33}$ surpasses $\mathbb{R}_\infty^{20}$ in mAP but lags behind in D_{avg}. López et al. (2021) similarly evaluated Euclidean, hyperbolic and spherical spaces, their products, and Siegel space at $n = 306$. Their best results on BIO-DISEASOME were $D_{avg} = 0.73$ and mAP $= 99.09$ for $\mathcal{S}_{F_1}^{17}$. In contrast, ℓ_∞^{20} and $\ell_1^{18} \times \ell_\infty^{18}$ achieved $D_{\text{avg}} = 0.5 \pm 0.01$ and mAP $= 99.4$, respectively. These results show that normed spaces efficiently yield low-distortion embeddings at much lower dimensions than other spaces.

4.2 APPLICATION 1: LINK PREDICTION

Experimental Setup. We comparably evaluate the impact of normed spaces on four popular architectures of graph neural networks (GNNs), namely GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), SGC (Wu et al., 2019) and GIN (Xu et al., 2019). Following (Kipf & Welling, 2016; Chami et al., 2019a), we evaluate GNNs in the link prediction task on two citation network datasets: Cora and Citeseer (Sen et al., 2008). This task aims to predict the presence of edges (links) between nodes that are not seen during training. We split each dataset into train, development and test sets corresponding to 70%, 10%, 20% of citation links that we sample at random. We report the average performance in AUC across five runs. We provide the training details in Appendix C.2.

Results. Tab. 4 reports the results of GNNs across different spaces for link prediction. While previous works showed the superiority of hyperbolic over Euclidean space for GNNs at lower

	$n=20$		$n=36$		$n=66$	
	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathbb{R}^n_{\ell_1}$	1.6±0.01	89.1	1.6±0.01	94.3	1.7±0.01	98.1
$\mathbb{R}^n_{\ell_2}$	3.8±0.01	76.3	3.8±0.01	85.9	3.9±0.01	86.2
$\mathbb{R}^n_{\ell_\infty}$	**0.5±0.01**	**98.2**	**0.5±0.01**	98.3	**0.6±0.01**	99.2
$\mathbb{H}^n_R$	6.8±0.08	91.2	5.8±0.06	93.6	5.9±0.05	93.2
SPD^k_R	2.5±0.00	82.6	2.4±0.02	87.8	2.3±0.02	90.5
$\mathbb{R}^{\frac{n}{2}}_{\ell_1} \times \mathbb{R}^{\frac{n}{2}}_{\ell_\infty}$	1.5±0.01	**98.2**	1.2±0.01	**99.4**	1.4±0.01	**99.8**
$\mathbb{R}^{\frac{n}{2}}_{\ell_1} \times \mathbb{H}^{\frac{n}{2}}_R$	1.9±0.01	93.7	1.8±0.01	95.8	1.7±0.01	98.4
$\mathbb{R}^{\frac{n}{2}}_{\ell_2} \times \mathbb{H}^{\frac{n}{2}}_R$	2.5±0.02	91.9	2.6±0.02	92.3	2.5±0.01	94.6
$\mathbb{R}^{\frac{n}{2}}_{\ell_\infty} \times \mathbb{H}^{\frac{n}{2}}_R$	1.4±0.02	96.5	1.1±0.02	98.6	0.8±0.01	98.8
$\mathbb{H}^{\frac{n}{2}}_R \times \mathbb{H}^{\frac{n}{2}}_R$	2.5±0.05	95.0	2.5±0.04	97.4	2.6±0.05	97.6

Table 3: Results on BIO-DISEASOME. When n takes 20, 36, 66, k in SPD^k_R takes 6, 8, 11.

dimensions on these datasets (Chami et al., 2019a; Zhao et al., 2023), our findings indicate the opposite (see the results from $\mathbb{H}^n_R$ and $\mathbb{R}^n_{\ell_2}$). This is attributed to the vanishing impact of hyperbolic space when operating GNNs in a larger dimensional space (with up to 128 dimension to achieve optimal performance on development sets). The ℓ_1 normed space consistently outperforms other spaces (including the ℓ_∞ and product spaces), demonstrating its superiority for link prediction.

	CORA				CITESEER			
	GCN	GAT	SGC	GIN	GCN	GAT	SGC	GIN
$\mathbb{R}^n_{l_1}$	**93.4±0.3**	**92.8±0.4**	**93.7±0.5**	**91.6±0.5**	**93.1±0.3**	**93.1±0.4**	93.8±0.4	**92.4±0.3**
$\mathbb{R}^n_{l_2}$	92.1±0.5	91.7±0.5	91.1±0.3	90.2±0.5	91.4±0.5	91.1±0.4	93.8±0.4	92.0±0.3
$\mathbb{R}^n_{l_\infty}$	89.5±0.4	88.2±0.5	88.8±0.3	88.4±0.5	90.3±0.4	89.5±0.5	91.7±0.3	90.5±0.3
$\mathbb{H}^n_R$	86.1±0.5	92.1±0.6	89.9±0.4	87.7±0.3	92.5±0.2	91.1±0.3	91.0±0.3	91.7±0.4
$\mathbb{R}^{\frac{n}{2}}_{\ell_1} \times \mathbb{R}^{\frac{n}{2}}_{\ell_\infty}$	93.0±0.5	92.3±0.3	93.5±0.5	90.9±0.4	92.9±0.5	92.8±0.6	**94.6±0.5**	**92.4±0.4**
$\mathbb{R}^{\frac{n}{2}}_{\ell_1} \times \mathbb{H}^{\frac{n}{2}}_R$	89.5±0.5	90.7±0.4	88.7±0.4	89.0±0.6	91.5±0.4	90.3±0.3	90.6±0.4	90.3±0.5
$\mathbb{R}^{\frac{n}{2}}_{\ell_2} \times \mathbb{H}^{\frac{n}{2}}_R$	90.2±0.4	90.6±0.5	88.3±0.6	87.8±0.4	90.8±0.5	91.2±0.3	91.0±0.5	90.4±0.3
$\mathbb{R}^{\frac{n}{2}}_{\ell_\infty} \times \mathbb{H}^{\frac{n}{2}}_R$	89.7±0.5	90.7±0.3	89.2±0.3	87.7±0.4	90.5±0.3	90.2±0.4	90.8±0.4	90.4±0.3
$\mathbb{H}^{\frac{n}{2}}_R \times \mathbb{H}^{\frac{n}{2}}_R$	85.2±0.3	88.0±0.4	88.7±0.4	87.6±0.3	90.4±0.3	87.2±0.5	89.2±0.3	91.3±0.5

Table 4: Results of GNNs in different spaces for link prediction, where n is a hyperparameter of space dimension that we tune on the development sets. Results are reported in AUC.

