HLoOP - Hyperbolic 2-space Local Outlier Probabilities

Clémence Allietta, Jean-Philippe Condomines, Jean-Yves Tourneret, Emmanuel Lochin

HLoOP – Hyperbolic 2-space Local Outlier Probabilities

Clémence Allietta[1] Jean-Philippe Condomines[1] Jean-Yves Tourneret[2] Emmanuel Lochin[1]

ENAC[1], IRIT[2], Université de Toulouse, France

{name.surname}@enac.fr[1], {jean-yves.tourneret}@irit.fr[2]

Abstract

Hyperbolic geometry has recently garnered considerable attention in machine learning due to its capacity to embed hierarchical graph structures with low distortions for further downstream processing. This paper introduces a simple framework to detect local outliers for datasets grounded in hyperbolic 2-space referred to as HLoOP (Hyperbolic Local Outlier Probability). Within a Euclidean space, well-known techniques for local outlier detection are based on the Local Outlier Factor (LOF) and its variant, the LoOP (Local Outlier Probability), which incorporates probabilistic concepts to model the outlier level of a data vector. The developed HLoOP combines the idea of finding nearest neighbors, density-based outlier scoring with a probabilistic, statistically oriented approach. Therefore, the method consists in computing the Riemmanian distance of a data point to its nearest neighbors following a Gaussian probability density function expressed in a hyperbolic space. This is achieved by defining a Gaussian cumulative distribution in this space. The HLoOP algorithm is tested on the WordNet dataset yielding promising results. Code and data will be made available on request for reproductibility.

1 Introduction and Prior Work

From social interaction analysis in social sciences to sensor networks in communication, machine learning has gained in importance in the last few years for analyzing large and complex datasets. Applying machine learning algorithms in an Euclidean space is efficient when data have an underlying Euclidean structure. However, in many applications such as computer graphics or computer vision, data cannot be embedded in a Euclidean space, which prevents the use of conventional algorithms [1]. As an example, in datasets having a hierarchical structure, the number of relevant features can grow exponentially with the depth of the hierarchy and thus these features cannot be embedded without distortions in an Euclidean space. In the quest for a more appropriate geometry of hierarchies, hyperbolic spaces and their models (Poincaré disk or upper-half plane conformal models, Klein non-conformal model, Beltrami hemisphere model and Lorentz hyperboloid model among others [2]) provide attractive properties that can lead to substantial performance and efficiency benefits for learning representations of hierarchical and graph data. Among several potential advantages, we can highlight [3] 1) a better generalization capability of the model, with less overfitting, computational complexity, and requirement of training data; 2) a reduction in the number of model parameters and embedding dimensions; 3) a better model understanding and interpretation. Empowered by these geometric properties, hierarchical embeddings have recently been investigated [4] for complex trees with low distortions [5–7]. This has led to rapid advances in machine learning and data science across many disciplines and research areas, including but not limited to graph networks [8–11], computer vision [12–16], network topology analysis [17–20], quantum science [21, 22]. Finally, it is interesting to mention the recent boom in hyperbolic neural networks and hyperbolic computer vision, which is for instance reported in the recent reviews [23, 3].

Preprint. Under review.

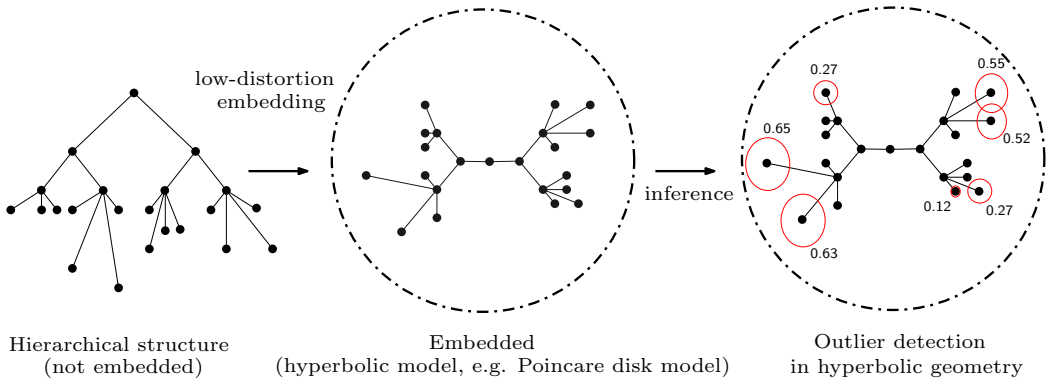


Figure 1: Illustration inspired from [24] of local outlier probabilities in a hyperbolic model: A hierarchical structure (left) is embedded in a hyperbolic space with low-distortion (middle). The point dataset is then described by *local* outlier probabilities in this hyperbolic space (right).

