GRAPH-BASED NEAREST NEIGHBOR SEARCH IN HYPERBOLIC SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

The nearest neighbor search (NNS) problem is widely studied in Euclidean space, and graph-based algorithms are known to outperform other approaches for this task. However, hyperbolic geometry is found to be very useful for data representation in various domains, including graphs, words, and images. In this paper, we show that graph-based approaches are also well suited for hyperbolic geometry. From a theoretical perspective, we rigorously analyze the time and space complexity of graph-based NNS, assuming that an n-element dataset is uniformly distributed within a d-dimensional ball of radius R in the hyperbolic space of curvature -1. Under some conditions on R and d, we derive the time and space complexity of graph-based NNS and compare the obtained results with known guarantees for the Euclidean case. Interestingly, in the dense setting ($d \ll \log n$) and under some assumptions on the radius R, graph-based NNS has *lower* time complexity in the hyperbolic space. This agrees with our experiments: we consider datasets embedded in hyperbolic and Euclidean spaces and show that graph-based NNS can be more efficient in the hyperbolic space. We also demonstrate that graph-based methods outperform other existing baselines for hyperbolic NNS. Overall, our theoretical and empirical analysis suggests that graph-based NNS can be considered a default approach for similarity search in hyperbolic spaces.

1 INTRODUCTION

Given a dataset $\mathcal{D}$ of n elements and an arbitrary query $\mathbf{q}$, the goal of the nearest neighbor search (NNS) is to quickly retrieve one or several elements from $\mathcal{D}$ closest to $\mathbf{q}$. NNS is extensively used in a wide range of machine learning systems: for non-parametric classification and regression, language modeling, information retrieval, recommendations, and many others (Bishop, 2006; Chen et al., 2018; May & Ozerov, 2015; Shakhnarovich et al., 2006). In large-scale applications, searching for the nearest neighbors can be a bottleneck, so this operation's efficiency is of the essence.

Many efficient NNS algorithms exist for Euclidean and, generally, ℓ_p spaces. Among them are methods based on recursive partitions of the space, e.g., k-d trees and random projection trees (Bentley, 1975; Dasgupta & Freund, 2008; Dasgupta & Sinha, 2015; Keivani & Sinha, 2018), well studied Locality Sensitive Hashing (LSH) (Indyk & Motwani, 1998), and methods based on similarity graphs (Fu et al., 2019; Iwasaki & Miyazaki, 2018; Malkov et al., 2014; Malkov & Yashunin, 2018).

However, hyperbolic geometry is often found to be more suitable for data representation in various domains, including natural language processing, computer vision, graph analysis, and recommendations (Chamberlain et al., 2017; Chami et al., 2020; Khrulkov et al., 2020; Mirvakhabova et al., 2020; Nickel & Kiela, 2017; 2018; Tifrea et al., 2018). Intuitively, the hyperbolic space has advantages over standard representations when data has a latent hierarchical structure which is often the case. Hence, a natural question is: how to perform efficient NNS in hyperbolic spaces? This problem has started to gain attention recently and was addressed in Wu & Charikar (2020). In a nutshell, Wu & Charikar (2020) show how any given *Euclidean* NNS algorithm can be applied to the *hyperbolic* space. To get convergence guarantees, one has to call the chosen NNS method several times, which leads to the increased NNS complexity compared to the original Euclidean algorithm. In the current paper, we choose an alternative approach and apply a leading NNS technique *directly* to the hyperbolic space. We demonstrate that under some assumptions, the proposed method does not have increased complexity and may even *reduce* the query time compared to its Euclidean counterpart.

We focus on graph-based NNS methods that are based on a best-first routing over similarity graphs. Namely, at the pre-processing step, a similarity graph is constructed on the elements of the dataset. Then, given a query $\mathbf{q}$, we start from some element of $\mathcal{D}$ and measure distances between its neighbors and $\mathbf{q}$. The neighbors are added to a priority queue. Then, we pick the closest to $\mathbf{q}$ candidate from this queue and continue the process (Fu et al., 2019; Iwasaki & Miyazaki, 2018; Malkov et al., 2014; Malkov & Yashunin, 2018). It was found empirically that graph-based NNS outperforms other approaches in many large-scale applications (Aumüller et al., 2019). In contrast to other NNS methods, graph-based approaches can be easily applied to non-Euclidean spaces. Indeed, for most algorithms, both graph construction and graph routing can be performed without appealing to the geometry of the underlying space — only the relative ordering of neighbors is used. Thus, one can replace Euclidean distance with the hyperbolic one in a favorite graph-based approach. However, it is a priori not known how graph-based approaches would work in such a non-standard scenario.

In this paper, we show that graph-based approaches are well suited for NNS in hyperbolic spaces. From the theoretical perspective, we analyze the time and space complexity of graph-based NNS, assuming that the data elements are distributed uniformly within a ball of radius R in the hyperbolic space of curvature -1. In this regard, we compare the obtained results with Prokhorenkova & Shekhovtsov (2020) who studied graph-based algorithms assuming the uniform distribution over a d-dimensional sphere. In the dense setting (assuming that $d \ll \log n$ and R does not grow too fast), we derive the time and space complexity using the fact that locally hyperbolic graph-based NNS behaves similarly to the Euclidean case. However, globally, graph-based search in hyperbolic spaces requires *fewer search steps*. Thus, hyperbolic NNS can be even *more efficient* compared to its Euclidean counterpart. In the sparse setting ($d \gg \log n$), we show that hyperbolic NNS has similar complexity to the Euclidean case.

From a practical perspective, we illustrate our theoretical findings via synthetic experiments on uniform data and also analyze graph-based hyperbolic NNS in practical applications where our theoretical results cannot be directly applied. For real datasets, we consider Poincaré GloVe word embeddings (Tifrea et al., 2018) and hyperbolic representations of queries and items for a recommendation task (Shevkunov & Prokhorenkova, 2021). We compare graph-based NNS with existing baselines (Wu & Charikar, 2020) and show that graph-based approaches have significantly better performance. We also compare the performance of graph-based NNS for Euclidean and hyperbolic representations of the same sets of elements and observe that hyperbolic NNS can be more efficient.

2 PRELIMINARIES

Graph-based nearest neighbor search Graph-based algorithms for NNS are known to outperform other approaches in many large-scale applications (Aumüller et al., 2019). The main idea is to construct a proximity graph at the pre-processing stage and traverse this graph towards the query $\mathbf{q}$ at query time. The best-first search is usually used for traversal: we maintain a priority queue, and at each step, we pick the closest to $\mathbf{q}$ candidate from this queue, measure distances between its neighbors and $\mathbf{q}$, and add the neighbors to the queue. Many algorithms are based on this idea (Baranchuk et al., 2019; Fu et al., 2019; Iwasaki & Miyazaki, 2018; Malkov & Yashunin, 2018).

Unfortunately, despite their success in practice, graph-based approaches do not have solid theoretical guarantees. This problem was first approached by Laarhoven (2018) who provided time–space trade-offs for approximate nearest neighbor (ANN) search in sparse datasets ($d \gg \log n$) uniformly distributed on a d-dimensional Euclidean sphere. This work considered nearest neighbor graphs, where each point is connected to its neighborhood lying within a ball of a fixed radius, and assumed greedy graph traversal. Prokhorenkova & Shekhovtsov (2020) extended this result: for the uniform distribution, the authors consider both sparse ($d \gg \log n$) and dense ($d \ll \log n$) regimes, and they analyze the effect of two improvements commonly used in practice: replacing greedy search by best-first (or beam) search and adding long edges to speed up graph traversal at early stages.

In this paper, we aim at understanding graph-based NNS in hyperbolic spaces. For this, we adopt a similar setup to Prokhorenkova & Shekhovtsov (2020), but instead of a uniform distribution over a sphere, we assume a uniform distribution within a ball of a certain radius in the hyperbolic space. The obtained time and space complexity heavily relies on this radius: for small values, the obtained results are similar to the spherical case, while for larger values, the exponential growth of hyperbolic volumes allows us to obtain even faster convergence in the dense regime.

Hyperbolic geometry In this section, we discuss the basics of hyperbolic geometry. Let $\mathbb{R}^{d,1}$ be the $(d+1)$-dimensional *Minkowski space*, i.e., the real vector space $\mathbb{R}^{d+1}$ equipped with the inner product of signature $(d,1)$:

$$\langle \mathbf{x}, \mathbf{y} \rangle_h = -x_0 y_0 + x_1 y_1 + \ldots + x_d y_d. \tag{1}$$

The *vector model* of the *d-dimensional hyperbolic Lobachevsky space* $\mathbb{H}^d$ is the connected component of the standard two-sheeted hyperboloid $H_d = \{\mathbf{x} \in \mathbb{R}^{d,1} \mid \langle \mathbf{x}, \mathbf{x} \rangle_h = -1\}$ contained in the *future light cone*: $\mathbb{H}^d = \{\mathbf{x} \in \mathbb{R}^{d,1} \mid \langle \mathbf{x}, \mathbf{x} \rangle_h = -1, x_0 > 0\}$. The hyperbolic metric is given by $\cosh \rho(\mathbf{x}, \mathbf{y}) = -\langle \mathbf{x}, \mathbf{y} \rangle_h$. From the definition of the hyperbolic space, we see that $x_0 = \sqrt{1 + x_1^2 + \ldots + x_d^2}$. The volume element in $\mathbb{H}^d$ is (see Ratcliffe (2019))

$$\frac{dx_1 \ldots dx_d}{\sqrt{1 + x_1^2 + \ldots + x_d^2}}.$$

Hyperbolic space is known to be useful for data representation in various domains. It is especially convenient for embedding hierarchical data (Nickel & Kiela, 2017). The reason is that the volume of a ball in the hyperbolic space grows exponentially with its radius, in contrast to the Euclidean space, where the growth is polynomial. Formally, let $B(\mathbf{x}, r)$ be a closed hyperbolic ball centered at a point $\mathbf{x} \in \mathbb{H}^d$ of radius $r > 0$. Then,

$$\mathrm{Vol}_{\mathbb{H}^d}(B(\mathbf{x}, r)) = \mathrm{Vol}(\mathbb{S}^{d-1}) \int_0^r \sinh^{d-1}(t)\, dt\,, \tag{2}$$

where $\mathbb{S}^{d-1}$ is the unit $(d-1)$-sphere and $\mathrm{Vol}(\mathcal{M})$ denotes the full volume of a Riemannian manifold $\mathcal{M}$. That is, $\mathrm{Vol}(\mathbb{S}^{d-1}) = 2\pi^{d/2}/\Gamma(d/2)$. For $r \gg \log d$, we have

$$\mathrm{Vol}_{\mathbb{H}^d}(B(\mathbf{x}, r)) \sim \mathrm{Vol}(\mathbb{S}^{d-1})\, e^{r(d-1)}\, \Theta(d^{-1} 2^{-d}). \tag{3}$$

The main term here is $e^{r(d-1)}$, so the ball volume $\mathrm{Vol}_{\mathbb{H}^d}(B(\mathbf{x}, r))$ grows exponentially when r is large. However, for small hyperbolic balls (as $r \to 0$), we have $\sinh(r) \sim r$. Hence, the volume of a ball (2) behaves similarly to the Euclidean case. In Section 4.1, we provide a formal general result that shows that not only the above argument is correct, but also the volume of a measurable subset of a hyperbolic ball behaves asymptotically exactly like its Euclidean analog.

Nearest neighbor search in hyperbolic space While there are many NNS algorithms for ℓ_p spaces, little is known about efficient hyperbolic analogs. Krauthgamer & Lee (2006) proved the existence of an approximate near neighbor algorithm for general geodesic spaces of hyperbolicity δ. From a theoretical perspective, the bound is given in terms of the doubling dimension. For uniform datasets, we may conclude that the obtained time complexity is $2^{O(d)} \log^2 n$, while the constant in the exponent is not given. In contrast, we specify this term as we focus on $d \to \infty$. Another difference is that the algorithm of Krauthgamer & Lee (2006) requires $O(n^2)$ storage that is usually infeasible in practice. In contrast, graph-based algorithms require $n\, M^d$ for some constant M if $d \ll \log n$. Finally, while the theoretical analysis is given in a very general setup, there is no implementation of the proposed algorithm, and extracting an efficient implementation from the proof is challenging.

In a recent paper, Wu & Charikar (2020) reduce the problem of hyperbolic NNS to Euclidean NNS. Namely, assume that we have an efficient Euclidean algorithm and a dataset is represented in the Poincaré ball model. Then, we can apply the algorithm to the Euclidean representation. To guarantee convergence, the algorithm should be applied several times. There are several alternatives proposed. In one of them, we apply the algorithm to the whole dataset, but after each run, appropriately changes the query point to get better hyperbolic neighbors at subsequent iterations. The time complexity of this algorithm is $k+1$ times larger than the Euclidean search if the Euclidean nearest neighbor for the query is the k-th nearest hyperbolic neighbor. In another algorithm, one splits the ball into thin annuli and performs the approximate Euclidean NNS for each of them separately. From a theoretical perspective, it is hard to compare our results since Wu & Charikar (2020) do not consider any particular algorithm. We cannot apply this general result to graph-based algorithms since known guarantees for the Euclidean space assume a uniform distribution. However, if we consider the uniform distribution in the Poincaré model of the hyperbolic space, the corresponding Euclidean distribution is far from being uniform, i.e., the guarantees cannot be transferred. From a practical perspective, we compare graph-based algorithms with the method of Wu & Charikar (2020) and show that graph-based NNS is more efficient.

3 SETUP AND NOTATION

The current paper aims at analyzing how hyperbolic geometry of the underlying space affects the time and space complexity of NNS. In this regard, we compare our results with Prokhorenkova & Shekhovtsov (2020), so we use a similar setup and notation for convenience.

To make the analysis feasible, Laarhoven (2018) and Prokhorenkova & Shekhovtsov (2020) assumed that the dataset is uniformly distributed on a sphere. The uniformity assumption is fairly strong. In Prokhorenkova & Shekhovtsov (2020), this setup was partially motivated by techniques allowing to make a dataset more uniform while approximately preserving the distances (Sablayrolles et al., 2018), which are helpful in some practical applications. We additionally note that in hyperbolic spaces, the uniform distribution of nodes within a ball leads to structures that are similar to real ones: if we consider a geometric graph on such nodes, we obtain the power-law degree distribution, small diameter, and high clustering coefficient (Krioukov et al., 2010), which further supports the meaningfulness of the uniform distribution in our scenario. The compactness of the sphere simplifies the analysis in previous work as it allows for avoiding boundary effects. However, we cannot make this assumption in our setup, as a sphere in the hyperbolic space is isometric to a sphere in the Euclidean space. Hence, to use the advantages of hyperbolic geometry, we assume that the dataset is uniformly distributed within a ball of some radius in the hyperbolic space.

Formally, consider a ball $B(R) \subset \mathbb{H}^d$ of radius $R > 0$. Assume that the dataset is a finite collection of points $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{H}^d$, where $\mathbf{x}_i \in \mathcal{D}$ are i.i.d. random vectors uniformly distributed in $B(R)$. For a given query $\mathbf{q} \in B(R)$, let $\bar{\mathbf{x}} \in \mathcal{D}$ be its nearest neighbor. Then, the goal of the exact NNS is to find $\bar{\mathbf{x}}$. For completeness, following Laarhoven (2018); Prokhorenkova & Shekhovtsov (2020), we also consider c, r-approximate near neighbor problem (c, r-ANN). In this scenario, it is assumed that $\mathbf{q}$ is *planted* uniformly within a fixed distance $r > 0$ from its nearest neighbor $\bar{\mathbf{x}}$ and the goal is to return any $\mathbf{x}'$ such that $\rho(\mathbf{q}, \mathbf{x}') \leq c\,r$ for a fixed value $c > 1$ (Chen et al., 2018). Formally, in the exact nearest neighbor search, we also assume that $\mathbf{q}$ is planted uniformly within a distance r from some element $\mathbf{x}$. However, we do not restrict the radius r (the only restriction is that $\mathbf{x}$ should be the nearest neighbor for $\mathbf{q}$).