4.3 APPLICATION 2: RECOMMENDER SYSTEMS

Experimental Setup. Following López et al. (2021), we conduct a comparative examination of the impact of the choice of metric spaces on a recommendation task. This task can be seen as a binary classification problem on a bipartite graph, in which users and items are treated as two distinct subsets of nodes, and recommendation systems are tasked with predicting the interactions between user-item pairs. We adopt the approach of prior research (Vinh Tran et al., 2020; López et al., 2021) and base recommendation systems on graph embeddings in metric spaces. Our experiments include three popular datasets: (a) ML-100K (Harper & Konstan, 2015) from MovieLens for movie recommendation; (b) LAST.FM (Cantador et al., 2011) for music recommendation, and (c) MEETUP (Pham et al., 2015) from Meetup.com in NYC for event recommendation. We use the train/dev/test sets of these datasets from the work of López et al. (2021), and report the average results across five runs in terms of two evaluation metrics: hit ratio (HR) and normalized discounted cumulative gain (nDG), both at 10. We provide the training details and the statistics of the graphs in Appendix C.3.

Results. Tab. 5 reports the results on three bipartite graphs. We find that the performance gaps between metric spaces are small on ML-100K. Therefore, the choice of metric spaces does not influence much the performance on this graph. In contrast, the gaps are quite noticeable on the other two graphs. For instance, we see that the ℓ_1 space largely outperforms all the other spaces, particularly on LAST.FM. This showcases the importance of choosing a suitable metric space for downstream

	ML-100K		LASTFM		MEETUP	
	HR@10	nDG	HR@10	nDG	HR@10	nDG
$\mathbb{R}^{20}_{\ell_1}$	54.5±1.2	28.2	**69.3±0.4**	48.9	**82.1±0.4**	**63.3**
$\mathbb{R}^{20}_{\ell_2}$	54.6±1.0	28.7	55.4±0.3	24.6	79.8±0.2	59.5
$\mathbb{R}^{20}_{\ell_\infty}$	50.1±1.1	25.5	54.9±0.5	31.7	70.2±0.2	45.3
$\mathbb{H}^{20}_{R}$	53.4±1.0	28.2	54.8±0.5	24.9	79.1±0.5	58.8
SPD^{6}_{R}	53.3±1.4	28.0	55.4±0.2	25.3	78.5±0.5	58.6
$\mathcal{S}^{4}_{F_1}$	**55.6±1.3**	**29.4**	61.1±1.2	38.0	80.4±0.5	61.1
$\mathbb{R}^{10}_{\ell_1} \times \mathbb{R}^{10}_{\ell_\infty}$	52.0±1.1	27.1	68.2±0.4	47.3	79.6±0.3	60.1
$\mathbb{R}^{10}_{\ell_1} \times \mathbb{H}^{10}_{R}$	53.1±1.2	27.6	69.2±0.5	**49.9**	80.6±0.3	61.2
$\mathbb{R}^{10}_{\ell_2} \times \mathbb{H}^{10}_{R}$	53.1±1.3	27.9	45.5±0.4	18.9	79.3±0.2	58.9
$\mathbb{R}^{10}_{\ell_\infty} \times \mathbb{H}^{10}_{R}$	54.9±1.2	28.4	66.2±0.5	48.2	77.8±0.4	57.2
$\mathbb{H}^{10}_{R} \times \mathbb{H}^{10}_{R}$	54.8±0.9	29.1	55.0±0.6	24.6	79.5±0.2	59.2

Table 5: Results on the three recommendation bipartite graphs. Higher HR@10 and nDG are better.

tasks. It is noteworthy that the ℓ_1 norm outperforms the ℓ_∞ norm on the recommender systems task, while the opposite is true for the graph reconstruction task. This raises intriguing questions about how normed space embeddings leverage the geometries of the underlying normed spaces.

5 Conclusion

Classical discrete geometry results suggest that normed spaces can abstractly embed a wide range of finite metric spaces, including graphs, with surprisingly low theoretical bounds on distortion. Motivated by these theoretical insights, we propose normed spaces as a valuable complement to popular manifolds for graph representation learning. Our empirical findings show that normed spaces consistently outperform other manifolds across several real-world and synthetic graph reconstruction benchmark datasets. Notably, normed spaces demonstrate an enhanced capacity to embed graphs of varying curvatures, an increasingly evident advantage as graph sizes get bigger. We further illustrate the practical utility of normed spaces on two applied graph embedding tasks, namely link prediction and recommender systems, underscoring their potential for applications. Moreover, while delivering superior performance, normed spaces require significantly fewer computational resources and pose fewer technical challenges than competing solutions, further enhancing their appeal. Our work not only emphasizes the importance of normed spaces for graph representation learning but also naturally raises several questions and motivates further research directions:

Modern and Classical AI/ML Applications. The potential of our normed space embeddings can be tested across a wide range of AI applications. In many machine learning applications, normed spaces provide a promising alternative to existing Riemannian manifolds, such as hyperbolic spaces (Nickel & Kiela, 2017; 2018; Chami et al., 2020a;b) and other symmetric spaces, as embedding spaces. Classical non-differentiable discrete methods for embedding graphs into normed spaces have found applications in various areas (Livingston & Stout, 1988; Linial et al., 1995; Deza & Shtogrin, 2000; Mohammed, 2005). Our work demonstrates the efficient computation of graph embeddings into normed spaces using a modern differentiable programming paradigm. Integrating normed spaces into deep learning frameworks holds the potential to advance graph representation learning and its applications, bridging modern and classical AI research.

Discrete Geometry. Further analysis is needed to describe how normed space embeddings leverage the geometry of normed spaces. It is also important to investigate which emergent geometric properties of the embeddings can be used for analyzing graph structures, such as hierarchies. Lastly, we anticipate our work will provide a valuable experimental mathematics tool.

Limitations. Our work and others in the geometric machine literature, such as (Nickel & Kiela, 2017; 2018; Chami et al., 2019a; Cruceru et al., 2020; López et al., 2021; Giovanni et al., 2022a), lack theoretical guarantees. It is crucial to connect the theoretical bounds on distortion for abstract embeddings and the empirical results, especially for real-world graphs. It would also be valuable to analyze more growing families of graphs, such as expanders and mixed-curvature graphs. Furthermore, embedding larger real-world networks would provide insights into scalability in practical settings. Lastly, future work should expand this study to dynamic graphs with evolving structures.