Motivated by these recent advances, identifying and dealing with outliers is crucial for generating trustworthy insights and making data-driven decisions in hyperbolic spaces, e.g., providing information about which nodes are highly connected (and hence more central) or which nodes correspond to outliers such that embedding methods can realistically be used to model real complex patterns. In this study, we focus on *local* outlier detection, which describes *local* properties of data, which is relevant in many applications involving Euclidean spaces. An overview of *local* anomaly detection methods can be found in the literature from many surveys or books ([25–28]). Initially, research related to local outlier detection was focused on intrusion detection, fraud detection[29] and medical applications[30]. Intrusion detection [31–33] consists of detecting abnormal traffic in networks due to suspicious data or violations of network management policies. Fraud detection[34–36] detects unexpected activities in banking or insurance data, such as fraudulent online payments by credit card or inconsistent insurance claims. In the wake of other disciplines, local outlier detection algorithms have been used for medical data [37, 30], e.g., to detect abnormal QRS complexes in electrocardiograms due to certain diseases (such as premature ventricular contraction).

A well-known technique for local outlier detection is the Local Outlier Factor (LOF) [38][39] and its variant the LoOP (Local Outlier Probability)[40] with probabilistic concepts allowing the outlier level of a data point to be defined. The properties and capabilities of these methods make the detection of historical data attractive, particularly because they provide local outlier scores based on the degree of isolation of each vector from the neighborhood. While the LOF detects the outlier data points using the score of an outlier, the LoOP detects them by providing for each data point p an outlier score (belonging to the interval $]0, 1[$) corresponding to the probability that p is an anomaly. Because the distances have positive values, the LoOP algorithm assumes a *half-Gaussian* distribution for these distances. Based on Bayesian inference, the outlier score is directly interpretable as an outlier probability.

Probabilistic inference for data embedding in hyperbolic spaces is a young research area, in which the first main contributions can be dated from the beginning of 2020 (see [24, 41–43] and the references therein). These insights led, for instance, to define the so-called Souriau Gibbs in the Poincaré disk with its Fisher information metric coinciding with the Poincaré Riemannian metric[44]. A novel parametrization for the density of Gaussian on hyperbolic spaces is presented in [41]. This density can be analytically calculated and differentiated with a simple random variate generation algorithm. An alternative is to use a simple Gaussian distribution in hyperbolic spaces,e.g., [45, 46] introduced Riemannian normal distributions for the *univariate normal model*, with an application to the classification of univariate normal populations. Along with the wrapped normal generalisation used in [41], [47] studies a thorough treatment of the maximum entropy normal generalisation. Meanwhile, a lot of applications combining hyperbolic geometry and Variational Auto-Encoders (VAEs) was investigated in [48, 47, 49, 43] based on the fact that VAE latent space components embedded in hyperbolic space help to represent and discover hierarchies. This work introduces a simple framework to detect local outliers for datasets grounded in hyperbolic 2-space referred to as HLoOP (Hyperbolic Local Outlier Probability).

The key contributions of this paper are:

(1) We extend the *Local Outlier Probabilities* (LoOP) algorithm to make it applicable to *hyperbolic models*, e.g, the Poincaré disk model, leading to Hyberbolic 2-space Local Outlier probabilities (HLoOP). Figure 1 illustrates the pipeline to obtain *local* outlier probability distributions in hyperbolic geometry from hierarchical structures.

(2) We derive an expression of a *Gaussian cumulative distribution in hyperbolic spaces* which ensures that the Probabilistic Local Outlier Factor (PLOF) is performed by fully exploiting the information geometry of the observed data.

The rest of this paper is structured as follows: section 2 briefly outlines some concepts from Riemannian geometry for univariate models. In Section 3 we introduce the local outlier probability detection in hyperbolic spaces and discuss how this probability can be computed. Section 4 evaluates the proposed approach on the benchmark dataset " taxonomy embedding from WordNet".

2 A Univariate Normal Model for Hyperbolic Spaces

This section briefly reminds some concepts of Riemannian geometry [46, 50, 47] for the univariate normal model, which are necessary to formally extend the LoOP detection algorithm.