We follow the standard assumption that the dimension $d = d(n)$ is an increasing function of n (Chen et al., 2018). We focus on the so-called dense regime, where $d \ll \log n$. For the sparse regime ($d \gg \log n$), we formally show that graph-based NNS in the hyperbolic space has similar complexity to NNS on the sphere.

Graph-based approaches usually imply constructing a nearest neighbor graph (or its approximation), where each element is connected to its k nearest neighbors. As we deal with uniformly distributed datasets, we assume that each node $\mathbf{x}$ is connected to all nodes $\mathbf{y}$ at a distance $\rho(\mathbf{x}, \mathbf{y}) \leq \rho^*$ for some ρ^* chosen at the preprocessing stage. This is more convenient for the analysis, while similar results can be obtained for k-NN graphs, see Appendix D. Hence, at the preprocessing stage, we construct a similarity graph using ρ^*. Next, we receive a query $\mathbf{q}$, sample a random node $\mathbf{x} \in \mathcal{D}$, and at each step of the graph-based greedy descent, we measure the distances between the neighbors of the current node and $\mathbf{q}$ and move to the closest neighbor while we can make such greedy steps.

4 CONVERGENCE AND COMPLEXITY OF GRAPH-BASED NNS

4.1 LOCAL PROPERTIES OF HYPERBOLIC SPACE

Further in this section, we give formal statements regarding the convergence and complexity of greedy graph-based algorithms. Convergence depends on the probability of making a step towards the query from a given element. This probability, in turn, depends on the volumes of some balls' intersections. As follows from our proof, for the dense regime, only 'local' properties are important for convergence. Let us show that under some conditions, the local properties in the hyperbolic space are similar to those in the Euclidean space.

Formally, let $\mathbf{z} = (1, 0, \dots, 0)$ and let $T_{\mathbf{z}}\mathbb{H}^d = \{\mathbf{x} \in \mathbb{R}^{d,1} \mid x_0 = 1\} \simeq \mathbb{R}^d$ be the tangent space. Let $\hat{c}$ be the central projection $\hat{c} : \mathbb{H}^d \to T_{\mathbf{z}}\mathbb{H}^d \simeq \mathbb{R}^d$ from the origin $\mathbf{o} = (0, \dots, 0)$. The following lemma holds.

Lemma 1. *Suppose that $U \subset \mathbb{H}^d$ is a measurable set such that $U \subset B(0,r)$. If $r \ll \frac{1}{\sqrt{d}}$ and $d \to \infty$, then we have* $\mathrm{Vol}_{\mathbb{H}^d}(U) = \mathrm{Vol}_{\mathbb{E}^d}(\hat{c}(U))\left(1 - O\left(d\, r^2\right)\right)$.

This result allows us to apply the bounds on volumes of Euclidean balls intersections to such volumes in the hyperbolic space for small radii.

4.2 Intuition behind the choice of parameters

Our theoretical analysis is asymptotic, meaning that the number of elements n tends to infinity. Hence, we assume that the dimension $d = d(n)$ and the ball radius $R = R(n)$ depend on n. This section discusses our restrictions on $d(n)$ and $R(n)$ and how they affect the analysis. Here, we do not aim at giving mathematically rigorous statements but rather explain underlying intuitions. The formal results can be found in Section 4.3.

Assuming that the dataset is uniformly distributed on a unit sphere, there are two principally different regimes: dense with $d \ll \log n$ and sparse with $d \gg \log n$ (Prokhorenkova & Shekhovtsov, 2020). In the dense regime on a sphere, the distance from any element to its nearest neighbor tends to zero as n grows. In this case, the greedy algorithm reaches the nearest neighbor with probability $1 - o(1)$ if graph degree is M^d for $M > \sqrt{2}$. In the sparse regime, informally speaking, all nodes are at a spherical distance about $\pi/2$ from each other, making the NNS much more challenging. In this case, the graph degree has to grow as n^{φ} for some $\varphi > 0$ in order to solve the approximate near neighbor problem (Prokhorenkova & Shekhovtsov, 2020).

In this paper, we mostly focus on the dense regime with $d \ll \log n$ and argue that for uniform data this setup is more realistic. Indeed, while in practice datasets are often embedded into spaces of large dimension, they usually have much smaller *intrinsic* dimension $d' \ll d$ (Beygelzimer et al., 2006; Lin & Zhao, 2019). Moreover, for sparse uniform datasets on a sphere, graph-based NNS is expected to converge in just two steps (Prokhorenkova & Shekhovtsov, 2020) which contradicts practical observations. Similarly, the optimal graph degree in practice rarely exceeds 64, which roughly corresponds to $d = 12$ in the dense regime (see our discussion on M^d above). It is also important to note that hyperbolic spaces are known to be beneficial when the dimension is small (Gu et al., 2018; Shevkunov & Prokhorenkova, 2021). For completeness of the study, we also analyze the sparse regime and theoretically show that graph-based NNS in the hyperbolic space has similar time and space complexity to NNS on the sphere.

We assume that the dataset is uniformly distributed within a ball $\mathcal{B} = B(R)$ in the hyperbolic space. Changing the radius R is equivalent to changing the curvature of the space (Petersen, 2016). So, we may either assume that the curvature $C = -1$ is fixed while the radius of the ball $R = R(n)$ may change or assume that the radius is fixed while the curvature $C = C(n)$ changes. We further assume that $C = -1$ is fixed. Thus, we have three parameters: the number of elements n, the dimension $d = d(n)$, and the radius $R = R(n)$.

Denote by $V_d(R)$ the volume of $\mathcal{B}$, i.e., $V_d(R) := \mathrm{Vol}_{\mathbb{H}^d}(B(R))$ and by $v_d(R)$ the normalized volume, i.e., $v_d(R) := V_d(R)/\mathrm{Vol}(\mathbb{S}^{d-1})$. Let us analyze (approximately) the distance α from an element to its nearest neighbor. Intuitively, we should have

$$\frac{\mathrm{Vol}_{\mathbb{H}^d}(B(\alpha))}{\mathrm{Vol}_{\mathbb{H}^d}(B(R))} \approx \frac{1}{n}\,, \;\; \mathrm{Vol}_{\mathbb{H}^d}(B(\alpha)) \approx \frac{V_d(R)}{n}\,. \tag{4}$$

Assume for now that $\alpha \ll 1/\sqrt{d}$. In this case, we have (see Section 4.1)

$$\mathrm{Vol}_{\mathbb{H}^d}(B(\alpha)) \sim \mathrm{Vol}_{\mathbb{E}^d}(B(\alpha)) = \frac{\mathrm{Vol}(\mathbb{S}^{d-1})\alpha^d}{d}.$$

Therefore, from (4),

$$\alpha^d \approx \frac{V_d(R)\, d}{n\, \mathrm{Vol}(\mathbb{S}^{d-1})} = \frac{v_d(R)\, d}{n}.$$

Let us find an upper bound on the radius R such that the condition $\alpha \ll 1/\sqrt{d}$ is satisfied. Note that $v_d(R) = O\left(e^{R(d-1)} \cdot 2^{-d} \cdot d^{-1}\right)$ (see Appendix B). Thus, we need

$$\left(\frac{v_d(R)\, d}{n}\right)^{1/d} \ll \frac{1}{\sqrt{d}}, \;\; \left(\frac{e^{R(d-1)}\, 2^{-d}}{n}\right)^{1/d} \ll \frac{1}{\sqrt{d}}, \;\; e^R \ll n^{1/(d-1)} \cdot d^{-1/2}\,.$$

Based on that, we further assume that in the dense regime $e^R \ll n^{1/(d-1)} \cdot d^{-1/2}$. In this case, the distance to the nearest neighbor is $\alpha = o\left(1/\sqrt{d}\right)$, and locally the graph-based search behaves similarly to the Euclidean case. This assumption is natural: we empirically show that larger values of R lead to less efficient NNS.

4.3 Main results

In this section, we formally analyze the performance of graph-based NNS for uniform datasets in the hyperbolic space. As discussed above, to construct the nearest neighbor graph, we connect each point to its neighborhood located within a ball of a fixed radius. In the dense regime, for any constant $M > 1$, let $G(M)$ be a graph obtained by connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ iff

$$\rho(\mathbf{x}_i, \mathbf{x}_j) \leq M \left(\frac{d\, v_d(R)}{n} \right)^{1/d}.$$

This radius is chosen to get about M^d neighbors for each node. The following theorem holds.

Theorem 1. *Assume that $d = d(n)$ is such that $\log\log n \ll d(n) \ll \log n$ and $R = R(n)$ is such that $e^R \ll n^{\frac{1}{d-1}} \cdot d^{-1/2}$. For the exact NNS, let M be a constant s.t. $M > \sqrt{2}$. Then, with probability $1 - o(1)$, $G(M)$-based NNS finds the exact nearest neighbor. For c, r-ANN with some constant $c > 1$, let M be a constant s.t. $M > \sqrt{\frac{4c^2}{3c^2-1}}$. Then, with probability $1 - o(1)$, $G(M)$-based NNS solves c, r-ANN for any r.*

In both cases, the time complexity is $O\left(M^d \cdot n^{1/d} \cdot R \cdot (v_d(R))^{-1/d}\right)$;

the space complexity is $O\left(n \cdot \log n \cdot M^d\right)$.

The formal proof of this theorem is given in Appendices A-C, and in Appendix D we prove the similar theorem for k-NN graphs with $k = [M^d]$.

Theorem 1 covers several different regimes. First, assume that $R \ll 1/\sqrt{d}$. In this case, the geometry is approximately Euclidean. We have $v_d(R) \sim R^d$, so the following corollary holds.

Corollary 1. *If $R = R(n) \ll 1/\sqrt{d}$, then in Theorem 1 the time complexity is $O\left(M^d \cdot n^{1/d}\right)$.*

As expected, this result is similar to the corresponding theorem in Prokhorenkova & Shekhovtsov (2020). However, as R becomes larger, time complexity changes. For instance, if $R \gg \log d$, then we have $v_d(R) \sim e^{R(d-1)}\, \Theta(d^{-1} 2^{-d})$. Thus, we get the following corollary.

Corollary 2. *If $R = R(n) \gg \log d$, then in Theorem 1 we obtain the time complexity $O\left(M^d \cdot n^{1/d} \cdot R \cdot e^{-R(d-1)/d}\right)$.*

From this corollary we see that the time complexity decreases with the radius R. Thus, to make the complexity smaller, we should take larger values of R. According to Theorem 1, we can take R such that $e^R = n^{\frac{1}{d-1}}/(\sqrt{d}\,\varphi)$, where $\varphi = \varphi(n) \to \infty$. We obtain the following corollary.

Corollary 3. *If for an arbitrary $\varphi = \varphi(n) \to \infty$ we have $e^R = n^{\frac{1}{d-1}}/(\sqrt{d}\,\varphi) \to \infty$, then in Theorem 1 we get the time complexity $O\left(M^d \cdot \log n \cdot d^{-1/2} \cdot \varphi\right)$.*

Thus, in the hyperbolic space, the number of steps can be reduced from $n^{1/d}$ to about $\log n$. This shows that NNS over plain nearest neighbor graphs can be more efficient in the hyperbolic space under the conditions considered in this paper. Intuitively, the difference between Euclidean and hyperbolic regimes is due to the fact that the volume of a hyperbolic ball grows exponentially with its radius. Thus, we can faster reach any point from, e.g., the origin of the ball. Note that in the Euclidean space, we can also reduce the number of steps to $\log n$ via adding long edges (Prokhorenkova & Shekhovtsov, 2020). In contrast, in hyperbolic space, this can be achieved without additional tricks. Our experiments on synthetic datasets confirm that without long edges, NNS is much more efficient in the hyperbolic space when dimension d is small.

Finally, let us discuss the sparse regime with $d \gg \log n$. Here we show that the elements are tightly concentrated near the boundary of the ball and thus the convergence guarantees resemble

those on the sphere. In this case, we construct the graph $G(M)$ by connect the elements $\mathbf{x}_i$ and $\mathbf{x}_j$ if $\cosh \rho(\mathbf{x}_i, \mathbf{x}_j) \leq \cosh^2 R - \sinh^2 R \cdot \sqrt{\frac{2M \log n}{d}}$. We also need the following notation. For any constant $c > 1$ let $\alpha_c := \frac{\cosh^2 R - \cosh(\operatorname{arccosh}(\cosh^2 R)/c)}{\sinh^2 R}$. The following theorem holds.

Theorem 2. *Assume that $d \gg \log n$ and $R \gg \log d$. Let $c > 1$ and let M be any constant such that $M < \frac{\alpha_c^2}{\alpha_c^2+1}$. Then, with probability $1 - o(1)$, $G(M)$-based NNS solves c, r-ANN (for any r); the time complexity of this procedure is $\Theta\left(n^{1-M+o(1)}\right)$; the space complexity is $\Theta\left(n^{2-M+o(1)}\right)$.*

The proof is given in Appendix E. This theorem shows that the complexity of hyperbolic NNS for sparse datasets is similar to those on a sphere. However, for large d, the calculation error in the hyperbolic space may become significant even for moderate values of R.

5 Experiments

In this section, we first compare graph-based approaches with the algorithm proposed by Wu & Charikar (2020) on Poincaré GloVe word embeddings (Tifrea et al., 2018) and on the recommendation dataset from (Shevkunov & Prokhorenkova, 2021). Then, we compare the efficiency of graph-based approaches for Euclidean and hyperbolic representations of the same sets of elements. Finally, we illustrate our theoretical results from Section 4.3 on synthetic datasets uniformly distributed within spaces of different curvatures. The code of our experiments supplements the submission.

5.1 Comparison with baseline

We compare graph-based approaches with an existing method of hyperbolic NNS on two datasets. First, we collect the largest publicly available Poincaré GloVe vocabulary[1] of $189,533$ unique tokens. Poincaré GloVe was shown to outperform the Euclidean representation for tasks of similarity, analogy, and hypernymy detection (Tifrea et al., 2018). The dimensionality of the embeddings is 100. The set of tokens is randomly split into the base set of 180K elements and the query set of the remaining elements for evaluation. We also consider the recommendation dataset from Shevkunov & Prokhorenkova (2021), where a DSSM model was trained on a Wikipedia search dataset of (search query, relevant page) pairs. We use their Euclidean and hyperbolic representations for $d = 10$.

On the base set, we construct the HNSW graph (Malkov & Yashunin, 2018) and the NSG graph (Fu et al., 2019), both are considered to be the current state-of-the-art similarity graphs for NNS. As for the search algorithm, we perform the best-first search. The parameter $efSearch$ determines the stopping criteria of the best-first search and allows us to vary the complexity versus accuracy trade-off. For the baseline, the only practical algorithm we are aware of is proposed by Wu & Charikar (2020). The algorithm is called Spherical Shell: it is suggested to split the Poincaré ball into thin annuli and perform approximate Euclidean NNS (the authors use LSH) for some of them separately. Unfortunately, the implementation is not provided by the authors, so we use our implementation. To make the comparison fair, we count the number of distance computations (DCS) instead of the actual time, as the latter may depend on the efficiency of our implementation. The algorithm details and parameter tuning are discussed in Appendix F.