REFERENCES

G. Bachmann, G. Bécigneul, and O.-E. Ganea. Constant curvature graph convolutional networks. In *37th International Conference on Machine Learning (ICML)*, 2020.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *7th International Conference on Learning Representations, ICLR*, New Orleans, LA, USA, May 2019. URL https://openreview.net/forum?id=r1eiqi09K7.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani (eds.), *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pp. 585–591. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/hash/f106b7f99d2cb30c3db1c3cc0fde9ccb-Abstract.html.

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

Béla Bollobás and Béla Bollobás. *Random graphs*. Springer, 1998.

Silvère Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58, 11 2011. doi: 10.1109/TAC.2013.2254619.

J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel J. Math.*, 52 (1-2):46–52, 1985. ISSN 0021-2172. doi: 10.1007/BF02776078. URL https://doi.org/10.1007/BF02776078.

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

Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys 2011, New York, NY, USA, 2011. ACM.

Ben Chamberlain, Marc Deisenroth, and James Clough. Neural embeddings of graphs in hyperbolic space. In *Proceedings of the 13th International Workshop on Mining and Learning with Graphs (MLG)*, 2017.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems 32*, pp. 4869–4880. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper/2019/file/0415740eaa4d9decbc8da001d3fd805f-Paper.pdf.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4869–4880, 2019b.

Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/ac10ec1ace51b2d973cd87973a98d3ab-Abstract.html.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6901–6914, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.617. URL https://www.aclweb.org/anthology/2020.acl-main.617.

C. Cruceru, G. Bécigneul, and O.-E. Ganea. Computationally tractable Riemannian manifolds for graph embeddings. In *37th International Conference on Machine Learning (ICML)*, 2020.

Michaël Defferrard, Martino Milani, Frédérick Gusset, and Nathanaël Perraudin. DeepSphere: A graph-based spherical CNN. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1e3OlStPB.

M Deza and Mikhail Ivanovich Shtogrin. Embeddings of chemical graphs in hypercubes. *Mathematical Notes*, 68:295–305, 2000.

Takuma Ebisu and Ryutaro Ichise. Toruse: Knowledge graph embedding on a Lie group. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 1819–1826. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16227.

Luca Falorsi, Pim de Haan, Tim R. Davidson, Nicola De Cao, Maurice Weiler, Patrick Forré, and Taco S. Cohen. Explorations in homeomorphic variational auto-encoding, 2018.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1646–1655, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/ganea18a.html.

Francesco Di Giovanni, Giulia Luise, and Michael M. Bronstein. Heterogeneous manifolds for curvature-aware graph embedding. *CoRR*, abs/2202.01185, 2022a. URL https://arxiv.org/abs/2202.01185.

Francesco Di Giovanni, Giulia Luise, and Michael M. Bronstein. Heterogeneous manifolds for curvature-aware graph embedding. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022b. URL https://openreview.net/forum?id=rtUxsN-kaxc.

Kwang-Il Goh, Michael E Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21): 8685–8690, 2007.

Daniele Grattarola, Daniele Zambon, Lorenzo Livi, and Cesare Alippi. Change detection in graph streams by learning graph embeddings on constant-curvature manifolds. *IEEE Trans. Neural Networks Learn. Syst.*, 31(6):1856–1869, 2020. doi: 10.1109/TNNLS.2019.2927301. URL https://doi.org/10.1109/TNNLS.2019.2927301.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (eds.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 855–864. ACM, 2016. doi: 10.1145/2939672.2939754. URL https://doi.org/10.1145/2939672.2939754.

Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJxeWnCcF7.

Michael Hahsler and Kurt Hornik. Tsp-infrastructure for the traveling salesperson problem. *Journal of Statistical Software*, 23(2):1–21, 2007.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

Sigurdur Helgason. *Differential geometry, Lie groups, and symmetric spaces*. Academic Press New York, 1978. ISBN 0123384605.

Samitha Herath, Mehrtash Tafazzoli Harandi, and Fatih Porikli. Learning an invariant Hilbert space for domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 3956–3965. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.421. URL https://doi.org/10.1109/CVPR.2017.421.

Zhiwu Huang and Luc Van Gool. A Riemannian network for SPD matrix learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 2036–2042. AAAI Press, 2017.

Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on Grassmann manifolds. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3279–3286. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16846.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemp. Math.*, pp. 189–206. Amer. Math. Soc., Providence, RI, 1984. doi: 10.1090/conm/026/737400. URL https://doi.org/10.1090/conm/026/737400.

William B. Johnson, Joram Lindenstrauss, and Gideon Schechtman. On Lipschitz embedding of finite metric spaces in low-dimensional normed spaces. In *Geometrical aspects of functional analysis (1985/86)*, volume 1267 of *Lecture Notes in Math.*, pp. 177–184. Springer, Berlin, 1987. doi: 10.1007/BFb0078145. URL https://doi.org/10.1007/BFb0078145.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016. URL http://arxiv.org/abs/1611.07308.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Max Kochurov, Rasul Karimov, and Sergei Kozlukov. Geoopt: Riemannian optimization in PyTorch. *ArXiv*, abs/2005.02819, 2020.

Robert Krauthgamer, James R Lee, Manor Mendel, and Assaf Naor. Measured descent: A new embedding method for finite metrics. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 434–443. IEEE, 2004.

Nils M. Kriege. Weisfeiler and leman go walking: Random walk kernels revisited. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 20119–20132. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/7eed2822411dc37b3768ae04561caafa-Paper-Conference.pdf.

Marc T. Law and Jos Stam. Ultrahyperbolic representation learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. ISSN 0209-9683. doi: 10.1007/BF01200757. URL https://doi.org/10.1007/BF01200757.

Marilynn Livingston and Quentin F Stout. Embeddings in hypercubes. *Mathematical and Computer Modelling*, 11:222–227, 1988.

Federico López, Benjamin Heinzerling, and Michael Strube. Fine-grained entity typing in hyperbolic space. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 169–180, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4319. URL https://www.aclweb.org/anthology/W19-4319.