2.1 Riemannian Geometry and Rao Distance

A Riemannian manifold is a real and smooth manifold denoted as $\mathcal{M}$ equipped with a positive definite quadratic form $g_{\boldsymbol{x}} : \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \times \mathcal{T}_{\boldsymbol{x}}\mathcal{M} \mapsto \mathbb{R}$ at each point $\boldsymbol{x} \in \mathcal{M}$, where $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ is the tangent space defined at the local coordinates $\boldsymbol{x} = (x_1, ..., x_n)^T$. Intuitively, it contains all the possible directions in which one can tangentially pass through x. A norm is induced by the inner product on $\mathcal{T}_{\boldsymbol{x}}\mathcal{M} : \| \cdot \|_x = \sqrt{< \cdot, \cdot >_x}$. An infinitesimal volume element is induced on each tangent space $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$. The quadratic form $g_{\boldsymbol{x}}$ is called a Riemannian metric and allows us to define the geometric properties of spaces, such as the angles and lengths of a curve. The Riemannian metric $g_{\boldsymbol{x}}$ is an n-by-n positive definite matrix such that an infinitesimal element of length ds^2 is defined as:

$$ds^2 = (dx_1 \ \cdots \ dx_n)\, g_{\boldsymbol{x}} \begin{pmatrix} dx_1 \\ \vdots \\ dx_n \end{pmatrix}. \tag{1}$$

The Riemannian metric is a well-known object in differential geometry. For instance, the Poincaré disk with a unitary constant negative curvature corresponds to the Riemannian manifold in the hyperbolic space $(\mathbb{H}, g^{\mathbb{H}}_{\boldsymbol{x}})$, where $\mathbb{H} = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\| < 1\}$ is the open unit disk[1]. Its metric tensor can be written from the Euclidean metric $g^E = I_n$ and the Riemannian metric such that $g^{\mathbb{H}}_{\boldsymbol{x}} = \lambda_x^2 g^E$, where $\lambda_{\boldsymbol{x}} = \frac{2}{1-\|\boldsymbol{x}\|^2}$ is the conformal factor. The Rao distance between two points $\boldsymbol{z}_1 = (x_1, y_1)^T$ and $\boldsymbol{z}_2 = (x_2, y_2)^T$ in $\mathbb{H}$ is given as:

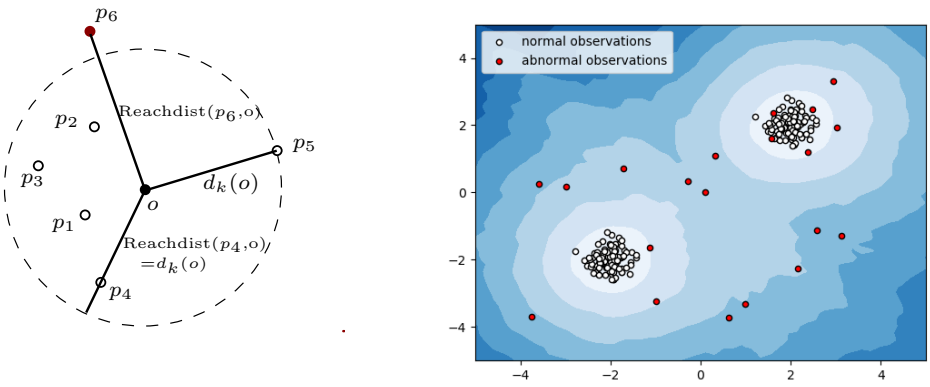
$$d_H(\boldsymbol{z}_1, \boldsymbol{z}_2) = \operatorname{arcosh}\left[1 + 2\frac{||\boldsymbol{z}_1 - \boldsymbol{z}_2||^2}{(1 - ||\boldsymbol{z}_1||^2)(1 - ||\boldsymbol{z}_2||^2)}\right], \tag{2}$$

where arcosh denotes the inverse hyperbolic cosine and $\| \cdot \|$ is the usual Euclidean norm. Different from Euclidean distance, hyperbolic distance grows exponentially fast as we move the points toward the boundary of the open unit disk. There exists many models of hyperbolic geometry (Klein non-conformal model, Beltrami hemisphere model and Lorentz hyperboloid model among others). We can transform one model of hyperbolic geometry into another one by using a one-to-one mapping, which yields an isometric embedding[51].

2.2 Riemannian Prior on the Univariate Normal Model

A Gaussian distribution in $\mathbb{H}$, denoted as $\mathcal{N}_H(\mu, \sigma)$, depends on two parameters, the Fréchet mean $\mu \in \mathbb{H}$ (i.e., the center of mass) and the dispersion parameter $\sigma > 0$, similarly to the Gaussian density

[1] A d-dimensional hyperbolic space, denoted $\mathbb{H}^d$, is a complete, simply connected, d-dimensional Riemannian manifold with constant negative curvature c.