To evaluate the algorithms, we plot Complexity-versus-Accuracy curves. As a primary measure of the search accuracy, we consider `Recall@1`. The search complexity is the average number of DCS per query. The results for Poincaré GloVe are shown in Figure 1. We see that both graph-based approaches significantly outperform the baseline. This confirms that applying an algorithm directly to the hyperbolic space is more efficient than reducing the problem to Euclidean NNS. Among graph-based algorithms, HNSW has better performance on this dataset. Due to space constraints, we provide additional experiments in Appendix F: here we show the results on Poincaré GloVe for $d = 10$ and on the recommendation dataset from Shevkunov & Prokhorenkova (2021). Importantly, the recommendation dataset has different properties compared to Poincaré GloVe — the learned vector norms are sufficiently larger. This property makes the dataset very challenging for Spherical Shell; see Appendix F for the details.

[1] `https://github.com/alex-tifrea/poincare_glove`

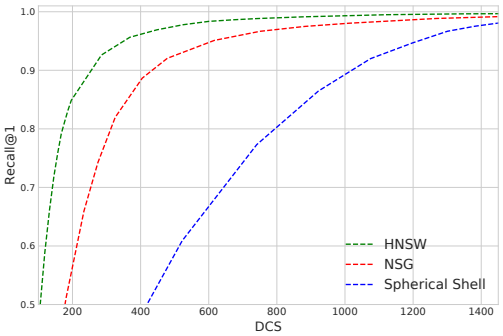


Figure 1: Graph-based methods vs Spherical Shell on Poincaré GloVe embeddings

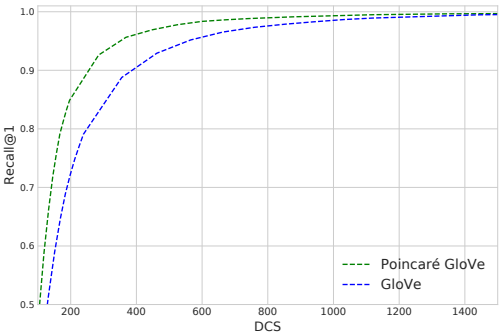


Figure 2: Efficiency of HNSW in hyperbolic space vs Euclidean space

5.2 Euclidean vs Poincaré GloVe

This section additionally illustrates that graph-based methods are well suited for hyperbolic NNS by comparing the efficiency of HNSW for Euclidean and hyperbolic spaces. For this, we construct HNSW graphs for Euclidean GloVe (Pennington et al., 2014) and Poincaré GloVe (Tifrea et al., 2018) with the same tokens and perform NNS for each dataset. Note that the results cannot be compared directly: GloVe and Poincaré GloVe have different geometry, so nearest neighbors for the same queries may differ. Thus, we aim to show that switching from Euclidean to hyperbolic embeddings does not make NNS significantly less efficient.

Figure 2 shows that HNSW built on embeddings in the hyperbolic space is even *more efficient* than HNSW over Euclidean embeddings.[2] Similar experiment for the GloVe embeddings with $d = 10$ and for the recommendation dataset are given in Appendix F.

5.3 Illustrating theoretical results on synthetic datasets

Finally, we illustrate Theorem 1 and our corollary that for simple nearest neighbor graphs and sufficiently small dimension d, graph-based NNS is more efficient in the hyperbolic space. For this, we follow the setup of Prokhorenkova & Shekhovtsov (2020) and perform a synthetic experiment. We generate datasets uniformly distributed in different spaces: 1) on a sphere (Prokhorenkova & Shekhovtsov, 2020), 2) within a Euclidean ball, 3) within a ball of radius R in the hyperbolic space of curvature -1. For all spaces, we fix $n = 10^6$ and vary the dimension $d \in \{2, 4, 6\}$.

For the hyperbolic space, we also vary the radius R. Recall that this is equivalent to changing the curvature. To get an intuition on which values of R are meaningful, recall that Theorem 1 requires $e^R \ll n^{1/(d-1)}/\sqrt{d}$. Thus, roughly speaking, we obtain the following upper bound on R: $R_{max} = \frac{\log n}{d-1} - \log\sqrt{d}$. For $d = 2$, we get $R_{max} \approx 13.5$, for $d = 4$, we get $R_{max} \approx 3.9$, while for $d = 6$ we have $R_{max} \approx 1.9$. Thus, we see that even for $d = 6$ the bound on the radius is sufficiently small and the obtained geometry becomes similar to the Euclidean one (if we consider the theoretically analyzed parameter range). Thus, we do not consider $d > 6$ in this experiment.

First, we analyze the performance of greedy search over nearest neighbor graphs; this setup corresponds to our theoretical analysis. To get the Complexity-versus-Accuracy curves, we vary the graph degree. In Figure 3 we see that spherical and Euclidean spaces are very similar. Hyperbolic space on a small scale, i.e., when points are located inside a ball of a sufficiently small radius R, is similar to the Euclidean one. Then, as R increases, the performance improves. However, after some point, larger values of R reduce the quality of NNS. It is worth noting that this threshold agrees well with the dynamic of R_{max} (while being smaller, as expected).

Then, we analyze what happens if we add long edges known to be essential for the NNS in spherical and Euclidean spaces when the dimension d is small. To incorporate long edges into NN graphs, we follow the procedure described in Prokhorenkova & Shekhovtsov (2020). As expected, long edges

[2]Note that computing the hyperbolic distance has the same complexity as computing the inner product, as follows from Equation (1).

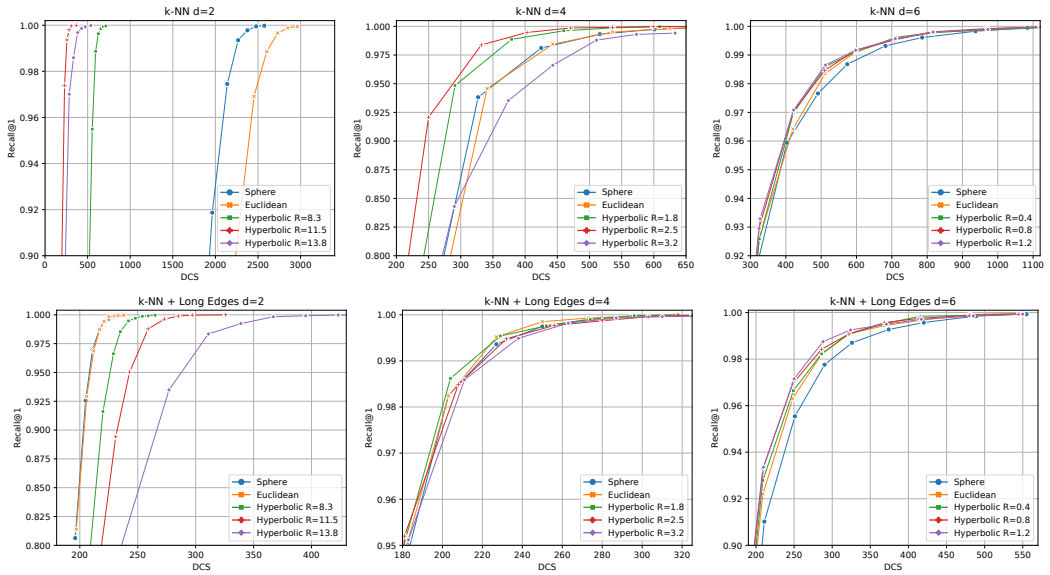


Figure 3: Greedy search over NN graphs without (top row) and with (bottom row) long edges

significantly boost the performance of NNS in spherical and Euclidean spaces. Note that they also improve performance in hyperbolic spaces, but the boost is smaller. As a result, the efficiency of NNS in all spaces becomes similar if R is sufficiently small, while larger values of R even lead to a performance downgrade.

We also conduct similar experiments with greedy search replaced by the best-first search, which is a standard technique for graph-based NNS performance improvement. The conclusions are the same; we include the figures in Appendix F.

6 Conclusion

This paper conducts a theoretical and empirical analysis of graph-based nearest neighbor search in hyperbolic spaces. First, we theoretically analyze the performance of greedy search over nearest neighbor graphs assuming that the dataset is uniformly distributed over a ball of radius R in the hyperbolic space. Assuming the dense regime ($d \ll \log n$) and that the absolute value of the curvature is not too large, we show that locally graph-based exploration is similar to the Euclidean case. However, globally, hyperbolic geometry allows for reducing the number of steps significantly. On the other hand, long edges added to the NN graph boost the performance of NNS in the Euclidean space, so the performance becomes similar to the hyperbolic case.

From a practical perspective, we show that graph-based algorithms outperform other existing methods on several datasets. Also, compared to the Euclidean NNS, switching to hyperbolic space does not lead to significantly increased complexity and may even reduce the query time.

Finally, let us discuss how our theoretical results relate to practice. Clearly, the uniformity assumption is strong and does not hold for real data. However, it allows us to get intuition regarding what would happen in practice. Informally speaking, the conditions on d, R, and n define the density of elements within the hyperbolic space. If the required conditions are locally satisfied, then the convergence with probability $1 - o(1)$ holds for NNS based on kNN graphs with $k = M^d$. In practice, the theoretical guarantees are expected to depend on *intrinsic dimension* d'. Informally speaking, a dataset can be locally approximated by a close-to-uniform distribution in a d'-dimensional space, and then our convergence results can be applied.

We hope that our research will attract more attention to NNS in hyperbolic spaces and, in particular, to graph-based methods.

REFERENCES

Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems*, 2019.

Dmitry Baranchuk, Dmitry Persiyanov, Anton Sinitsin, and Artem Babenko. Learning to route in similarity graphs. *arXiv preprint arXiv:1905.10987*, 2019.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd international conference on Machine learning*, pp. 97–104, 2006.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*, 2017.

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, 2020.

George H Chen, Devavrat Shah, et al. Explaining the success of nearest neighbor methods in prediction. *Foundations and Trends® in Machine Learning*, 10(5-6):337–588, 2018.

Sanjoy Dasgupta and Yoav Freund. Random projection trees and low dimensional manifolds. In *STOC*, volume 8, pp. 537–546. Citeseer, 2008.

Sanjoy Dasgupta and Kaushik Sinha. Randomized partition trees for nearest neighbor search. *Algorithmica*, 72(1):237–263, 2015.

Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. Fast approximate nearest neighbor search with the navigating spreading-out graph. *Proceedings of the VLDB Endowment*, 12(5):461–474, 2019.

Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2018.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613. ACM, 1998.

Masajiro Iwasaki and Daisuke Miyazaki. Optimization of indexing based on k-nearest neighbor graph for proximity search in high-dimensional data. *arXiv preprint arXiv:1810.07355*, 2018.

Omid Keivani and Kaushik Sinha. Improved nearest neighbor search using auxiliary information and priority functions. In *International Conference on Machine Learning*, pp. 2578–2586, 2018.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6418–6428, 2020.

Robert Krauthgamer and James R Lee. Algorithms on negatively curved spaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 119–132. IEEE, 2006.

Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

Thijs Laarhoven. Graph-based time-space trade-offs for approximate near neighbors. In *34th International Symposium on Computational Geometry (SoCG 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Peng-Cheng Lin and Wan-Lei Zhao. Graph based nearest neighbor search: Promises and failures. *arXiv. org*, 2019.

Yury Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. *Information Systems*, 45:61–68, 2014.

Yury A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 203–228. Springer, 2015.

Leyla Mirvakhabova, Evgeny Frolov, Valentin Khrulkov, Ivan Oseledets, and Alexander Tuzhilin. Performance of hyperbolic geometry models on top-n recommendation tasks. In *Fourteenth ACM Conference on Recommender Systems*, pp. 527–532, 2020.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pp. 6338–6347, 2017.

Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pp. 3776–3785, 2018. URL `https://arxiv.org/abs/1806.03417`.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

P. Petersen. *Riemannian Geometry (3rd edition)*. Springer, 2016.

Liudmila Prokhorenkova and Aleksandr Shekhovtsov. Graph-based nearest neighbor search: From practice to theory. In *International Conference on Machine Learning*, pp. 7803–7813. PMLR, 2020.

John G. Ratcliffe. *Foundations of hyperbolic manifolds (3rd edition)*. Springer, 2019.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018.

Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. *Nearest-neighbor methods in learning and vision: theory and practice (neural information processing)*. The MIT press, 2006.

Kirill Shevkunov and Liudmila Prokhorenkova. Overlapping spaces for compact graph representations. *Advances in Neural Information Processing Systems*, 34, 2021.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*, 2018.

Xian Wu and Moses Charikar. Nearest neighbor search for hyperbolic embeddings. *arXiv preprint arXiv:2009.00836*, 2020.

A AUXILIARY GEOMETRICAL RESULTS

In this section, we prove some auxiliary geometrical results that will be useful throughout the proofs. First, we analyze the properties of Euclidean balls and their intersections. Then, we obtain similar results for the hyperbolic space if the ball's radii are sufficiently small.

A.1 EUCLIDEAN BALLS AND THEIR INTERSECTIONS

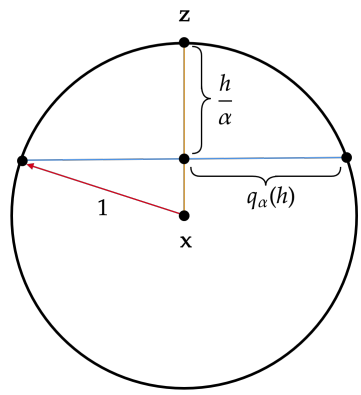
For a ball $B = B(1)$ of unit radius in a d-dimensional Euclidean space, we have that its volume satisfies

$$\mathrm{Vol}_{\mathbb{E}^d}(B) = \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)}.$$

In this section, we estimate volumes of spherical caps, and then use the obtained bounds to estimate the volumes of balls' intersections.

Let $B(\mathbf{x}, \alpha)$ be a ball of radius α centered at $\mathbf{x}$, and $C_\alpha(\mathbf{x}, \mathbf{z}, h) \subseteq B(\mathbf{x}, \alpha)$ be a d–dimensional spherical cap of height h centered at $\mathbf{z} \in \mathcal{S}_\alpha(\mathbf{x})$, i.e., $C_\alpha(\mathbf{x}, \mathbf{z}, h) = \{\mathbf{y} \in B(\mathbf{x}, \alpha) : \langle \mathbf{y}-\mathbf{x}, \mathbf{z}-\mathbf{x}\rangle \geq \alpha(\alpha - h)\}$. Let $C_\alpha(h) := \mathrm{Vol}_{\mathbb{E}^d}\left(C_\alpha(\mathbf{x}, \mathbf{z}, h)\right)$ be the volume of this spherical cap.