Federico López, Beatrice Pozzetti, Steve Trettel, Michael Strube, and Anna Wienhard. Symmetric spaces for graph embeddings: A finsler-riemannian approach. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7090–7101. PMLR, 2021. URL http://proceedings.mlr.press/v139/lopez21a.html.

Federico López, Beatrice Pozzetti, Steve Trettel, Michael Strube, and Anna Wienhard. Vector-valued distance and gyrocalculus on the space of symmetric positive definite matrices. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021.

Alex Lubotzky. *Discrete groups, expanding graphs and invariant measures*, volume 125. Springer Science & Business Media, 1994.

Julian McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pp. 539–547, Red Hook, NY, USA, 2012. Curran Associates Inc.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. Spherical text embedding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 8208–8217. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/043ab21fc5a1607b381ac3896176dac6-Paper.pdf.

Qatawneh Mohammed. Embedding linear array network into the tree-hypercube network. *European Journal of Scientific Research*, 10(2):72–76, 2005.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6341–6350. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/59dfa2df42d9e3d41f5b02bfc32229dd-Paper.pdf.

Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3779–3788, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/nickel18a.html.

Wouter De Nooy, Andrej Mrvar, and Vladimir Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, USA, 2011. ISBN 0521174805.

OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences, 2023. Published electronically at http://oeis.org.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani (eds.), *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 701–710. ACM, 2014. doi: 10.1145/2623330.2623732. URL https://doi.org/10.1145/2623330.2623732.

T. N. Pham, X. Li, G. Cong, and Z. Zhang. A general graph-based model for recommendation in event-based social networks. In *2015 IEEE 31st International Conference on Data Engineering*, pp. 567–578, 2015. doi: 10.1109/ICDE.2015.7113315.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Re. Representation tradeoffs for hyperbolic embeddings. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4460–4469, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/sala18a.html.

Guillaume Salha, Romain Hennequin, Viet Anh Tran, and Michalis Vazirgiannis. A degeneracy framework for scalable graph autoencoders. In *28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In Marc van Kreveld and Bettina Speckmann (eds.), *Graph Drawing*, pp. 355–366, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-25878-7.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Carl Ludwig Siegel. Symplectic geometry. *American Journal of Mathematics*, 65(1):1–86, 1943. ISSN 00029327, 10806377. URL `http://www.jstor.org/stable/2371774`.

Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R.G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *J. Mach. Learn. Res.*, 11:1517–1561, August 2010. ISSN 1532-4435.

Lovro Šubelj and Marko Bajec. Robust network community detection using balanced propagation. *The European Physical Journal B*, 81:353–362, 2011.

Lei Tang and Huan Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23:447–478, 2011.

The Sage Developers. *SageMath, the Sage Mathematics Software System (Version x.y.z)*, 2023. `https://www.sagemath.org`.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincare GloVe: Hyperbolic word embeddings. In *7th International Conference on Learning Representations, ICLR*, New Orleans, LA, USA, May 2019. URL `https://openreview.net/forum?id=Ske5r3AqK7`.

Salil P Vadhan et al. *Pseudorandomness*, volume 7. Now Publishers, Inc., 2012.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJXMpikCZ`.

Lucas Vinh Tran, Yi Tay, Shuai Zhang, Gao Cong, and Xiaoli Li. HyperML: A boosting metric learning approach in hyperbolic space for recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pp. 609–617, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371850. URL `https://doi.org/10.1145/3336191.3371850`.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

Tong Yang, Long Sha, and Pengyu Hong. *NagE: Non-Abelian Group Embedding for Knowledge Graphs*, pp. 1735–1742. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450368599. URL `https://doi.org/10.1145/3340531.3411875`.

Wei Zhao, Federico Lopez, J Maxwell Riestenberg, Michael Strube, Diaaeldin Taha, and Steve Trettel. Modeling graphs beyond hyperbolic: Graph neural networks in symmetric positive definite matrices. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2023.

A EMBEDDING SPACES

Metric Spaces. Let X be a non-empty set. A *metric space* is an ordered pair (X, d), where $d : X \times X \to \mathbb{R}$ is a function, called the *metric* or *distance function*, that satisfies the following properties for all $x, y, z \in X$: (i) $d(x, y) \geq 0$, (ii) $d(x, y) = 0$ if and only if $x = y$, (iii) $d(x, y) = d(y, x)$, and (iv) $d(x, z) \leq d(x, y) + d(y, z)$. A map $f : X \to Y$ between two metric spaces (X, d_X) and (Y, d_Y) is an *isometric embedding* if it preserves distances, i.e., $d_Y(f(x_1), f(x_2)) = d_X(x_1, x_2), \forall$ $x_1, x_2 \in X$.

Riemannian Manifolds. Let M be a smooth manifold, $p \in M$ be a point, and T_pM be the tangent space at the point p. A Riemannian manifold (M, g) is a smooth manifold M equipped with a Riemannian metric g given by a smooth inner product $g_p \colon T_pM \times T_pM \to \mathbb{R}$ at each point $p \in M$. Euclidean space is the simplest example of a Riemannian manifold. Let V be any n-dimensional real vector space endowed with the Euclidean metric g given by $g(v, w) = \langle v, w \rangle$ for any $p \in V$ and any $v, w \in T_pV \cong V$.

Normed Spaces. A *normed space* is a vector space V over the real numbers $\mathbb{R}$ or complex numbers $\mathbb{C}$ equipped with a norm. A *norm* is a function $\|\cdot\| \colon V \to [0, +\infty)$ satisfying the following properties for all vectors $x, y \in V$ and scalars $\alpha \in \mathbb{F}$: (i) $\|x\| \geq 0$, with equality if and only if $x = 0$, (ii) $\|\alpha x\| = |\alpha|\|x\|$, and (iii) $\|x + y\| \leq \|x\| + \|y\|$. Normed spaces induce metric spaces via the *induced distance function*, defined as $d(x, y) = \|x - y\|$. The p-norms are among the most important examples of norms. For a real number $p \geq 1$, the p-norm of a vector $x \in \mathbb{R}^d$ is given by $\|x\|_p := (|x_1|^p + |x_2|^p + \cdots + |x_d|^p)^{\frac{1}{p}}$. The definition is extended for $p = \infty$ as $\|x\|_\infty := \max_{1 \leq i \leq d} |x_i|$. The space $\mathbb{R}^d$ equipped with p-norm is denoted as ℓ_p^d. Here we focus on the cases $p = 1, 2$, and ∞.