(a) Example of reachability distance.

(b) Example of LOF algorithm (scikit-learn library).

Figure 2: (a) Illustration of the reachability distance for different data points p with regard to o, when $k = 5$. (b) The local density deviation of a given data point with respect to its neighbors.

in the Euclidean space. The Gaussian probability density function in the hyperbolic space, denoted as $p_H(x|\mu, \sigma)$ is defined as follows [45]:

$$p_H(x|\mu, \sigma) = \frac{1}{Z(\sigma)} \exp\left[-\frac{d_H^2(x, \mu)}{2\sigma^2}\right]. \tag{3}$$

Several remarks can be made from Eq. (3): (i.) the main difference between the hyperbolic density $p_H(\cdot)$ and the Gaussian density in the Euclidean space is the use of the squared distance $d_H^2(x, \mu)$ in the exponential (referred to as Rao distance) and a different dispersion dependent normalization constant $Z(\sigma)$ which reduces to $\sqrt{2\pi\sigma^2}$ in the Euclidean case. Note that the constant $Z(\sigma)$ is linked to the underlying geometry of the hyperbolic space (ii.). To define a Gaussian distribution $\mathcal{N}_H(\mu, \sigma)$, through its probability density function, it is necessary to have an exact expression of the normalizing constant $Z(\sigma)$. This constant can be determined using hyperbolic polar coordinates $r = d_H(x, \mu)$ (i.e, a pulling-back) to calculate $Z(\sigma)$ using an integral depending on the Riemannian volume element (iii.). By introducing the parametrization $\boldsymbol{z} = (x, y)^T$ where $x = \mu/\sqrt{2}$, $y = \sigma$ and the Riemannian metric for the univariate Gaussian model is $ds^2(z) = (dx^2 + dy^2)/y^2$, the Riemannian area (since $\mathbb{H}$ is of dimension 2) is $dA(z) = dxdy/y^2$ or $dA(z) = \sinh(r)drd\varphi$ in polar coordinates. For a two-dimensional parameter space, the normalization constant $Z(\sigma)$ was computed in [45], leading to:

$$Z(\sigma) = \int_{\mathbb{H}} \exp\Big(-\frac{r^2}{2\sigma^2}\Big) dA(z) = 2\pi\sigma\sqrt{\frac{\pi}{2}} \exp\left(\frac{\sigma^2}{2}\right) \text{erf}\left(\frac{\sigma}{\sqrt{2}}\right), \tag{4}$$

where erf is the error function. Formula 4 completes the definition of the Gaussian distribution $\mathcal{N}_H(\mu, \sigma)$. In [47] authors shown that when σ get smaller (resp. bigger), the Riemmannian normal pdf get closer (resp. futher) to the wrapped normal pdf [47].

3 Hyperbolic 2-space Local Outlier Probability

3.1 Density-based outlier scoring with a probabilistic approach

This subsection briefly presents the main theoretical principles of some research studies dealing with local outlier probability concepts. A local outlier is a data point that is different or far from most elements of the entire dataset as compared to its local neighborhood, which is measured by the k-Nearest Neighbors (kNN) algorithm [52]. Therefore, the local outlier detection covers a small subset of data points at a given time (Figure 2a). To compute the degree of outlier of a point p in a dataset $\mathcal{D}$, several distances have to be introduced [53]. The k-distance of a point $p \in \mathcal{D}$ denoted as $d_k(p)$ is the distance between $p \in \mathcal{D}$ and its k-nearest neighbor. The notion of k-distance must be used to delimit a neighborhood that contains the k-nearest neighborhood of p. This neighborhood denoted as $N_k(p)$ is defined as $N_k(p) = \{q \in \mathcal{D}\backslash\{p\} | d(q, p) \leq d_k(p)\}$. Thus, the reachability distance denoted $\text{reachdist}_k(p, o)$ of a point $p \in \mathcal{D}$ regarding a point o is defined

as $\mathrm{reachdist}_k(p,o) = \max\{d_k(p), d(p,o)\}$. Based on these definitions, the LOF (Local Outlier Factor) algorithm has been introduced in [38] to improve the kNN approach in the senorio where, for intance in two-dimentional data set, the density of one cluster is significantly higher (resp. lower) than another cluster. To do this, it calculates the local reachable density of the data, and calculates the local outlier factor score according to the local reachable density. Figure 2b presents the local density deviation of a given data point with respect to its neighbors. It considers as outlier samples that have a substantially lower density than their neighbors. In this work, we introduce the HLOF (Hyperbolic Local Outlier Factor) algorithm by replacing $d(q,p)$ with the Rao distance using Eq.(2).