Definition 1. *Let $q_\alpha(h) := \left(1 - \left(1 - \frac{h}{\alpha}\right)^2\right)^{1/2}$ be the "relative radius" of a spherical cap (see $q_\alpha(h)$ in the figure).*



Lemma 2. *The volume $C_\alpha(h)$ of a spherical cap $C_\alpha(\boldsymbol{x}, \boldsymbol{z}, h)$ can be estimated as follows:*

$$\Theta(d^{-1}) \leq \frac{C_\alpha(h)}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot \alpha^d \cdot q_\alpha(h)^{d+1}} \leq \Theta(d^{-\frac{1}{2}}).$$

Proof. For brevity, let $q = q_\alpha(h)$. Put $\gamma = \arccos(1 - h/\alpha)$, $0 \leq \gamma \leq \frac{\pi}{2}$. The volume of a spherical cap of height h is known to be:

$$C_\alpha(h) = \mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot \alpha^d \cdot \int_0^\gamma \sin^d(t)\, dt.$$

First, rewrite the integral:

$$\int_0^\gamma \sin^d(t)\, dt = |x = \sin^2 t| = \frac{1}{2}\int_0^{q^2} x^{\frac{d-1}{2}}(1-x)^{-\frac{1}{2}} dx$$

$$= |x = q^2 t| = \frac{1}{2} q^{d+1} \int_0^1 t^{\frac{d-1}{2}} \left(1 - q^2 t\right)^{-\frac{1}{2}} dt.$$

To prove the lemma, it remains to show that

$$\Theta(d^{-1}) \leq \int_0^1 t^{\frac{d-1}{2}} \left(1 - q^2 t\right)^{-\frac{1}{2}} dt \leq \Theta\left(d^{-\frac{1}{2}}\right).$$

Indeed,

$$\Theta(d^{-1}) = \int_0^1 t^{\frac{d-1}{2}} dt \leq \int_0^1 t^{\frac{d-1}{2}} \left(1 - q^2 t\right)^{-\frac{1}{2}} dt$$

$$\leq \int_0^1 t^{\frac{d-1}{2}} (1-t)^{-\frac{1}{2}} dt = \mathrm{B}\left(\frac{d+1}{2}, \frac{1}{2}\right) = \Theta\left(d^{-\frac{1}{2}}\right),$$

where $\mathrm{B}(\cdot,\cdot)$ denotes the beta function. This concludes the proof. □

Let us now estimate the volume of the intersection of two Euclidean balls. Let $W_{\mathbf{x}_i,\mathbf{x}_j}(\alpha,\beta)$ denote the intersection of $B(\mathbf{x}_i,\alpha)$ and $B(\mathbf{x}_j,\beta)$, i.e., $W_{\mathbf{x}_i,\mathbf{x}_j}(\alpha,\beta) = \{\mathbf{y} : \rho(\mathbf{x}_i,\mathbf{y}) \leq \alpha,\ \rho(\mathbf{x}_j,\mathbf{y}) \leq \beta\}$. Let $s := \rho(\mathbf{x}_i,\mathbf{x}_j)$ be the distance between the centers and $W(\alpha,\beta,s) := \mathrm{Vol}_{\mathbb{E}^d}\left(W_{\mathbf{x}_i,\mathbf{x}_j}(\alpha,\beta)\right)$ denote the volume of the intersection. The following lemma estimates the value $W(\alpha,\beta,s)$ under various conditions on α,β,s.

Lemma 3. *W.l.o.g., assume that $\alpha \leq \beta$. Define $A := \frac{(\alpha+\beta+s)(\alpha+\beta-s)(s+\alpha-\beta)(s+\beta-\alpha)}{4s^2}$.*

1. *If $\alpha+\beta < s$, then $W(\alpha,\beta,s) = 0$.*
2. *If $s-\alpha \leq \beta < \sqrt{\alpha^2+s^2}$, then*

$$\Theta\left(d^{-1}\right) \leq \frac{W(\alpha,\beta,s)\cdot\alpha\beta}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B)\cdot(\alpha+\beta)\cdot A^{\frac{d+1}{2}}} \leq \Theta\left(d^{-\frac{1}{2}}\right). \tag{5}$$

3. *If $\sqrt{\alpha^2+s^2} \leq \beta < s+\alpha$, then*

$$\frac{1}{2} \leq \frac{W(\alpha,\beta,s)}{\mathrm{Vol}_{\mathbb{E}^d}(B)\cdot\alpha^d} \leq 1, \tag{6}$$

$$\frac{\mathrm{Vol}_{\mathbb{E}^d}(B(\alpha)) - W(\alpha,\beta,s)}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B)\cdot A^{\frac{d+1}{2}}\cdot\alpha^{-1}} \leq \Theta\left(d^{-\frac{1}{2}}\right) \tag{7}$$

4. *If $s+\alpha \leq \beta$, then*

$$W(\alpha,\beta,s) = \mathrm{Vol}_{\mathbb{E}^d}(B)\cdot\alpha^d. \tag{8}$$

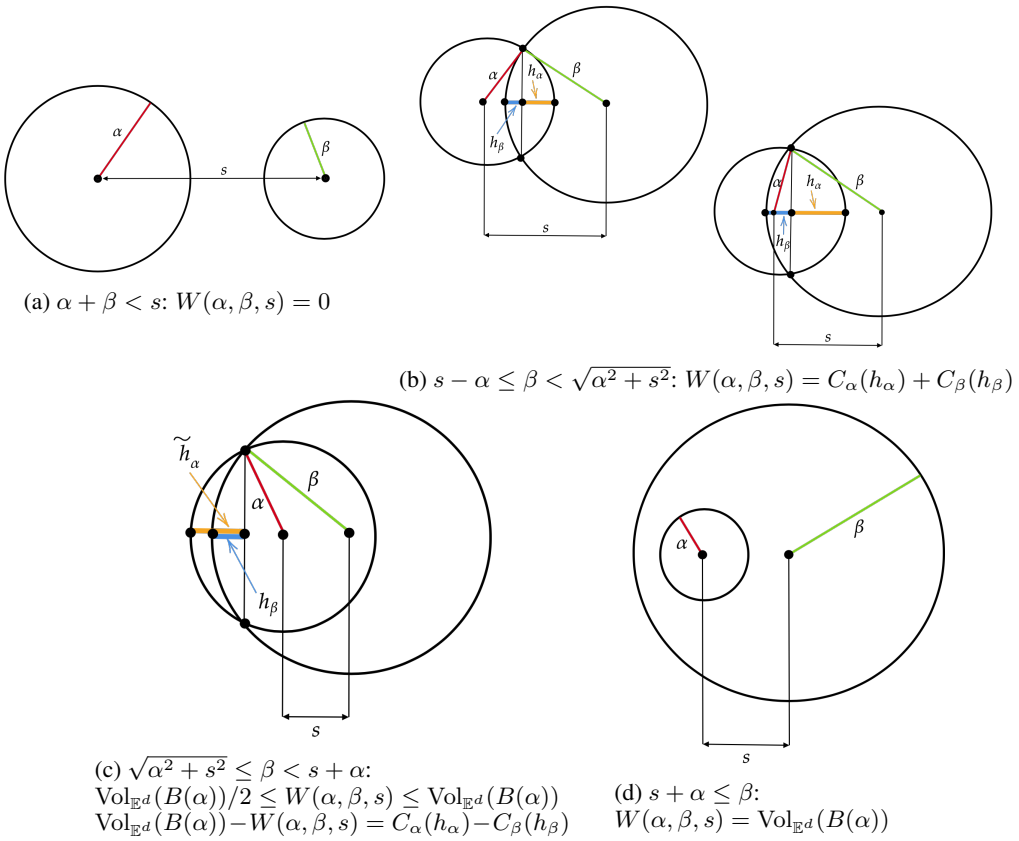
Proof. Let $h_\alpha := \frac{\beta^2-(s-\alpha)^2}{2s}$, $h_\beta := \frac{\alpha^2-(s-\beta)^2}{2s}$, $\widetilde{h}_\alpha := \frac{(\alpha+s)^2-\beta^2}{2s}$.

1. If $\alpha+\beta < s$, then the balls do not intersect and thus $W(\alpha,\beta,s) = 0$, see Figure 4a.
2. If $s-\alpha \leq \beta < \sqrt{\alpha^2+s^2}$, then, according to Figure 4b,

$$W(\alpha,\beta,s) = C_\alpha(h_\alpha) + C_\beta(h_\beta).$$

By Lemma 2, we have

$$\Theta\left(d^{-1}\right) \leq \frac{W(\alpha,\beta,s)}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B)\cdot\left(\alpha^d q_\alpha^{d+1}(h_\alpha) + \beta^d q_\beta^{d+1}(h_\beta)\right)} \leq \Theta\left(d^{-\frac{1}{2}}\right).$$



(a) $\alpha + \beta < s$: $W(\alpha, \beta, s) = 0$

(b) $s - \alpha \leq \beta < \sqrt{\alpha^2 + s^2}$: $W(\alpha, \beta, s) = C_\alpha(h_\alpha) + C_\beta(h_\beta)$

(c) $\sqrt{\alpha^2 + s^2} \leq \beta < s + \alpha$:
$\mathrm{Vol}_{\mathbb{E}^d}(B(\alpha))/2 \leq W(\alpha, \beta, s) \leq \mathrm{Vol}_{\mathbb{E}^d}(B(\alpha))$
$\mathrm{Vol}_{\mathbb{E}^d}(B(\alpha)) - W(\alpha, \beta, s) = C_\alpha(h_\alpha) - C_\beta(h_\beta)$

(d) $s + \alpha \leq \beta$:
$W(\alpha, \beta, s) = \mathrm{Vol}_{\mathbb{E}^d}(B(\alpha))$

Figure 4: Different cases in the proof of Lemma 3

Substituting $q_\alpha,\ q_\beta,\ h_\alpha,\ h_\beta$, we obtain that

$$\begin{aligned}
&\alpha^d q_\alpha^{d+1}(h_\alpha) + \beta^d q_\beta^{d+1}(h_\beta) \\
&\qquad = \alpha^d \left(\frac{\alpha^2 - (\alpha - h_\alpha)^2}{\alpha^2}\right)^{\frac{d+1}{2}} + \beta^d \left(\frac{\beta^2 - (\beta - h_\beta)^2}{\beta^2}\right)^{\frac{d+1}{2}} \\
&\qquad = \frac{1}{\alpha}\left(h_\alpha(2\alpha - h_\alpha)\right)^{\frac{d+1}{2}} + \frac{1}{\beta}\left(h_\beta(2\beta - h_\beta)\right)^{\frac{d+1}{2}} \\
&= \frac{1}{\alpha}\left(\frac{(\beta^2 - (s-\alpha)^2)((s+\alpha)^2 - \beta^2)}{4s^2}\right)^{\frac{d+1}{2}} + \frac{1}{\beta}\left(\frac{(\alpha^2 - (s-\beta)^2)((s+\beta)^2 - \alpha^2)}{4s^2}\right)^{\frac{d+1}{2}} \\
&\qquad = \left(\frac{1}{\alpha} + \frac{1}{\beta}\right) A^{\frac{d+1}{2}}.
\end{aligned}$$

3. If $\sqrt{\alpha^2 + s^2} \le \beta < s + \alpha$, then (see Figure 4c)

$$\frac{\mathrm{Vol}_{\mathbb{E}^d}(B(\alpha))}{2} \le W(\alpha, \beta, s) \le \mathrm{Vol}_{\mathbb{E}^d}(B(\alpha)),$$

Also, according to Figure 4c,

$$\mathrm{Vol}_{\mathbb{E}^d}(B(\alpha)) - W(\alpha, \beta, s) = C_\alpha(\widetilde{h}_\alpha) - C_\beta(h_\beta) \le C_\alpha\left(\widetilde{h}_\alpha\right).$$

By Lemma 2, we have

$$\frac{\mathrm{Vol}_{\mathbb{E}^d}(B) \cdot \alpha^d - W(\alpha, \beta, s)}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot \alpha^d \cdot q_\alpha^{d+1}(\widetilde{h}_\alpha)} \le \Theta\left(d^{-\frac{1}{2}}\right).$$

Substituting q_α and $\widetilde{h}_\alpha$, we get

$$\begin{aligned}
\alpha^d q_\alpha^{d+1}(\widetilde{h}_\alpha) = \alpha^d \left(\frac{\alpha^2 - (\alpha - \widetilde{h}_\alpha)^2}{\alpha^2}\right)^{\frac{d+1}{2}} &= \frac{1}{\alpha}\left(\widetilde{h}_\alpha(2\alpha - \widetilde{h}_\alpha)\right)^{\frac{d+1}{2}} \\
&= \frac{1}{\alpha}\left(\frac{((s+\alpha)^2 - \beta^2)(\beta^2 - (s-\alpha)^2)}{4s^2}\right)^{\frac{d+1}{2}} = \frac{1}{\alpha} A^{\frac{d+1}{2}}.
\end{aligned}$$

4. If $s + \alpha \le \beta$, then $W(\alpha, \beta, s) = \mathrm{Vol}_{\mathbb{E}^d}(B(\alpha)) = \mathrm{Vol}_{\mathbb{E}^d}(B) \cdot \alpha^d$, see Figure 4d.

□

The following lemma will also be helpful in our proofs.

Lemma 4. *$W(\alpha, \beta, \alpha)$ and $W(\beta, \alpha, \alpha)$ are non-decreasing functions of α.*

Proof. Let us consider the configuration of balls shown in Figure 5. The d-dimensional ball centered at $\mathbf{q}'$ is contained within the ball centered at $\mathbf{q}$. Thus, the intersection $\mathbf{A}'\mathbf{B}'\mathbf{x}$ of the balls centered at $\mathbf{x}$ and $\mathbf{q}'$ is contained in the intersection $\mathbf{ABx}$ of the balls centered at $\mathbf{x}$ and $\mathbf{q}$. Clearly, the center $\mathbf{q}'$ of the ball which is closer to $\mathbf{x}$ can be taken on the geodesic between $\mathbf{q}$ and $\mathbf{x}$ without loss of generality. The same argument works when the ball centered at $\mathbf{x}$ contains the centers $\mathbf{q}$, $\mathbf{q}'$ of the other balls. □

A.2 Hyperbolic volumes

First, we briefly recall the basic properties of hyperbolic space, and then move towards the study of hyperbolic volumes and their Euclidean counterparts. It turns out that on the small scale hyperbolic volumes behave in an almost Euclidean way, while on larger scale their behaviour is fundamentally different.

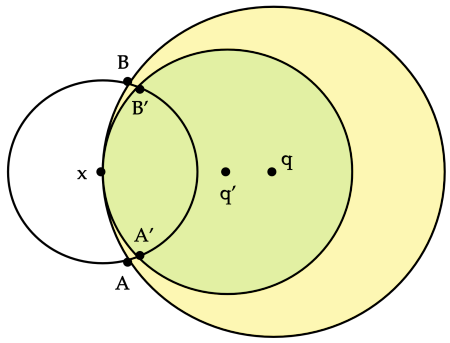


Figure 5: Monotonicity of $W(\alpha, \beta, \alpha)$ and $W(\beta, \alpha, \alpha)$

A.2.1 Background on hyperbolic space

Let $\mathbb{R}^{d,1}$ be the $(d+1)$-dimensional *Minkowski space*, i.e., the real vector space of dimension $(d+1)$ equipped with the inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle_h = -x_0 y_0 + x_1 y_1 + \ldots + x_d y_d$$

of signature $(d, 1)$.