Hyperbolic Space. Hyperbolic space is a Riemannian manifold with a constant negative curvature. There are several models of hyperbolic space, such as the Poincaré ball model and Lorentz model. The models are essentially the same in a mathematical sense (they are pairwise isometric), but one model can have computational advantages over another.

Definition 1 (Poincaré Ball Model). *Let $\|\cdot\|$ be the Euclidean norm. Given a negative curvature c, the Poincaré ball model is a Riemannian manifold $(\mathcal{B}_c^n, g_{\mathbf{x}}^{\mathcal{B}})$, where $\mathcal{B}_c^n = \left\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|^2 < -1/c\right\}$ is an open ball with radius $1/\sqrt{|c|}$ and $g_{\mathbf{x}}^{\mathcal{B}} = (\lambda_{\mathbf{x}}^c)^2 \mathrm{Id}$, where $\lambda_{\mathbf{x}}^c = 2/(1 + c\|\mathbf{x}\|_2^2)$ and Id is the identity matrix.*

Product Manifold. Let $M_1, M_, \ldots, M_k$ be a sequence of smooth manifolds. The product manifold is given by the Cartesian product $M = M_1 \times M_2 \times \cdots \times M_k$. Each point $p \in M$ has the coordinates $p = (p_1, \ldots, p_k)$, with $p_i \in M_i$ for all i. Similarly, a tangent vector $v \in T_pM$ can be written as $(v_1, \ldots, v_k)$, with each $v_i \in T_{p_i}M_i$. If each M_i is equipped with a Riemannian metric g_i, then the product manifold M can be given the product metric where $g(v, w) = \sum_{i=1}^k g_i(v_i, w_i)$.

Riemannian Symmetric Spaces. Riemannian symmetric spaces are connected Riemannian manifolds such that the geodesic symmetry[1] at each point defines a global isometry of the space. For simply connected manifolds this condition is equivalent to having covariantly constant curvature tensor. A key consequence of the definition is that symmetric spaces are homogeneous manifolds. Intuitively, this means that the manifold "looks the same" at every point. Furthermore, simply connected symmetric spaces decompose into products of irreducible symmetric spaces and Euclidean space. Irreducible symmetric spaces can be described in terms of semisimple Lie groups. Basic examples of Riemannian symmetric spaces include Euclidean spaces, hyperbolic spaces and spheres. In the following we will describe two further special cases: Siegel space and the space of symmetric positive definite (SPD) matrices.

Siegel spaces, HypSPD_n, are matrix versions of the hyperbolic plane, accommodating many products of hyperbolic planes and the copies of SPD as submanifolds. These spaces support Finsler metrics that induce the ℓ_1 and the ℓ_∞ metric on the Euclidean subspaces. HypSPD_n has the two following

[1]For any point p in any Riemannian manifold, there exists a sufficently small $\epsilon > 0$ such that the map $S_p \colon B(p, \epsilon) \to B(p, \epsilon)$ defined by $S_p(c(t)) = c(-t)$ is well-defined for any unit-speed geodesic $c \colon (-\epsilon, \epsilon) \to B(p, \epsilon)$ with $c(0) = p$. Such a map S_p is called the *geodesic symmetry* at p.

models with $n(n+1)$ dimensions, both of which are open subsets of the space $\mathrm{Sym}(n, \mathbb{C})$ over $\mathbb{C}$. These two models generalize the Poincaré disk and the upper half plane model of the hyperbolic space.

Definition 2 (Bounded Symmetric Domain Model)**.** *The bounded symmetric domain model for HypSPD$_n$ generalizes the Poincaré disk. It is given by* $\mathcal{B}_n := \{Z \in \mathrm{Sym}(n, \mathbb{C}) | \; \mathrm{Id} - Z^*Z \gg 0\}$.

Definition 3 (Siegel Upper Half Space Model)**.** *The Siegel upper half space model for HypSPD$_n$ generalizes the upper half plane model of the hyperbolic plane by* $\mathcal{S}_n := \{Z = X + iY \in \mathrm{Sym}(n, \mathbb{C}) | \; Y \gg 0\}$.

There exists an isomorphism from $\mathcal{B}_n$ to $\mathcal{S}_n$ given by the Cayley transform, which is a matrix analogue of the familiar map from the Poincare disk to upper half space model of the hyperbolic plane:

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

We refer readers to Siegel (1943) and López et al. (2021) for an in-depth overview of Siegel spaces and their applications in graph embeddings.

Definition 4 (SPD Space)**.** SPD_n *is the space of positive definite real symmetric* $n \times n$ *matrices, given by* $\mathrm{SPD}(n, \mathbb{R}) := \{X \in \mathrm{Sym}(n, \mathbb{R}) | \; X \gg 0\}$. *It has the structure of a Riemannian manifold of non-positive curvature of* $n(n+1)/2$ *dimensions. The Riemannian metric on* SPD_n *is defined as follows: if* $U, V \in S_n$ *are tangent vectors based at* $P \in \mathrm{SPD}_n$*, their inner product is given by* $\langle U, V \rangle_P = \mathrm{Tr}(P^{-1}UP^{-1}V)$.

The tangent space to any point of SPD_n can be identified with the vector space S_n of all real symmetric $n \times n$ matrices. SPD_n is more flexible than Euclidean or hyperbolic geometries, or products thereof. In particular, it contains n-dimensional Euclidean subspaces, $(n-1)$-dimensional hyperbolic subspaces, products of $\lfloor \frac{n}{2} \rfloor$ hyperbolic planes, and many other interesting spaces as totally geodesic submanifolds, see the reference (Helgason, 1978) for an in-depth introduction.

B Graph Reconstruction Loss Function

The graph reconstruction task aims to empirically quantify the capacity of a space for embedding graph structure given through its node-to-node shortest paths. Recent work has generally employed local, global, or hybrid loss functions, focusing on close neighborhood information, all-node interactions, or an intermediate of both. Local loss functions emphasize preserving neighborhoods, exemplified by the loss function

$$\mathcal{L}(f) = -\sum_{(u,v)\in E} \log \frac{\exp\big(-d_Y(f(u), f(v))\big)}{\sum_{w\in\mathcal{N}(u)} \exp\big(-d_Y(f(u), f(w))\big)}.$$

from Nickel & Kiela (2017; 2018), where $\mathcal{N}(u) = \{w \mid (u, w) \notin E\} \cup \{v\}$ is the set of negative examples for u (including v). The resulting embeddings are typically favored by rank-based evaluation metrics such as mean average precision mAP. On the other hand, global functions emphasize preserving distances directly via loss functions motivated by generalized MDS (Bronstein et al. (2006)), exemplified by the loss function

$$\mathcal{L}(f) = \sum_{u\sim v} \left| \left(\frac{d_Y(f(u), f(v))}{d_{\mathcal{G}}(u, v)} \right)^2 - 1 \right|,$$

from Gu et al. (2019), and which we use in this work. The resulting embeddings are typically favored by average distortion D_{avg}. Lastly, hybrid loss functions, such as the Riemannian Stochastic Neighbor Embedding (RSNE) from Cruceru et al. (2020), aim to balance the emphasis on local and global, sometimes with a tunable parameter for controlling the optimization goal.