While the LOF (Local Outlier Factor) algorithm detects the outlier data points using the reachibility distance, the Local Outlier Probability (LoOP) algorithm introduces the probabilistic distance of $o \in \mathcal{D}$ to a context set $\mathcal{S} \subseteq \mathcal{D}$, referred to as $\mathrm{pdist}(o,\mathcal{S})$ with the following property:

$$\forall s \in \mathcal{S} : \mathcal{P}[d(o,s) \leq \mathrm{pdist}(o,\mathcal{S})] \geq \varphi. \tag{5}$$

This probabilistic distance corresponds to the radius of a disk that contains a data point of $\mathcal{S}$, obtained from the kNN algorithm, with a certain probability, denoted as φ. The reciprocal of the probabilistic distance can be considered as an estimation of the density of $\mathcal{S}$, i.e, $\mathrm{pden}(\mathcal{S}) = \frac{1}{\mathrm{pdist}(o,\mathcal{S})}$. Assuming that o is the center of $\mathcal{S}$ and the local density is approximately a *half-Gaussian* distribution, the probabilistic set distance of o to $\mathcal{S}$ can be defined as

$$\mathrm{pdist}(o,\mathcal{S}) = \lambda\sigma(o,\mathcal{S}), \tag{6}$$

where $\sigma(o,\mathcal{S}) = \left(\sum_{s\in\mathcal{S}} d(o,s)^2 / |\mathcal{S}|\right)^{1/2}$ is the standard Euclidean distance of o in $\mathcal{S}$ which is similar to the standard deviation. The parameter λ is linked to the selectivity of the detection through the *quantile function* of the normal distribution via the relation $\lambda = \sqrt{2}\mathrm{erfinv}(\varphi)$, where erfinv is the inverse error function.

To be detected as an anomaly for the set $\mathcal{S}$, a data point should deviate from the center of $\mathcal{S}$ for more than λ times the standard distance. For instance, $\lambda = 3$ means that a circle of radius $\mathrm{pdist}(o,\mathcal{S})$ and center o contains any data point of $\mathcal{S}$ with a probability $\varphi \approx 99.7\%$. The resulting probability is the Local Outlier Probability (LoOP) given by

$$\mathrm{LoOP}_{\mathcal{S}}(o) = \max\left\{0, \mathrm{erf}\left(\frac{\mathrm{PLOF}_{\lambda,\mathcal{S}}(o)}{\mathrm{nPLOF}\sqrt{2}}\right)\right\}, \tag{7}$$

where the Probabilisitic Local Outlier Factor (PLOF) is defined as $\mathrm{PLOF}_{\lambda,\mathcal{S}}(o) = (\mathrm{pdist}(\lambda,o,\mathcal{S}))/(\mathrm{E}_{s\in\mathcal{S}}[\mathrm{pdist}(\lambda,s,\mathcal{S})]) - 1$ and a normalization factor nPLOF is such that $\mathrm{nPLOF} = \lambda(\mathrm{E}[\mathrm{PLOF}^2])^{1/2}$. The LoOP value is directly interpretable as the probability of o being an outlier, i.e, close to 0 for points within dense regions and close to 1 for density-based outliers.

3.2 HLoOP Algorithm

This subsection presents the main contribution of this study, which is an adaptation of the LoOP algorithm to data lying in a hyperbolic 2-space. As mentioned above, the LoOP algorithm in an Euclidean space exploits a probabilistic set distance, called $\mathrm{pdist}(o,\mathcal{S})$ (see Eq. 6), to pick the density around o in the context set $\mathcal{S}$ with a probability of φ. The parameter λ gives control over the approximation of the density. To define a local outlier probability adapted to hyperbolic geometry, it is necessary to calculate a new parameter λ_H, which ensures that $\mathrm{pdist}(o,\mathcal{S})$ is performed without undermining hyperbolic geometry. To come up with such a solution, the key idea is to derive a new *quantile function* through an expression of a Gaussian cumulative distribution function (c.d.f) that can be obtained by integrating the probability density function (3) in $\mathbb{H}$. Using polar coordinates (see subsection 2.2, remark iii.), it is possible to calculate this Gaussian c.d.f explicitly. To find the parameter λ_H, we consider the probabilistic distance of $o \in \mathcal{D}$ to a context set $\mathcal{S} \subseteq \mathcal{D}$ using a Riemannian distance $d_H(o,s)$ and the following statistical property:

$$\forall s \in \mathcal{S} : \varphi = \mathcal{P}[0 < d_H(o,s) \leq \lambda_H\sigma_r] = \mathcal{G}_H(\lambda_H\sigma_r). \tag{8}$$