The *vector model* of the *d-dimensional hyperbolic Lobachevsky space* $\mathbb{H}^d$ is the connected component of the standard two-sheeted hyperboloid contained in the *future light cone*:

$$\mathbb{H}^d = \{\mathbf{x} \in \mathbb{R}^{d,1} \mid \langle \mathbf{x}, \mathbf{x} \rangle_h = -1, x_0 > 0\}. \tag{9}$$

The *hyperbolic metric* is given by

$$\cosh \rho(\mathbf{x}, \mathbf{y}) = -\langle \mathbf{x}, \mathbf{y} \rangle_h.$$

The coordinates $(x_0, x_1, \ldots, x_d)$ satisfy the following system of equations:

$$\begin{aligned} x_0 &= \cosh(t_1), \\ x_1 &= \sinh(t_1)\cos(t_2), \\ &\ldots \\ x_{d-1} &= \sinh(t_1)\sin(t_2)\ldots\sin(t_{d-1})\cos(t_d), \\ x_d &= \sinh(t_1)\sin(t_2)\ldots\sin(t_{d-1})\sin(t_d). \end{aligned}$$

This hyperbolic coordinate parameterization of $\mathbb{H}^d$ is the map

$$\chi\colon [0, +\infty) \times [0, \pi]^{d-2} \times [0, 2\pi] \to \mathbb{H}^d$$

defined by

$$\chi(t_1, \ldots, t_d) = (x_0, x_1, \ldots, x_d).$$

A subset $U \subset \mathbb{H}^d$ is called measurable if $\chi^{-1}(U)$ is measurable with respect to the standard Lebesgue measure on $\mathbb{R}^d$. Let $U \subset \mathbb{H}^d$ be a measurable set. Then, the hyperbolic volume of U is defined by

$$\mathrm{Vol}_{\mathbb{H}^d}(U) = \int\limits_{\chi^{-1}(U)} \sinh^{d-1}(t_1)\sin^{d-2}(t_2)\ldots\sin(t_{d-1})\, dt_1 dt_2 \ldots dt_{d-1} dt_d.$$

A.2.2 Approximating small hyperbolic volumes by their Euclidean counterparts

In this section, we show that small hyperbolic volumes can be approximated by Euclidean ones. This is very helpful for our analysis and will allow us to obtain analogs of Lemma 2 and 3 for small hyperbolic balls.

From the definition of the hyperbolic space (9), we see that $x_0 = \sqrt{1 + x_1^2 + \ldots + x_d^2}$. Let $U \subset \mathbb{H}^d$ be a measurable set containing the point $\mathbf{z} = (1, 0, \ldots, 0)$, and let U' be the projection of U along the x_0–coordinate from the hyperboloid onto the tangent space $T_{\mathbf{z}}\mathbb{H}^d = \{\mathbf{x} \in \mathbb{R}^{d,1} \mid x_0 = 1\} \simeq \mathbb{R}^d$ realised by the map $p : \mathbb{H}^d \to \mathbb{R}^d$. Let c be the central projection $c : \mathbb{H}^d \to T_{\mathbf{z}}\mathbb{H}^d \simeq \mathbb{R}^d$ from the origin $\mathbf{o} = (0, \ldots, 0)$. Let $F : U' \to \hat{c}U)$ be the map that satisfies $c = F \circ p$.

Let us note that the central projection c maps balls centered at $\mathbf{z}$ in $\mathbb{H}^d$ to balls centered at $p(\mathbf{z}) = \mathbf{z}$ in $\mathbb{R}^d$, and also spherical caps of such balls in $\mathbb{H}^d$ have spherical caps in $\mathbb{R}^d$ as their respective images, since every spherical cap is an intersection of some ball with a half-space, which is bounded by some hyperplane in $\mathbb{R}^{d,1}$ (the central projection of this hyperplane is also a hyperplane in $\mathbb{R}^d$).

According to Ratcliffe (2019),

$$\mathrm{Vol}_{\mathbb{H}^d}(U) = \int\limits_{U'} \frac{dx_1 \ldots dx_d}{\sqrt{1 + x_1^2 + \ldots + x_d^2}}.$$

We need to compare $\mathrm{Vol}_{\mathbb{H}^d}(U)$ to $\mathrm{Vol}_{\mathbb{E}^d}(\hat{c}U))$ in order to conclude that these quantities approximate each other in some appropriate regime.

Lemma 5. *Suppose that $U \subset \mathbb{H}^d$ is a measurable set such that $U \subset B(0, r)$. If $r \ll \frac{1}{\sqrt{d}}$ and $d \to \infty$, then we have*

$$\mathrm{Vol}_{\mathbb{H}^d}(U) = \mathrm{Vol}_{\mathbb{E}^d}(\hat{c}U)) \left(1 - O\left(d\, r^2\right)\right).$$

Proof. We have that $U' = p(U) \subset B_{\mathbb{R}^d}(0, \sinh(r))$, hence $x_1^2 + \ldots + x_d^2 \leq \sinh^2(r)$. Then,

$$\int\limits_{U'} \frac{dx_1 \ldots dx_d}{\cosh(r)} < \mathrm{Vol}_{\mathbb{H}^d}(U) = \int\limits_{U'} \frac{dx_1 \ldots dx_d}{\sqrt{1 + x_1^2 + \ldots + x_d^2}} < \int\limits_{U'} dx_1 \ldots dx_d = \mathrm{Vol}_{\mathbb{E}^d}(U').$$

By using Taylor series expansion, we obtain

$$\int\limits_{U'} \frac{dx_1 \ldots dx_d}{\cosh(r)} = \frac{1}{\cosh(r)} \mathrm{Vol}_{\mathbb{E}^d}(U') = \left(1 - \frac{r^2}{2} + \Theta\left(r^4\right)\right) \mathrm{Vol}_{\mathbb{E}^d}(U'),$$

and thus

$$\left(1 - \frac{r^2}{2} + \Theta\left(r^4\right)\right) \mathrm{Vol}_{\mathbb{E}^d}(U') < \mathrm{Vol}_{\mathbb{H}^d}(U) < \mathrm{Vol}_{\mathbb{E}^d}(U').$$

Now let us show that

$$\mathrm{Vol}_{\mathbb{E}^d}(U') = \mathrm{Vol}_{\mathbb{E}^d}(\hat{c}U)) \left(1 - O\left(dr^2\right)\right).$$

Indeed, the map $F : U' \to \hat{c}U)$ satisfies $c = F \circ p$ and is given by $F = (F_1, \ldots, F_d)$ with

$$F_i : x_i \mapsto \frac{x_i}{\sqrt{1 + x_1^2 + \ldots + x_d^2}}, \text{ for } i = 1, \ldots, d,$$

and thus the Jacobian of F satisfies

$$\det\left(\frac{\partial F_i}{\partial x_j}\right)_{i,j=1}^{d} = \frac{1}{(1 + x_1^2 + \ldots + x_d^2)^{\frac{d+2}{2}}} = 1 - O\left(dr^2\right),$$

whenever $x_1^2 + \ldots + x_d^2 \leq \sinh^2(r)$ and $dr^2 \to 0$. One can conveniently use spherical coordinates in order to perform the respective computation. The lemma now follows from standard Taylor series techniques. □

Let $B_d(\mathbf{x}, r)$ be a closed hyperbolic ball centered at a point $\mathbf{x} \in \mathbb{H}^d$ of radius $r > 0$. Then we can compute its volume as Ratcliffe (2019):

$$\mathrm{Vol}_{\mathbb{H}^d}(B_d(\mathbf{x}, r)) = \mathrm{Vol}\left(\mathbb{S}^{d-1}\right) \int\limits_0^r \sinh^{d-1}(t)\, dt.$$

For $r \gg \log d$, we have

$$\mathrm{Vol}_{\mathbb{H}^d}(B_d(\mathbf{x}, r)) \sim \mathrm{Vol}(\mathbb{S}^{d-1})\, \exp(r(d-1))\, \Theta(d^{-1} 2^{-d}).$$

For $r \ll 1/\sqrt{d}$, we have from Lemma 5 that

$$\mathrm{Vol}_{\mathbb{H}^d}(B_d(\mathbf{x}, r)) \sim \mathrm{Vol}_{\mathbb{E}^d}(B_d(\mathbf{x}, r)) \sim \mathrm{Vol}(\mathbb{S}^{d-1})\, d^{-1}\, r^d\, \Theta(1).$$

A.2.3 SMALL HYPERBOLIC BALLS AND THEIR INTERSECTIONS

Lemma 5 allows us to transfer our results for Euclidean balls and their intersections to the hyperbolic space when the radii of the balls are sufficiently small. Let $C_\alpha(h)$ be the volume of a spherical cap of height h in the hyperbolic space. The following lemma follows from Lemma 2 and Lemma 5.

Lemma 6. *For $\alpha \ll 1/\sqrt{d}$, the volume of a hyperbolic spherical cap can be estimated as follows:*

$$\Theta(d^{-1}) \leq \frac{C_\alpha(h)}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot \alpha^d \cdot q_\alpha(h)^{d+1}} \leq \Theta(d^{-\frac{1}{2}}).$$

Let us now estimate the volume of the intersection of two hyperbolic balls. As before, denote by $W_{\mathbf{x}_i,\mathbf{x}_j}(\alpha,\beta)$ the intersection of $B(\mathbf{x}_i,\alpha)$ and $B(\mathbf{x}_j,\beta)$, i.e., $W_{\mathbf{x}_i,\mathbf{x}_j}(\alpha,\beta) = \{\mathbf{y} : \rho(\mathbf{x}_i,\mathbf{y}) \leq \alpha,\ \rho(\mathbf{x}_j,\mathbf{y}) \leq \beta\}$. Let $s := \rho(\mathbf{x}_i,\mathbf{x}_j)$ be the distance between the centers and $W(\alpha,\beta,s) := \mathrm{Vol}_{\mathbb{H}^d}\left(W_{\mathbf{x}_i,\mathbf{x}_j}(\alpha,\beta)\right)$ denote the volume of the intersection. The following lemma estimates the value $W(\alpha,\beta,s)$ for various relationships between α,β,s. Without loss of generality, we assume that $\alpha \leq \beta$.

Lemma 7. *Assume that $\alpha \ll 1/\sqrt{d}$, $\beta \ll 1/\sqrt{d}$. W.l.o.g., we may also assume that $\alpha \leq \beta$. Define $A := \frac{(\alpha+\beta+s)(\alpha+\beta-s)(s+\alpha-\beta)(s+\beta-\alpha)}{4s^2}$.*

1. *If $\alpha+\beta < s$, then $W(\alpha,\beta,s) = 0$.*
2. *If $s-\alpha \leq \beta < \sqrt{\alpha^2+s^2}$, then*
$$\Theta\left(d^{-1}\right) \leq \frac{W(\alpha,\beta,s)\cdot\alpha\beta}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B)\cdot(\alpha+\beta)\cdot A^{\frac{d+1}{2}}} \leq \Theta\left(d^{-\frac{1}{2}}\right). \tag{10}$$
3. *If $\sqrt{\alpha^2+s^2} \leq \beta < s+\alpha$, then*
$$\frac{1}{2} \leq \frac{W(\alpha,\beta,s)}{\mathrm{Vol}_{\mathbb{E}^d}(B)\cdot\alpha^d\cdot(1+o(1))} \leq 1, \tag{11}$$
$$\frac{\mathrm{Vol}_{\mathbb{H}^d}(B(\alpha)) - W(\alpha,\beta,s)}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B)\cdot A^{\frac{d+1}{2}}\cdot\alpha^{-1}} \leq \Theta\left(d^{-\frac{1}{2}}\right). \tag{12}$$
4. *If $s+\alpha \leq \beta$, then*
$$W(\alpha,\beta,s) = \mathrm{Vol}_{\mathbb{H}^d}(B(\alpha)) \sim \mathrm{Vol}_{\mathbb{E}^d}(B)\cdot\alpha^d. \tag{13}$$

Proof. To prove this lemma, we follow the proof of Lemma 3, and use Lemma 6 instead of Lemma 2. □

The following lemma is an analog of Lemma 4. Indeed, the argument uses only monotonicity of volumes under set-theoretic inclusions, which holds for both Euclidean and hyperbolic spaces.

Lemma 8. *$W(\alpha,\beta,\alpha)$ and $W(\beta,\alpha,\alpha)$ are non-decreasing functions of α.*

B PROOF OVERVIEW

In this section, we overview the proof of Theorem 1.

Recall that we assume that the distribution of the dataset is uniform over a d-dimensional ball $\mathcal{B}$ of radius $R = R(n)$ such that

$$e^R \ll \frac{n^{1/(d-1)}}{d^{1/2}}. \tag{14}$$

Then, the graph $G(M)$ is constructed by using balls of radius $\alpha_M = M\delta$ with

$$\delta = \left(\frac{dv_d(R)}{n}\right)^{1/d}. \tag{15}$$

Let us show that under the conditions of the theorem, we have $\delta \ll 1/\sqrt{d}$. First, note that

$$v_d(R) = \int\limits_0^R \sinh^{d-1}(t)\, dt \leq \int\limits_0^R \frac{e^{t(d-1)}}{2^{d-1}}\, dt \leq \frac{e^{R(d-1)}}{2^{d-1}(d-1)}\,.$$

Thus, we have

$$\delta \leq \left(\frac{de^{R(d-1)}}{2^{d-1}(d-1)n}\right)^{1/d} = \Theta\left(e^{R(d-1)/d} n^{-1/d}\right) \ll 1/\sqrt{d}.$$

The last inequality is due to the conditions of Theorem 1. Thus, we can apply Lemma 7 throughout the proof.

To prove that the algorithm succeeds w.h.p., we need to guarantee that it does not stop in a local optimum until we find the exact (or approximate) nearest neighbor of the query **q**. Given a point **x** such that $\rho(\mathbf{q}, \mathbf{x}) = \alpha_s$, the probability of making a step towards **q** is determined by $W(\alpha_s, \alpha_M, \alpha_s)$: the distance between the points is α_s, the neighbors of **x** are within the ball of radius α_M centered at **x**, and we accept those neighbors that are closer to **q** than **x**, i.e., within a ball of radius α_s centered at **q**. Lemma 9 analyzes the asymptotic behavior of $W(\alpha_s, \alpha_M, \alpha_s)$.

Lemma 4 states that $W(\alpha, \beta, \alpha)$ is a non-increasing function of α. Informally, it means that it is easier to make steps towards the query when $\rho(\mathbf{x}, \mathbf{q})$ is larger. In contrast, when **x** becomes closer to **q**, the probability of getting stuck in a local optimum increases. Therefore, to obtain lower bounds on $W(\alpha_s, \alpha_M, \alpha_s)$, we can consider only the case when α_s is small. Below we assume that $\alpha_s = s\delta$, where s is some constant.

Lemma 11 gives the following lower bounds on $W(\alpha_s, \alpha_M, \alpha_s)$:

1. if $M < s\sqrt{2}$, then $\frac{W(\alpha_s, \alpha_M, \alpha_s)}{V_d(R)} \geq \frac{1}{n} \cdot \left(M^2 - \frac{M^4}{4s^2}\right)^{\frac{d}{2}+o(d)}$;
2. if $M \geq s\sqrt{2}$, then $\frac{W(\alpha_s, \alpha_M, \alpha_s)}{V_d(R)} \geq \frac{1}{n} \cdot s^{d+o(d)}$.

Let us consider one step of the algorithm. We can move forward w.h.p. if $W(\alpha_s, \alpha_M, \alpha_s) \gg 1/n$. Formally speaking, Corollary 4 of Lemma 11 states that if M and s satisfy

$$s > 1 \text{ and either } M \geq s\sqrt{2} \text{ or } M^2 - M^4/4s^2 > 1, \tag{16}$$

then there exists a constant $S > 1$ such that we can make a step towards **q** with probability

$$1 - \left(1 - \frac{S^{d+o(d)}}{n}\right)^{n-1} = 1 - O\left(e^{-S^{d+o(d)}}\right)$$

since the expected number of nodes in the intersection of balls is much greater than 1.