We note that though we employ a global loss function, the resulting normed space embeddings notably perform well on both D_{avg} and mAP.

C Experiments

Hardware and Code Release. All experiments were executed on an Intel(R) Xeon(R) CPU E5-2650 computer, equipped with 48 CPUs operating at 2.2 GHz and a single Tesla P40 GPU with a 24GB of memory running on CUDA 11.2. We release our code and datasets at `https://anonymous.4open.science/r/ICLR24-2E26`.

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 × 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 6: Characteristics of synthetic graphs.

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
MARGULIS	625	2,500	195,000
PALEY	101	5,050	5,050
CHORDAL	523	1,569	136,503

Table 7: Characteristics of real-world and expander graphs.

C.1 GRAPH RECONSTRUCTION

Implementation Details. In all setups, we use the RADAM optimizer (Bécigneul & Ganea, 2019), and run the same grid search to to train graph embeddings. The implementation of all baselines are taken from Geoopt (Kochurov et al., 2020) and López et al. (2021). We train for 3000 epochs, and stop training when the average distortion has not decreased for 200 epochs. We experiment with three hyperparameters: (a) learning rate $\in \{0.1, 0.01, 0.001\}$; (b) batch size $\in \{512, 1024, 2048, -1\}$ with -1 as the node count within a graph and (c) maximum gradient norm $\in \{10, 50, 250\}$. Table 6 and 7 report the stats of all the synthetic and real-world graphs.

Evaluation Metrics. We evaluate the quality of the learned embeddings using distortion and precision metrics. Consider a graph G, a target metric space Y, and a metric embedding $f : G \to Y$. The distortion of the embedding of a pair of nodes u, v is given by:

$$\text{distortion}(u, v) = \frac{|d_Y(f(u), f(v)) - d_G(u, v)|}{d_G(u, v)}.$$

We denote the average of distortion over all pairs of nodes by D_{avg}.

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. For the mean average precision (mAP) metric, consider $G = (V, E)$ as a graph and $\mathcal{N}_a$ as the neighborhood of the node $a \in V$. Let R_{a,b_i} be the smallest neighborhood of $f(a)$ in the space Y that contains $f(b_i)$, with $f : G \to \mathcal{P}$ as a metric embedding. Then, mAP can be defined as follows:

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

mAP quantifies how well the embedding approximates graph isomorphism, applicable only to unweighted graphs. mAP measures the average discrepancy between the neighborhood of each node $u \in V$ and the neighborhood of $f(u) \in Y$. It's important to note that an embedding with zero average distortion guarantees a perfect mean average precision score (i.e., 100.00), but the inverse is not always true: an embedding that effectively preserves the adjacency structure might not be an isometry.

C.2 LINK PREDICTION

Implementation Details. For each dataset, we use grid search to tune hyperparameters on the development set. Our hyperparameters include (a) dimension $\in \{32, 64, 128\}$ and (b) learning rate

$\in \{0.1, 0.01, 0.001\}$. We set batch size to the number of nodes present in each graph dataset. We train for 1000 epochs and stop training when the loss on the development set has not been decreased for 200 epochs. We report the average performance in AUC across five runs. Following Chami et al. (2019a), we use the Fermi-Dirac decoder to compute the likelihood of a link between node pairs, and generate negative sets by randomly selecting links from non-connected node pairs. All graph neural networks are trained by optimizing the cross-entropy loss function. We extend the implementation of Poincaré GCN (Chami et al., 2019a) to support the other three architectures, enabling them to operate in both hyperbolic space and product spaces. We reduce the learning rate by a factor of 5 if GNNs cannot improve the performance after 50 epochs for hyperbolic and product spaces.

C.3 RECOMMENDER SYSTEMS

Implementation Details. We follow a metric learning approach Vinh Tran et al. (2020), with the implementation of all baselines taken from López et al. (2021). We minimize the hinge loss function for ML-100K and LAST.FM, while minimizing the binary cross-entropy (BCE) function for MEETUP. We use the RSGD optimizer (Bonnabel, 2011) to tune graph node embeddings. In all setups, we run the same grid search to train recommender systems. We train for 500 epochs, reduce the learning rate by a factor of 5 if the model does not improve the performance after 50 epochs. We stop training when the loss on the dev set has not been decreased for 50 epochs. We use the burn-in strategy (Nickel & Kiela, 2017; Cruceru et al., 2020) that trains recommender systems with a 10 times smaller learning rate for the first 10 epochs. We experiment with three hyperparameters: (a) learning rate $\in \{0.1, 0.01, 0.001\}$; (b) batch size $\in \{512, 1024, 2048\}$ and (c) maximum gradient norm $\in \{5, 10, 50\}$. Table 8 reports the stats of all the bipartite graphs in the recommendation task.

Model. Given a set of entities $\mathcal{E}$ and a target metric space (X, d_X), we associate with each entity $e \in \mathcal{E}$ an embedding $f(e) \in X$ and bias terms $b_{e,\text{lhs}}, b_{e,\text{rhs}} \in \mathbb{R}$, where $f : \mathcal{E} \to X$ is a learnable embedding function. Given a pair of entities $e_1, e_2 \in \mathcal{E}$, the model computes a similarity score $\phi(e_1, e_2)$ as follows

$$\phi_{f,b,X}(e_1, e_2) := b_{e_1,\text{lhs}} + b_{e_2,\text{rhs}} - d_X^2(f(e_1), f(e_2)). \tag{2}$$

Subtracting the square distance ensures that the entities whose embeddings are closer in the metric space have a higher score, making it a suitable representation of similarity. The model we use is *shallow*: It learns a collection of points $f(e) \in M$ indexed by the entities $e \in \mathcal{E}$. In our setting, $\mathcal{E} = \mathcal{U} \cup \mathcal{V}$, where $\mathcal{U}$ is the space of users and $\mathcal{V}$ is the space of items.