We make steps of the greedy algorithm towards the query **q** as long as the probability of making the next step is large. Below we also prove that the overall probability of failure (taking into account all steps of the algorithm) is small. Now, let us show that under the conditions of the theorem, we reach some neighborhood of **q** and then either find the approximate near neighbor or find the exact nearest neighbor by making one more step.

First, let us consider the exact NNS. In this case, we assume that the nearest neighbor is located within a ball of radius r, $0 < r \leq 1$, centered at **q**. Our goal is to find the exact nearest neighbor. According to the statement of the theorem, we have $M > \sqrt{2}$. Observe that this implies

$$\frac{M}{\sqrt{2}} > \frac{M^2}{2\sqrt{M^2-1}}. \tag{17}$$

While we move towards the query, s becomes smaller. Let us figure out up to which s we can make steps. According to (16), we need either $1 < s \leq M/\sqrt{2}$ or $s > M^2/2\sqrt{M^2-1}$. According to (17), it reduces to $s > 1$. In other words, we can move forward to any $s > 1$. Then, according to Lemma 14, it is sufficient to have $M^2 > s^2 + r^2$ to reach the exact nearest neighbor of **q** in one step.

Thus, we can take any s s.t. $1 < s < \sqrt{M^2 - r^2}$: we can reach this distance with greedy search and then jump directly to the nearest neighbor with high probability.

Now, let us turn to the approximate near neighbor search c, r-ANN. In this case, we need to find an element that is within a distance rc from $\mathbf{q}$. According to the statement of the theorem, we have $M > \sqrt{4c^2/(3c^2 - 1)}$.

Note that $F(r) := 2r^2c^2\left(1 - \sqrt{1 - 1/r^2c^2}\right)$ decreases with r for $r > 1/c$ and equals 2 at $r = 1/c$, while $G(r) := \frac{2}{3}\left(r^2 + 1 + \sqrt{r^4 - r^2 + 1}\right)$ increases with r and equals 2 at $r = 1$. Both $F(r)$ and $G(r)$ attain $4c^2/(3c^2 - 1)$ at $r_0 = \sqrt{4c^2/(c^2 + 1)(3c^2 - 1)}$. Therefore, we either have

$$1 \geq r > r_0 \text{ and } M^2 > F(r) \tag{18}$$

or

$$r_0 \geq r > \frac{1}{c} \text{ and } M^2 > G(r). \tag{19}$$

To solve c, r-ANN, either we can reach s such that $rc > s$ or we can reach s such that at the next step we can find the exact nearest neighbor. Let us consider the first possibility. Note that

$$s > 1 \text{ and } M^2 > 2s^2\left(1 - \sqrt{1 - \frac{1}{s^2}}\right) \tag{20}$$

is equivalent to (16). In turn, since $2s^2(1 - \sqrt{1 - 1/s^2})$ is a decreasing function of s on $s > 1$, we get that (18) implies the existence of s such that $rc > s > 1$ and

$$M^2 > 2s^2\left(1 - \sqrt{1 - \frac{1}{s^2}}\right) > F(r).$$

Hence, since (16) holds, we can move forward to such s. Let us consider the second possibility. Observe that (19) is equivalent to

$$M^2 - r^2 > \frac{M^4}{4(M^2 - 1)}.$$

Therefore, there exists s such that the condition $s^2 < M^2 - r^2$ of Lemma 14 and (16) hold. Since (16) holds, we can move forward to such s and reach the exact nearest neighbor in one step as the condition of Lemma 14 is satisfied.

Above we show that under the conditions of the theorem, each step of the greedy search is successful with probability $1 - O\left(e^{-S^{d+o(d)}}\right)$. To formally prove that the algorithm succeeds with high probability, we need to limit the number of steps and estimate the overall probability of failure. For this, we show that at each step of the algorithm we become closer to the query by some fixed positive value. In Lemma 12, we prove that each step reduces the distance from $\mathbf{q}$ by at least $\varepsilon\delta$, where ε is some positive constant. Clearly, we start at a distance $O(R)$ from q. Hence, we are able to bound the number of steps by $O(R/\varepsilon\delta)$ which is $O\left(Rn^{1/d}(v_d(R))^{-1/d}\right)$.

To estimate the overall success probability, we have to consider the dependence of consecutive steps of the algorithm. Lemma 13 implies that consecutive steps are "almost independent" and the overall success probability is $1 - O\left(Rn^{1/d}(v_d(R))^{-1/d}e^{-S^{d+o(d)}}\right)$. The assumption $d \gg \log\log n$ and the fact that $v_d(R) = \Omega(R^d)$ yield $O\left(Rn^{1/d}(v_d(R))^{-1/d}e^{-S^{d+o(d)}}\right) = O\left(e^{\frac{\log n}{d} - S^{d+o(d)}}\right) = o(1)$.

While in most of the analysis we do not consider the cases when α_M-neighborhoods of the dataset elements intersect the boundary $\partial\mathcal{B}$, we need to show that possible intersections do not affect the result. The formal statement of this fact is given in Lemma 15.

Finally, let us discuss the time and space complexities of the algorithm. Let v be a fixed arbitrary node of $G(M)$ and let $N(v)$ denote the (random) number of its neighbors in $G(M)$. Let $f = f(n) = (n - 1)\mathrm{Vol}_{\mathbb{H}^d}\left(B(\alpha_M)\right)/\mathrm{Vol}_{\mathbb{H}^d}\left(\mathcal{B}\right)$ denote the upper bound for the expected number of neighbors of v (this is an upper bound due to possible boundary effects). In Lemma 16, we show that $N(v) \leq 3f/2$ with probability at least $1 - O(1/f)$.

To obtain the final time complexity of graph-based NNS, we sum up the complexity of all algorithm's steps. In Lemma 17, we show that if k steps of the NNS are made then the time complexity is $O(kfd)$ with probability $1 - O(1/kf)$.

In Lemma 18, we show that the number of edges in the graph is $E(G) \leq 3fn/4$ with probability $1 - O(1/fn)$. If we store the graph as the adjacency lists, then the space complexity is $O(E(G) \cdot \log n) = O(fn \log n)$.

Due to the choice of δ, we have $f = O\left(M^d\right)$. Hence, w.h.p. the space complexity is $O\left(n \cdot \log n \cdot M^d\right)$ and the time complexity is $O\left(R \cdot n^{1/d} \cdot (v_d(R))^{-1/d} \cdot M^d\right)$.

C Results used throughout the proof

In this section, we formulate supplemental statements used in the proof. We need these results for the hyperbolic space, but they are also true for the Euclidean one. Thus, we do not specify a particular space in $W(\cdot,\cdot,\cdot)$.

C.1 Analysis of convergence

Recall that $\alpha_M = M\delta$ and $\alpha_s = s\delta$, where M and s are some constants and δ is defined in (15). Recall that $\delta \ll 1/\sqrt{d}$.

Lemma 9. *Define* $A := \frac{\alpha_M^2(4\alpha_s^2 - \alpha_M^2)}{4\alpha_s^2}$.

If $\alpha_M \leq \alpha_s\sqrt{2}$, *then*

$$\Theta\left(d^{-1}\right) \leq \frac{W(\alpha_s, \alpha_M, \alpha_s) \cdot \alpha_M \alpha_s}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot (\alpha_M + \alpha_s) \cdot A^{\frac{d+1}{2}}} \leq \Theta\left(d^{-\frac{1}{2}}\right). \tag{21}$$

If $\alpha_M > \alpha_s\sqrt{2}$, *then*

$$\frac{W(\alpha_s, \alpha_M, \alpha_s)}{\mathrm{Vol}_{\mathbb{E}^d}(B)\, \alpha_s^d} = \Theta(1). \tag{22}$$

Proof. We apply Lemma 7. First, consider the case $\alpha_M \leq \alpha_s$. Then, the case 2 applies to $W(\alpha_M, \alpha_s, \alpha_s)$. From (10) we obtain

$$\Theta\left(d^{-1}\right) \leq \frac{W(\alpha_s, \alpha_M, \alpha_s) \cdot \alpha_M \alpha_s}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot (\alpha_M + \alpha_s) \cdot A^{\frac{d+1}{2}}} \leq \Theta\left(d^{-\frac{1}{2}}\right). \tag{23}$$

Now, consider the case $\alpha_s < \alpha_M$.

1. If $\alpha_s < \alpha_M \leq \alpha_s\sqrt{2}$, then from (10) we obtain
$$\Theta\left(d^{-1}\right) \leq \frac{W(\alpha_s, \alpha_M, \alpha_s) \cdot \alpha_M \alpha_s}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot (\alpha_M + \alpha_s) \cdot A^{\frac{d+1}{2}}} \leq \Theta\left(d^{-\frac{1}{2}}\right).$$
2. If $\alpha_s\sqrt{2} < \alpha_M < 2\alpha_s$, then (11) yields
$$W(\alpha_s, \alpha_M, \alpha_s) = \Theta\left(\mathrm{Vol}_{\mathbb{E}^d}(B) \cdot \alpha_s^d\right).$$
3. If $2\alpha_s \leq \alpha_M$, then (13) gives
$$W(\alpha_s, \alpha_M, \alpha_s) = \mathrm{Vol}_{\mathbb{E}^d}(B) \cdot \alpha_s^d\,(1 - o(1)).$$

This concludes the proof of the lemma. □

Lemma 10. *Denote by* $A = \delta^2 \cdot \frac{(M+r+s)(M+r-s)(s+M-r)(s+r-M)}{4s^2}$. *If* $\alpha_s - \alpha_r < \alpha_M < \sqrt{\alpha_s^2 + \alpha_r^2}$, *then*

$$\Theta\left(d^{-1}\right) \leq \frac{W(\alpha_r, \alpha_M, \alpha_s) \cdot \delta}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot A^{\frac{d+1}{2}}} \leq \Theta\left(d^{-\frac{1}{2}}\right). \tag{24}$$

If $\alpha_M > \sqrt{\alpha_s^2 + \alpha_r^2}$*, then*

$$\frac{\mathrm{Vol}_{\mathbb{H}^d}(B(\alpha_r)) - W(\alpha_r, \alpha_M, \alpha_s)}{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot A^{\frac{d+1}{2}} \cdot \delta^{-1}} \leq \Theta\left(d^{-\frac{1}{2}}\right). \tag{25}$$

Proof. The result follows directly from Lemma 7. □

Lemma 11. *If* $M > s\sqrt{2}$*, then*

$$W(\alpha_s, \alpha_M, \alpha_s) \geq \frac{V_d(R)}{n} \cdot s^{d+o(d)}.$$

If $M \leq s\sqrt{2}$*, then*

$$W(\alpha_s, \alpha_M, \alpha_s) \geq \frac{V_d(R)}{n} \cdot \left(M^2 - \frac{M^4}{4s^2}\right)^{\frac{d}{2}+o(d)}.$$

Proof. If $M > s\sqrt{2}$, then we can apply (22) to obtain

$$\begin{aligned} W(\alpha_s, \alpha_M, \alpha_s) &\geq \Theta\left(d^{-1} \cdot (s\delta)^d \cdot \mathrm{Vol}_{\mathbb{E}^d}(B)\right) = \Theta\left(d^{-1} \cdot s^d \cdot \frac{dv_d(R)}{n} \cdot \mathrm{Vol}_{\mathbb{E}^d}(B)\right) \\ &= \Theta\left(s^d \cdot \frac{V_d(R)}{n} \cdot \frac{\mathrm{Vol}_{\mathbb{E}^d}(B)}{\mathrm{Vol}(\mathbb{S}^{d-1})}\right) = \frac{V_d(R)}{n} s^{d+o(d)}. \end{aligned}$$

If $M \leq s\sqrt{2}$ then we can apply (21) to obtain

$$\begin{aligned} W(\alpha_s, \alpha_M, \alpha_s) &\geq \mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot \delta^{-1} A^{\frac{d+1}{2}} \Theta(d^{-1}) \\ &= \mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot \delta^{-1} \cdot \alpha_M^{d+1} \cdot \left(1 - \frac{\alpha_M^2}{4\alpha_s^2}\right)^{\frac{d+1}{2}} \cdot \Theta(d^{-1}) \\ &= \mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot \delta^d \cdot M^{d+1} \left(1 - \frac{M^2}{4s^2}\right)^{\frac{d+1}{2}} \cdot \Theta(d^{-1}) \\ &= \frac{\mathrm{Vol}_{\mathbb{E}^{d-1}}(B)}{\mathrm{Vol}(\mathbb{S}^{d-1})} \cdot \frac{V_d(R)}{n} \cdot M^{d+1} \left(1 - \frac{M^2}{4s^2}\right)^{\frac{d+1}{2}} \cdot \Theta(1) \\ &= \Theta\left(d^{-1}\right) \cdot \frac{V_d(R)}{n} \cdot \left(M^2 - \frac{M^4}{4s^2}\right)^{\frac{d+1}{2}} = \frac{V_d(R)}{n} \cdot \left(M^2 - \frac{M^4}{4s^2}\right)^{\frac{d}{2}+o(d)}. \end{aligned}$$

□

Corollary 4. *If* $s > 1$ *and either* $M > s\sqrt{2}$ *or* $M^2 - M^4/4s^2 > 1$*, then there exists a constant* $S > 1$ *such that* $W(\alpha_s, \alpha_M, \alpha_s) \geq \frac{V_d(R)}{n} S^{d+o(d)}$.

Lemma 12. *Assume that* $s > 1$*. If*

$$M > s\sqrt{2} \text{ or } M^2 - \frac{M^4}{4s^2} > 1, \tag{26}$$

then there exist constants $S > 1$ *and* $\varepsilon > 0$*, such that*

$$W\left(\alpha_M, \alpha_s, \alpha_s + \varepsilon\delta\right) \geq \frac{V_d(R)}{n} S^{d+o(d)}.$$

Proof. If $M > s\sqrt{2}$, then we can choose a sufficiently small $\varepsilon > 0$ such that $M^2 \geq s^2 + (s+\varepsilon)^2$. Then, the result follows from Lemma 11 and the fact that $s > 1$. Otherwise $M^2 < s^2 + (s+\varepsilon)^2$, and instead of the condition $M^2 - \frac{M^4}{4s^2} > 1$ from (24) we have

$$\frac{4s^2(s+\varepsilon)^2 - (M^2 - s^2 - (s+\varepsilon)^2)^2}{4(s+\varepsilon)^2} > 1.$$

As it holds for $\varepsilon = 0$, we can choose a small enough $\varepsilon > 0$ that it is still satisfied. □

Lemma 13. *The dependence of consecutive steps is negligible.*

Proof. First, consider the case $M \leq s\sqrt{2}$. It suffices to show that $W_1 = W(\alpha_M, \alpha_s, \alpha_s + 2\varepsilon\delta)$ is negligible in comparison with $W_1 + W_2 = W(\alpha_M, \alpha_s, \alpha_s + \varepsilon\delta)$ and the same for $W_1 = W(\alpha_s, \alpha_M, \alpha_s + 2\varepsilon\delta)$ compared to $W_1 + W_2 = W(\alpha_s, \alpha_M, \alpha_s + \varepsilon\delta)$. From (24) we can write their ratio

$$\frac{W_1}{W_1 + W_2} = \left(\frac{4(s+\varepsilon)^2}{2\left(M^2 s^2 + s^2(s+\varepsilon)^2 + M^2(s+\varepsilon)^2\right) - \left(M^4 + s^4 + (s+\varepsilon)^4\right)} \right.$$
$$\left. \cdot \frac{2\left(M^2 s^2 + s^2(s+2\varepsilon)^2 + M^2(s+2\varepsilon)^2\right) - \left(M^4 + s^4 + (s+2\varepsilon)^4\right)}{4(s+2\varepsilon)^2} \right)^{\frac{d+1}{2}} = o(1).$$

Since at each step we reduce the radius of the ball centered at $\mathbf{q}$ which contains the current position by at least $\varepsilon\delta$, any ball encountered in the current iteration intersects only a constant number of other balls, so their overall effect is negligible.

In the case when $M > s\sqrt{2}$ and $s > 1$ with probability $1 - o(1)$ we find the nearest neighbor in one step. □

Lemma 14. *If for some constants M, s, r we have $M^2 > s^2 + r^2$, then*

$$\mathrm{Vol}_{\mathbb{H}^d}\left(B(\alpha_r)\right) - W\left(\alpha_r, \alpha_M, \alpha_s\right) \leq \mathrm{Vol}_{\mathbb{H}^d}\left(B(\alpha_r)\right) \cdot \beta^d$$

with some $\beta < 1$.

Proof. It follows from (25) that

$$\mathrm{Vol}_{\mathbb{H}^d}\left(B(\alpha_r)\right) - W\left(\alpha_r, \alpha_M, \alpha_s\right) \leq \Theta\left(d^{-\frac{1}{2}}\right) \cdot \mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot A^{\frac{d+1}{2}} \cdot \delta^{-1}.$$

It remains to bound from above the following ratio:

$$\frac{\Theta\left(d^{-\frac{1}{2}}\right) \cdot \mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot A^{\frac{d+1}{2}} \cdot \delta^{-1}}{\mathrm{Vol}_{\mathbb{H}^d}\left(B(\alpha_r)\right)} = \frac{\Theta\left(d^{-\frac{1}{2}}\right) \cdot \mathrm{Vol}_{\mathbb{E}^{d-1}}(B) \cdot A^{\frac{d+1}{2}}}{\mathrm{Vol}_{\mathbb{E}^d}(B) \cdot r^d \cdot \delta^{d+1}} = \Theta(1) \cdot \left(\frac{A}{\delta^2 r^2}\right)^{\frac{d+1}{2}}$$
$$= \Theta(1) \cdot \left(\frac{(M+r+s)(M+r-s)(s+M-r)(s+r-M)}{4r^2 s^2}\right)^{\frac{d+1}{2}} =$$
$$\Theta(1) \cdot \left(1 - \frac{(M^2 - s^2 - r^2)^2}{4s^2 r^2}\right)^{\frac{d+1}{2}} \leq \beta^{d+1}$$

for some $\beta < 1$. □

C.2 Analysis of the boundary effect

Lemma 15. *The boundary effect is negligible and does not affect our results.*

Proof. Consider the worst-case scenario when the current element is located on the boundary of the ball $\mathcal{B}$ (see Figure 6). Indeed, in this case, the largest part of the neighborhood is located outside $\mathcal{B}$, which may reduce the number of neighbors of a given element.

First, consider the Euclidean space. According to Lemma 5, this is equivalent to the hyperbolic case with $R \ll 1/\sqrt{d}$. Let us show that $W(\alpha, R, R)$ converges to $\frac{1}{2}\mathrm{Vol}_{\mathbb{E}^d} B(\alpha)$ if $\alpha = \Theta(\delta)$ with $\delta = \left(\frac{d\mathrm{Vol}_{\mathbb{E}^d}(B(R))}{\mathrm{Vol}(\mathbb{S}^{d-1})n}\right)^{1/d}$. Note that $W(\alpha, R, R) = C_\alpha(h) + C_R(\alpha - h) < \frac{1}{2}\mathrm{Vol}_{\mathbb{E}^d} B(\alpha)$. It is easy to see that $h = \alpha - \frac{\alpha^2}{2R}$. Then,

$$C_\alpha(h) = \frac{1}{2}\,\mathrm{Vol}_{\mathbb{E}^{d-1}}(B)\alpha^d \mathrm{B}\left(\frac{d+1}{2}, \frac{1}{2}; 1 - \frac{\alpha^2}{4R^2}\right) \to$$
$$\frac{1}{2}\,\mathrm{Vol}_{\mathbb{E}^{d-1}}(B)\alpha^d \mathrm{B}\left(\frac{d+1}{2}, \frac{1}{2}\right) = \frac{1}{2}\,\mathrm{Vol}_{\mathbb{E}^d} B(\alpha),$$

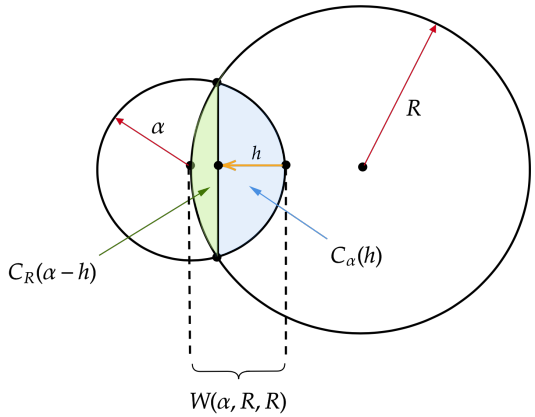


Figure 6: Boundary effect is negligible

if $\alpha \ll 1/\sqrt{d}$. Indeed,

$$\mathrm{B}\left(\frac{d+1}{2}, \frac{1}{2}\right) - \mathrm{B}\left(\frac{d+1}{2}, \frac{1}{2}; 1 - \frac{\alpha^2}{4R^2}\right) = \int_0^{\frac{\alpha^2}{4R^2}} t^{(d-1)/2}(1-t)^{-1/2} dt$$
$$\leq \frac{1}{\sqrt{1 - \frac{\alpha^2}{4R^2}}} \int_0^{\frac{\alpha^2}{4R^2}} t^{(d-1)/2} dt = \frac{2\left(\frac{\alpha}{2R}\right)^{d+1}}{(d+1)\sqrt{1 - \frac{\alpha^2}{4R^2}}} \to 0$$

if $(\alpha/R)^d \to 0$, which is the case since

$$\left(\frac{\alpha}{R}\right)^d = \frac{\Theta(1)^d d \mathrm{Vol}_{\mathbb{E}^d}(B(R))}{R^d \mathrm{Vol}(\mathbb{S}^{d-1}) n} = \frac{\Theta(1)^d}{n} \to 0$$

for $d \ll \log n$.

In the hyperbolic case, the "small" ball of radius α is small enough ($\alpha \ll 1/\sqrt{d}$) so that its geometry can be considered Euclidean, and thus we can apply Lemma 5 to estimate the volume of $C_\alpha(h)$. The "large" hyperbolic ball of radius R in this case can be though of as a Euclidean ball of radius $\sim \frac{1}{2}e^{2R}$ in the upper half–space model, and then we can carry out the same analysis on the Euclidean scale of the "small" radius α ball. Note that in this case

$$\left(\frac{2\alpha}{e^{2R}}\right)^d = \frac{\Theta(1)^d d v_d(R)}{e^{2Rd} n} \leq \frac{\Theta(1)^d e^{R(d-1)}}{e^{2Rd} n} \to 0\,.$$

□

C.3 Time and space complexity

In this section, we estimate time and space complexity. Recall that $N(v)$ denotes the number of neighbors of an element v in $G(M)$ and $f = (n-1)\mathrm{Vol}_{\mathbb{H}^d}\left(B(\alpha_M)\right)/\mathrm{Vol}_{\mathbb{H}^d}\left(\mathcal{B}\right)$. The proofs below mostly follow the corresponding proofs from Prokhorenkova & Shekhovtsov (2020). We keep the proofs here for completeness.

Lemma 16. *The probability of the event "$N(v)$ satisfies $N(v) \leq \frac{3}{2}f$" is at least $1 - \frac{4}{f}$.*

Proof. Note that the random variable $N(v)$ is upper bounded by the random variable $N'(v)$ having the binomial distribution $\mathrm{Bin}\left(n-1, \frac{\mathrm{Vol}_{\mathbb{H}^d}(B(\alpha_M))}{\mathrm{Vol}_{\mathbb{H}^d}(\mathcal{B})}\right)$ with the expected value f. Thus, via the Chebyshev's inequality, we obtain

$$\mathsf{P}\left(N(v) > \frac{3}{2}f\right) \leq \mathsf{P}\left(N'(v) > \frac{3}{2}f\right) \leq \mathsf{P}\left(|N'(v) - f| > \frac{f}{2}\right) \leq \frac{4\,\mathsf{Var}\left(N'(v)\right)}{f^2} \leq \frac{4}{f}.$$

□

Lemma 17. *The time complexity of k steps of the graph-based NNS is $O(kfd)$ with probability $1 - O\left(\frac{1}{kf}\right)$.*

Proof. We can upper bound the number of distance computations by the random variable distributed according to $\mathrm{Bin}\left(k(n-1), \frac{\mathrm{Vol}_{\mathbb{H}^d}(B(\alpha_M))}{\mathrm{Vol}_{\mathbb{H}^d}(\mathcal{B})}\right)$ and proceed as in the proof of Lemma 16. Finally, note that one distance computation takes $\Theta(d)$. □

Lemma 18. *The probability of the event "$E(G)$ satisfies $E(G) \leq \frac{3}{4}fn$" is $1 - O\left(\frac{1}{fn}\right)$.*

Proof. We mostly follow the proof of Prokhorenkova & Shekhovtsov (2020), but some modifications are needed to handle the boundary. We use Chebyshev's inequality, so we have to estimate the variance $\mathsf{Var}E(G)$:

$$\begin{aligned}\mathsf{Var}E(G) &= \mathbb{E}\left(E(G)^2\right) - (\mathbb{E}E(G))^2 \\ &= \sum_{e_1, e_2 \in \binom{\mathcal{D}}{2}} \mathsf{P}(e_1, e_2 \in E(G)) - \mathsf{P}(e_1 \in E(G)) \cdot \mathsf{P}(e_2 \in E(G)).\end{aligned}$$

Obviously, if e_1 and e_2 do not share common endpoints, then $\mathsf{P}(e_1, e_2 \in E(G)) = \mathsf{P}(e_1 \in E(G)) \cdot \mathsf{P}(e_2 \in E(G))$. Also, one can see that if e_1 and e_2 share one element, then we also have $\mathsf{P}(e_1, e_2 \in E(G)) = \mathsf{P}(e_1 \in E(G)) \cdot \mathsf{P}(e_2 \in E(G))$. Indeed, assume that both e_1 and e_2 contain a common element v. Then the equality holds conditioned on the position of v since both sides of the equation are defined in terms of v-th neighborhood size.

Thus, only the terms with $e_1 = e_2$ remain in the expression:

$$\mathsf{Var}E(G) = \sum_{e_1 \in \binom{\mathcal{D}}{2}} \mathsf{P}(e_1 \in E(G)) - (\mathsf{P}(e_1 \in E(G)))^2 \leq \mathbb{E}E(G).$$

Finally, we get

$$\mathsf{P}\left(|E(G) - \mathbb{E}E(G)| > \frac{\mathbb{E}E(G)}{2}\right) \leq \frac{4\mathsf{Var}E(G)}{(\mathbb{E}E(G))^2} \leq \frac{4}{\mathbb{E}E(G)}.$$

It remains to note that $\mathbb{E}E(G) \leq nf/2$ and due to Lemma 15 we also have $\mathbb{E}E(G) = \Theta(nf)$. From this the lemma follows.

□

D k-NN GRAPHS

Note that in Theorem 1 we consider undirected graphs since the condition on $\rho(\cdot, \cdot)$ is symmetric. In this section, we extend the theorem to nearest neighbor graphs. In this case, we assume that each node is connected to a fixed number of its nearest neighbors by directed edges. During graph traversal, outgoing edges are used to explore the neighborhood.

Theorem 3. *Assume that $d = d(n)$ is such that $\log\log n \ll d(n) \ll \log n$ and $R = R(n)$ is such that $e^R \ll n^{\frac{1}{d-1}} \cdot d^{-1/2}$.*

For the exact NNS, let M be a constant s.t. $M > \sqrt{2}$ and let $k = [M^d]$. Then, with probability $1 - o(1)$, NNS based on the k-NN graph finds the exact nearest neighbor. For c, r-ANN with some constant $c > 1$, let M be a constant s.t. $M > \sqrt{\frac{4c^2}{3c^2-1}}$ and $k = [M^d]$. Then, with probability $1 - o(1)$, NNS based on the k-NN graph solves c, r-ANN for any r.

In both cases, the time complexity is $O\left(k \cdot n^{1/d} \cdot R \cdot (v_d(R))^{-1/d}\right)$;

the space complexity is $O(n \cdot \log n \cdot k)$.

Proof. First, note that the space complexity is straightforward: each node has exactly k outgoing edges. Similarly, the complexity of one step is $O(k \cdot d)$.

Thus, it remains to show that kNN-based NNS succeeds with probability $1 - o(1)$ and that the number of steps can be upper bounded similarly to Theorem 1. To prove this, it is sufficient to show that the distance to the k-th neighbor is sufficiently large, so that the convergence can be guaranteed by the proof of Theorem 1.

For every node v define a random variable $\rho_k(v)$ to be the distance from v to its k-th nearest element. It is sufficient to show that $\rho_k(v) \geq \widehat{M}\delta$, where $\delta = \left(\frac{dv_d(R)}{n}\right)^{1/d}$ and $\widehat{M}$ satisfies the required conditions on M in Theorem 3. For some $\varepsilon > 0$, let $\widehat{M} := M(1-\varepsilon)$. Obviously, by choosing small enough ε, we can guarantee that the required conditions are satisfied.

Note that $N_r(v)$ — the number of elements at distance at most r from v — can be upper bounded by a random variable having the binomial distribution $\mathrm{Bin}\left(n-1, \frac{\mathrm{Vol}_{\mathbb{H}^d}(B(r))}{\mathrm{Vol}_{\mathbb{H}^d}(\mathcal{B})}\right)$ (the upper bound is caused by possible boundary effects). If $r = \widehat{M}\delta$, then $\frac{\mathrm{Vol}_{\mathbb{H}^d}(B(r))}{\mathrm{Vol}_{\mathbb{H}^d}(\mathcal{B})} = \widehat{M}^d/n$. Thus, $N_{\widehat{M}\delta}(v)$ is upper bounded by $\mathrm{Bin}\left(n, \widehat{M}^d/n\right)$.

Recall that $k = [M^d]$. It is sufficient to show that with high probability we have $N_{\widehat{M}\delta}(v) < k$ for all v. Using the union bound over all possible v and the Chernoff's bound, we obtain the following:

$$\mathsf{P}\left(\exists\, v : N_{\widehat{M}\delta}(v) \geq k\right) \leq n \cdot e^{-\frac{\left(k-\widehat{M}^d\right)^2}{2(\widehat{M}^d + (k-\widehat{M}^d)/3)}} \leq e^{\log n - \frac{M^{2d}(1-o(1))}{\Theta(M^d)}} \to 0.$$

as $n \to \infty$, since $d \gg \log\log n$.