Hinge Loss Function. Given a set $\mathcal{T} = \{(u, v)\}$ of observed user-item interactions, the hinge loss function is given by:

$$\mathcal{L} = \sum_{(u,v)\in\mathcal{T}} \sum_{(u,w)\notin\mathcal{T}} [m + \phi_{f,b,X}(u, v) - \phi_{f,b,X}(u, w)]_+, \tag{3}$$

where w is an item the user u has not interacted with, m is the hinge margin, and $[z]_+ = max(0, z)$. For each user u, we generate a negative set by randomly selecting 100 items that the user has not interacted with.

Binary Cross-Entropy Loss Function. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be a set of observed user-item interactions and a set of non-interactions, respectively. Consider $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ as the collection of all interactions and non-interactions. For each pair $(u, v) \in \mathcal{T}$, let $y_{u,v} \in \{0, 1\}$ denote the true label: If the pair belongs to $\mathcal{T}_1$, then $y_{u,v} = 1$, otherwise $y_{u,v} = 0$. The BCE loss function is given by:

$$\mathcal{L} = \sum_{(u,v)\in\mathcal{T}} -y_{u,v} \cdot \log(\sigma(\phi_{f,b,X}(u, v))) - (1 - y_{u,v}) \cdot \log(1 - \sigma(\phi_{f,b,X}(u, v))), \tag{4}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. For each user u, we generate a negative set by randomly selecting one item that the user has not interacted with.

D SUPPLEMENTARY GRAPH RECONSTRUCTION ANALYSIS

Expander Graphs. We run a graph reconstruction analysis on three expanded graphs, namely Margulis-Gabber-Galil, Paley, and Chordal-Cycle graphs (Bollobás & Bollobás, 1998; Lubotzky,

Dataset	Users	Items	Interactions	Density (%)
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 8: Recommender system dataset stats

$(\|V\|, \|E\|)$	MARGULIS $(625, 2500)$		PALEY $(101, 5050)$		CHORDAL $(523, 1569)$	
	D_{avg}	mAP	D_{avg}	mAP	D_{avg}	mAP
$\mathbb{R}^{20}_{\ell_1}$	**13.4±0.00**	**87.97**	22.7±0.00	65.84	10.7±0.01	**99.66**
$\mathbb{R}^{20}_{\ell_2}$	14.0±0.01	83.99	23.6±0.02	60.80	12.8±0.01	87.79
$\mathbb{R}^{20}_{\ell_\infty}$	14.2±0.01	82.73	**16.1±0.01**	**66.88**	**10.5±0.01**	98.39
$\mathbb{H}^{20}_{R}$	16.8±0.01	69.47	23.8±0.02	60.76	22.8±0.02	59.19
SPD^{6}_{R}	14.1±0.01	84.98	23.6±0.01	61.76	12.8±0.01	77.59
$\mathcal{S}^{4}_{F_1}$	24.2±0.02	2.24	26.6±0.01	51.94	38.1±0.02	1.40
$\mathcal{B}^{4}_{F_1}$	24.1±0.01	2.17	26.5±0.01	52.97	37.2±0.01	1.43
$\mathbb{R}^{10}_{\ell_1} \times \mathbb{R}^{10}_{\ell_\infty}$	13.8±0.00	87.25	20.4±0.01	60.09	10.6±0.01	99.47
$\mathbb{R}^{10}_{\ell_1} \times \mathbb{H}^{10}_{R}$	14.2±0.00	83.63	23.3±0.00	62.77	11.7±0.00	82.95
$\mathbb{R}^{10}_{\ell_2} \times \mathbb{H}^{10}_{R}$	14.4±0.00	79.12	23.7±0.01	60.72	12.8±0.00	81.93
$\mathbb{R}^{10}_{\ell_\infty} \times \mathbb{H}^{10}_{R}$	14.6±0.01	86.26	20.8±0.00	60.57	12.1±0.01	88.98
$\mathbb{H}^{10}_{R} \times \mathbb{H}^{10}_{R}$	15.4±0.01	75.77	23.7±0.01	60.33	17.2±0.00	58.25

Table 9: Results on the three expander graphs.

1994; Vadhan et al., 2012). Unlike grids and tree graphs, expander graphs present complex structures due to their high degree of sparsity and connectivity. Linial et al. (1995) considered expander graphs as the worst-case scenarios for their embedding theorems of normed spaces. Table 1 and 9 report the results on synthetic graphs.

As reported in Table 9, none of the metric spaces investigated align well with these intricate structures, leading to substantial distortion of graph structures across all spaces. Nonetheless, graph structures in the ℓ_1 and ℓ_∞ spaces endure much lower distortion in terms of D_{avg} than other spaces. Even with considerable distortion, the ℓ_1 and ℓ_∞ spaces yield favorable mAP scores, particularly on CHORDAL. This suggests that the CHORDAL graph, while not isometrically embedded, is nearly isomorphic in these spaces, indicative of high-quality embeddings.

Asymmetric Loss Function. Although our distance-based loss function is a popular choice for embeddings graphs with low distortion in low-dimensional space in the literature (Gu et al., 2019; López et al., 2021; Giovanni et al., 2022b), it exhibits asymmetry by penalizing nodes that are closer than they should in the embedding space by at most 1, whereas nodes farther apart than they should receive an unbounded penalty. Here we examine whether this asymmetry has negative impacts on the results of graph embeddings. Figure 4 (a) shows that distortions between nodes are mostly -1, meaning that nodes initially are very close in the embedding space, due to their random initialization

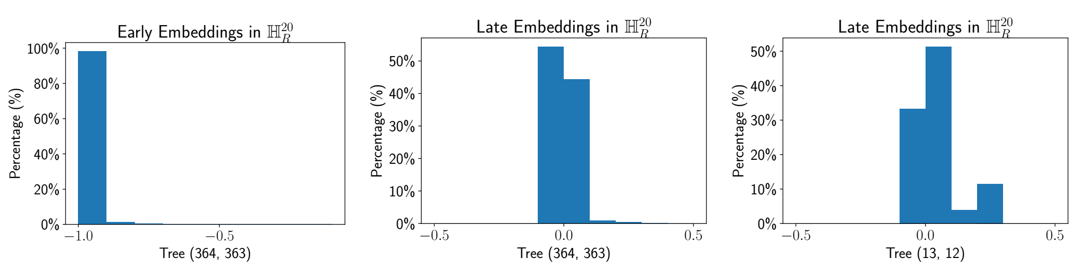

Figure 4: Histograms of distortions between nodes on small- and medium-sized trees, denoted as $\text{distortion}(a, b) = \frac{d_{\mathcal{P}}(f(a), f(b))}{d_G(a,b)} - 1$. Early embeddings: 5 epochs, Late embeddings: after training.