□

E GRAPH-BASED NNS IN SPARSE REGIME

In this section, we prove Theorem 2: we show that the elements are tightly concentrated near the boundary of the ball and thus the convergence guarantees resemble those on the sphere.

Let us denote by $\varphi(\mathbf{x}_i, \mathbf{x}_j)$ the angle $\angle \mathbf{x}_i O \mathbf{x}_j$. For the sphere, graph $G(M)$ is constructed by connecting such pairs of elements that (Prokhorenkova & Shekhovtsov, 2020):

$$\cos\varphi(\mathbf{x}_i, \mathbf{x}_j) \geq \sqrt{\frac{2M\log n}{d}}\,. \tag{27}$$

For the hyperbolic space, we connect the elements $\mathbf{x}_i$ and $\mathbf{x}_j$ if

$$\cosh\rho(\mathbf{x}_i, \mathbf{x}_j) \leq \cosh^2 R - \sinh^2 R \cdot \sqrt{\frac{2M\log n}{d}}. \tag{28}$$

Note that this corresponds to the elements $\mathbf{x}_i$ and $\mathbf{x}_j$ located on the boundary of $\mathcal{B}$ with $\varphi(\mathbf{x}_i, \mathbf{x}_j)$ as in (27).

Recall that for $c > 0$ we set

$$\alpha_c := \frac{\cosh^2 R - \cosh(\operatorname{arccosh}(\cosh^2 R)/c)}{\sinh^2 R}\,. \tag{29}$$

Let us explain this choice. Below we prove that NNS in the hyperbolic space behaves similarly to those on a sphere. In other words, we can project all the elements to the boundary of $\mathcal{B}$ and this will not significantly affect the results. However, we have to properly adjust α_c taking into account the multiplicative approximation factor c. In Prokhorenkova & Shekhovtsov (2020), α_c is the cosine of the angle $\frac{\pi}{2c}$. There, $\frac{\pi}{2c}$ is the spherical distance that is c times smaller than the distance between the elements with $\varphi(\mathbf{x}_i, \mathbf{x}_j) = \pi/2$. The hyperbolic distance D between the elements with $\varphi(\mathbf{x}_i, \mathbf{x}_j) = \pi/2$ located on the boundary of $\mathcal{B}$ satisfies the following equality:

$$\cosh D = \cosh^2 R, \text{ i.e., } D = \operatorname{arccosh}(\cosh^2 R)\,.$$

Then, we need to compute $\cos\varphi(\mathbf{x}_i, \mathbf{x}_j)$ if $\rho(\mathbf{x}_i, \mathbf{x}_j) = D/c$. This follows from the hyperbolic law of cosines:

$$\cos\varphi(\mathbf{x}_i, \mathbf{x}_j) = \frac{\cosh^2(R) - \cosh(D/c)}{\sinh^2(R)},$$

which is equal to α_c in Equation (29).

Thus, it remains to show that the convergence guarantees transfer from the spherical space to the hyperbolic case. Let us first prove that the elements are tightly concentrated near the boundary of the ball $\mathcal{B}$.

Lemma 19. *With probability $1 - o(1)$ all n elements are located within a shell of width $\Theta\left(\frac{\log n}{d}\right)$ near the boundary of the ball $\mathcal{B}$.*

Proof. According to (3), we have

$$\frac{\mathrm{Vol}_{\mathbb{H}^d}(B(R-\varepsilon))}{\mathrm{Vol}_{\mathbb{H}^d}(B(R))} = \Theta\left(e^{-\varepsilon(d-1)}\right).$$

Let $\varepsilon = \frac{1}{d-1}(\log n + \varphi)$, where $\varphi = \varphi(n)$ is any function s.t. $\varphi \to \infty$ as $n \to \infty$. Then,

$$\frac{\mathrm{Vol}_{\mathbb{H}^d}(B(R-\varepsilon))}{\mathrm{Vol}_{\mathbb{H}^d}(B(R))} = \Theta\left(e^{-\log n - \varphi}\right) = o(1/n).$$

From this the lemma follows. □

Lemma 20. *If two elements $\boldsymbol{x}_i, \boldsymbol{x}_j$ satisfy* (27) *with constant $M + \varepsilon$ for some $\varepsilon > 0$, then with probability $1 - o(1)$ they satisfy* (28) *with constant M.*

Proof. Let the distances from $\mathbf{x}_i$ and $\mathbf{x}_j$ to the center of the ball $\mathcal{B}$ be r_i and r_j, respectively. From Lemma 19, r_i and r_j are equal to $R - \Theta\left(\frac{\log n}{d}\right)$. Using the hyperbolic law of cosines, we get

$$\begin{aligned}
\cosh\rho(\mathbf{x}_i, \mathbf{x}_j) &= \cosh r_i \cosh r_j - \sinh r_i \sinh r_j \cos\varphi(\mathbf{x}_i, \mathbf{x}_j) \\
&\le \cosh r_i \cosh r_j - \sinh r_i \sinh r_j \sqrt{\frac{2M\log n}{d}} \\
&= \cosh^2(R - \Theta(\log n/d)) - \sinh^2(R - \Theta(\log n/d))\sqrt{\frac{2M\log n}{d}}.
\end{aligned}$$

It is sufficient to show that

$$\begin{aligned}
&\cosh^2(R - \Theta(\log n/d)) - \sinh^2(R - \Theta(\log n/d))\sqrt{\frac{2M\log n}{d}} \\
&\qquad \ge \cosh^2 R - \sinh^2 R \cdot \sqrt{\frac{2(M+\varepsilon)\log n}{d}}.
\end{aligned}$$

Note that $\cosh(R - \Theta(\log n/d)) = \frac{e^R}{2}(1 + \Theta(\log n/d))$. The same holds for $\sinh(R - \Theta(\log n/d))$. Therefore, we need

$$\frac{e^{2R}}{4}(1 + \Theta(\log n/d))\left(1 - \sqrt{\frac{2M\log n}{d}}\right) \ge \frac{e^{2R}}{4}\left(1 + \Theta\left(e^{-2R}\right)\right)\left(1 - \sqrt{\frac{2(M+\varepsilon)\log n}{d}}\right).$$

Now we divide the left hand side by the right hand side and obtain

$$\begin{aligned}
&\frac{\left(1 - \sqrt{\frac{2M\log n}{d}}\right)}{\left(1 - \sqrt{\frac{2(M+\varepsilon)\log n}{d}}\right)}\left(1 + \Theta(\log n/d) + \Theta\left(e^{-2R}\right)\right) \\
&= \left(1 + \left(\sqrt{\frac{2(M+\varepsilon)\log n}{d}} - \sqrt{\frac{2M\log n}{d}}\right)\left(1 + \Theta\left(\sqrt{\log n/d}\right)\right)\right)\left(1 + \Theta(\log n/d) + \Theta\left(e^{-2R}\right)\right) \\
&\qquad = \left(1 + \Omega(\sqrt{\log n/d})\right)\left(1 + \Theta(\log n/d) + \Theta\left(e^{-2R}\right)\right).
\end{aligned}$$

From this the lemma follows, since $\sqrt{\log n/d} \gg \log n/d$ and $\sqrt{\log n/d} \gg e^{-2R}$. □

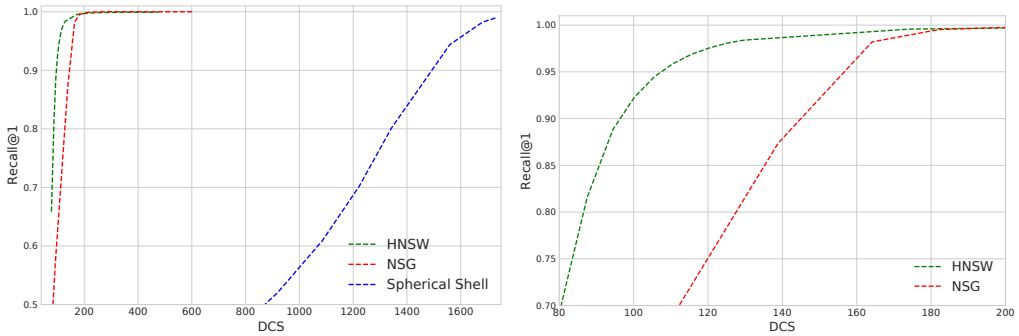


Figure 7: Comparison of algorithms on the recommendation dataset with $d = 10$

Let us choose such small $\varepsilon > 0$ that the required condition on M is also satisfied for $M + \varepsilon$. It follows from Lemma 20 that our graph $G(M)$ contains the spherical nearest neighbor graph defined via (27) with constant $M + \varepsilon$. Thus, we may follow the proof of Theorem 2 in (Prokhorenkova & Shekhovtsov, 2020) (Appendix B.5) with the corresponding nearest neighbor graph.

Note that all spherical caps and distances mentioned in the proof of (Prokhorenkova & Shekhovtsov, 2020) are defined by the condition

$$\cos\varphi \geq \sqrt{\frac{2S\log n}{d}}$$

According to Lemma 20, such constants are negligibly affected by deviations from the boundary, and thus we can directly follow the proof of Prokhorenkova & Shekhovtsov (2020) to obtain the convergence guarantees and complexity.

F Experiments

Implementation Our implementation supplements the submission.

Best-first search In the main text, for the synthetic experiments, we use the greedy search. However, in graph-based NNS algorithms, the greedy search is often replaced by the best-first search (a.k.a. beam search). In this case, instead of keeping track of only the best candidate, we maintain a dynamic list of several candidates. In particular, this technique is used in HNSW (Malkov & Yashunin, 2018), and the size of the candidate list is controlled by the parameter $efSearch$.

Hyperparameters Recall that to plot Figure 3, we vary the graph degree to control the complexity and accuracy of NNS. In Figures 1 and 2, we use the best-first search, so we vary the parameter $efSearch$ and fix the graph degree. In this case, we tune M and $efConstruction$ HNSW hyperparameters, which are responsible for the graph degree and quality of the graph construction, respectively. For both GloVe and Poincaré GloVe databases, we get M=10 and $efConstruction$=300. For other datasets, we get M=8. For NSG, the degree is controlled by the parameter R and the optimal degree is $R = 2M$.

Spherical Shell implementation As a baseline, we implement Spherical Shell proposed by Wu & Charikar (2020). As in the original paper, we split the elements into 25 bands such that $w^{i-1} \leq \frac{1}{1-\|x\|^2} \leq w^i$. To achieve this, we take $w = 1.05$. For each band, we construct LSH with 50 tables. Each table has $[\log(m)] - 1$ bins, where m is the number of elements in the band. The number of probes is tuned for each band to reach 0.95 recall (this value has been chosen to achieve better results). When searching for the nearest neighbor for a query q, we compute the nearest neighbors within l bands closest to q and take the best one according to the hyperbolic distance. The number of bands l is varied from 1 to 10, thus giving the curve on Figure 1.

Additional experiments on real datasets Figures 9 and 8 compare the algorithms on the recommendation dataset with $d = 10$ from (Shevkunov & Prokhorenkova, 2021) and the Poincaré GloVe

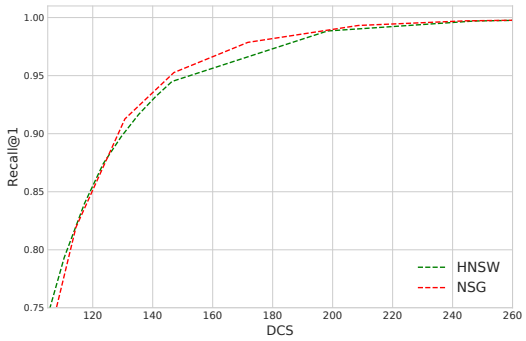


Figure 8: Comparison of graph-based algorithms on Poincaré GloVe with $d = 10$

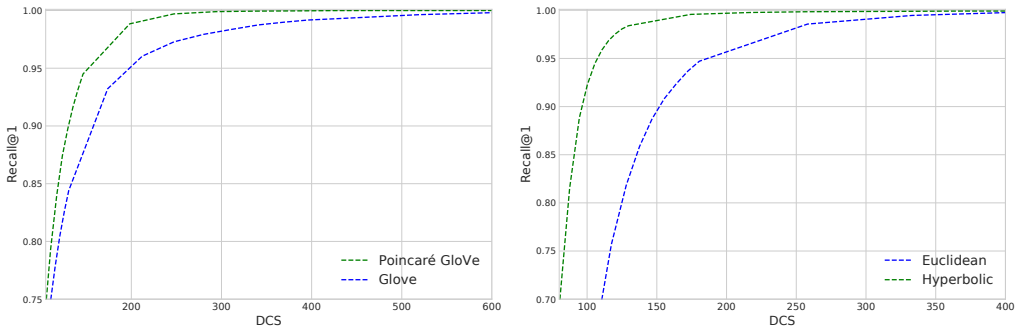


Figure 9: Efficiency of HNSW in hyperbolic space vs Euclidean space on Poincaré GloVe with $d = 10$ (left) and recommendations (right)

embedding trained for $d = 10$. We can see that HNSW and NSG have similar performance on Poincaré GloVe, while HNSW outperforms NSG on the recommendation dataset. Spherical Shell needs a significantly larger number of distance calculations on the recommendation dataset. This can be explained by the fact that this dataset significantly differs from the Poincaré GloVe: the learned vector norms are much larger. Because of this, we either have annuli of large width (increased approximation error) or many annuli (increased total complexity).

Then, we compare efficiency of NNS in Euclidean and hyperbolic spaces. Figure 9 shows that hyperbolic NNS is more efficient.

Additional synthetic experiments Figure 10 is similar to Figure 3, but now we use the best-first search instead of the greedy search. We see that the conclusions are the same.

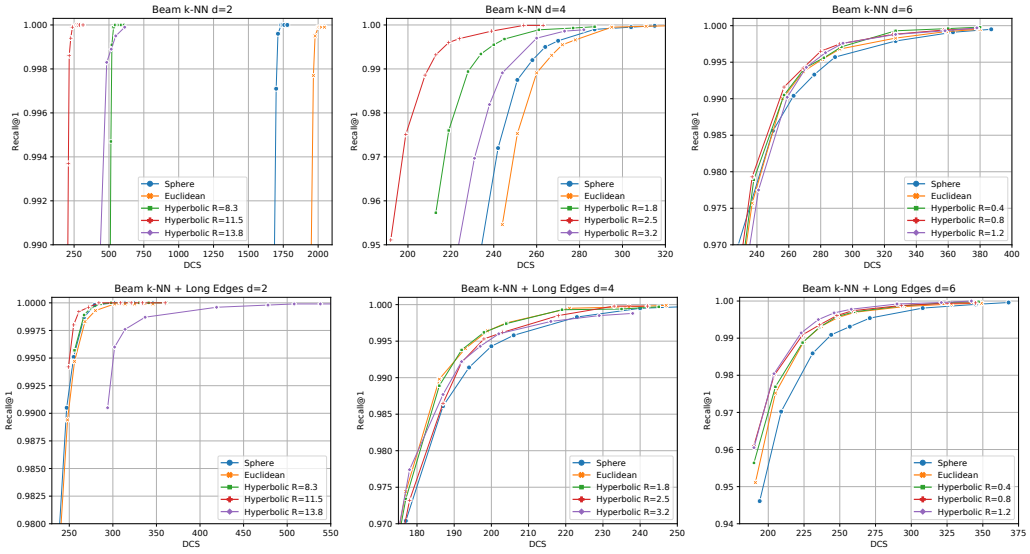


Figure 10: Best-first search over NN graphs without (top row) and with (bottom row) long edges