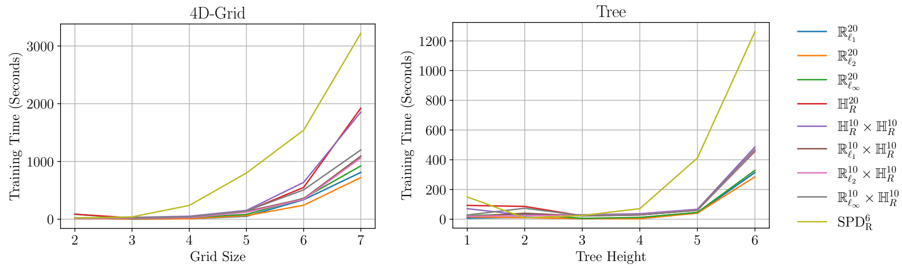

Figure 5: Training time scales up as the graph size increases.

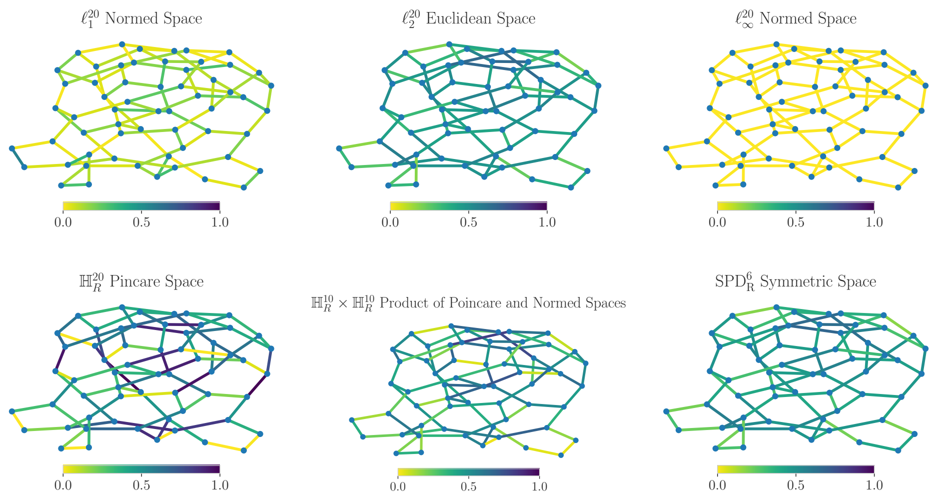

Figure 6: Embedding distortion shown in various spaces on a small expander-chordal graph. Color range indicates distortion levels.

within the small interval (-1e-3, 1e-3). Figure 4 (b) shows that distortions approach zero from both sides after training, implying that node embeddings farther apart than they should are penalized harshly during the early stage of training. Over time, node embeddings become either closer or apart slightly than they should. In this case, they are penalized similarly. By comparing Figure 4 (b) and (c), we see that the asymmetry has a bigger impact on the small tree, and then its impact appears to diminish as the tree size grows. This might be because the small tree has fewer nodes and edges, and therefore a small fluctuation in the embedding space can incur a considerable influence on the overall distortion of the tree. However, this issue becomes less pronounced when dealing with larger trees.

Training Efficiency. We provide additional results for comparing the training time of different spaces on grids in Figure 5.

Additional Results. For readability, we choose to embed a small expander graph into various spaces and compare embedding distortion in these spaces, as displayed in Figure 6. We find that both normed spaces perform much better than other spaces. Further, we see that the graph undergoes small distortion in the ℓ_1 space and unnoticeable distortion in the ℓ_∞ space when dealing with a small expander, although embedding expanders into normed spaces is a well-known challenge (Linial et al., 1995, Proposition 4.2).

E Trees, Grids, and Fullerenes

In our Large Graph Representational Capacity experiments (Section 4), we used trees, grids, and fullerenes as discretizations of manifolds with negative, zero, and positive curvatures, respectively.

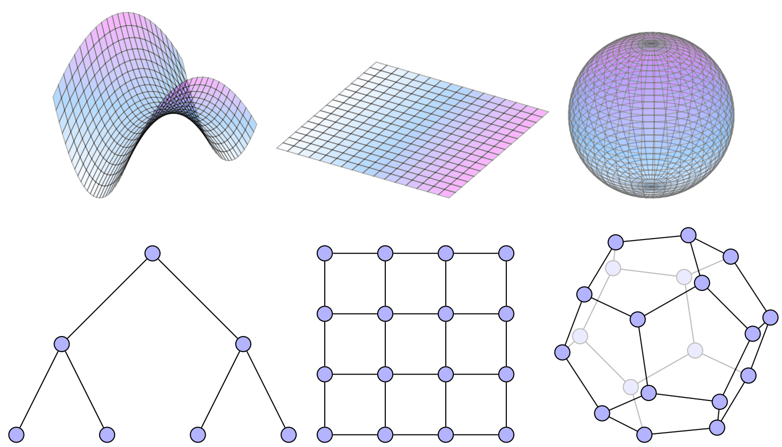

Figure 7: Top: surfaces of negative, zero, and positive curvature (from left to right). Bottom: graphs of negative, zero, and positive curvature (from left to right).

Refer to Figure 7 for a visual illustration of these discretizations. In chemistry, a *fullerene* is any molecule composed entirely of carbon in the form of a hollow spherical, ellipsoidal, or cylindrical mesh. In our experiments, we used combinatorial graphs representing spherical fullerenes. We generated the fullerene graphs using the `graphs.fullerenes()` function from SageMath (The Sage Developers, 2023). The number of possible fullerenes grows fast as a function in the number of nodes (OEIS Foundation Inc., 2023, A007894), and we used the first fullerene graph generated by `graphs.fullerenes()` for each node count. The graph data for the specific fullerenes used in our experiments can be found in our code repository, ensuring reproducibility and facilitating further analysis. Trees and grids are well-known and require no further description.