Assuming that o is the center of $\mathcal{S}$ and the set of distances of $s \in \mathcal{S}$ to o is approximately *half-Gaussian* in a hyperbolic space, one can compute the standard deviation σ_r using the Riemannian distance $d_H(o,s)$ with a mean $d_H(o,o) = 0$. Note that the standard deviation of r denoted as σ_r and its probability density function can be determined from the function $\mathcal{G}_H(R) = \mathcal{P}[0 < r < R]$, e.g., $p_H(r,\sigma_r) = \mathcal{G}'_H(r,\sigma_r)$. Theorem 1 presents the main result of this paper.

Theorem 1. *Given $r \in \mathbb{H}$, $\sigma_r > 0$, the Riemannian geometry of the Gaussian cumulative model associated with the distribution defined in Eq.* (3) *is given by*

$$\mathcal{G}_H(r, \sigma_r) = \frac{\pi\sqrt{2\pi}\sigma_r e^{\frac{\sigma_r^2}{2}}}{2Z(\sigma_r)} \times \left(2\mathrm{erf}\Big(\frac{\sigma_r}{\sqrt{2}}\Big) + \mathrm{erf}\Big(\frac{r-\sigma_r^2}{\sigma_r\sqrt{2}}\Big) - \mathrm{erf}\Big(\frac{r+\sigma_r^2}{\sigma_r\sqrt{2}}\Big)\right). \tag{9}$$

Proof. Let $\mathcal{P}[0 < r \leq R]$ and $dA(z) = \sinh(r)drd\varphi$ such that:

$$\mathcal{G}_H(R) = \int_0^{2\pi}\int_0^R \frac{1}{Z(\sigma_r)} \exp\Big(-\frac{r^2}{2\sigma_r^2}\Big)\sinh(r)drd\varphi.$$

The probability density function (pdf) $p_H(\cdot)$ must satisfy the following condition:

$$\int_{\mathbb{H}} p_H(x|\mu,\sigma)d(\mu,\sigma) = 1, \tag{10}$$

where $d(\mu,\sigma)$ is the Lebesgue measure. The cumulative distribution function of the univariate Gaussian distribution of pdf $p_H(\cdot)$ can be computed using Eq. (10) as follows:

$$\begin{aligned}
\mathcal{G}_H(R) &= \frac{2\pi}{Z(\sigma_r)}\int_0^R \frac{e^{\frac{\sigma_r^2}{2}}}{2}\left(e^{-\frac{(r-\sigma_r^2)^2}{2\sigma_r^2}} - e^{-\frac{(r+\sigma_r^2)^2}{2\sigma_r^2}}\right)dr \\
&= \frac{\pi\sqrt{2\pi}\sigma_r e^{\frac{\sigma_r^2}{2}}}{2Z(\sigma_r)}\left(\frac{2}{\sqrt{\pi}}\int_{\frac{-\sigma_r}{\sqrt{2}}}^{\frac{R-\sigma_r^2}{\sqrt{2}\sigma_r}} e^{-u_1^2}du_1 - \frac{2}{\sqrt{\pi}}\int_{\frac{\sigma_r}{\sqrt{2}}}^{\frac{R+\sigma_r^2}{\sqrt{2}\sigma_r}} e^{-u_2^2}du_2\right) \\
&= \frac{\pi\sqrt{2\pi}\sigma_r e^{\frac{\sigma_r^2}{2}}}{2Z(\sigma_r)}\left(2\mathrm{erf}\Big(\frac{\sigma_r}{\sqrt{2}}\Big) + \mathrm{erf}\Big(\frac{R-\sigma_r^2}{\sigma_r\sqrt{2}}\Big) - \mathrm{erf}\Big(\frac{R+\sigma_r^2}{\sigma_r\sqrt{2}}\Big)\right).
\end{aligned} \tag{11}$$

Taking the limit $\mathcal{G}_H(R) \underset{R\to 1}{\longrightarrow} 1$ yields in Eq. (11)

$$Z(\sigma_r) = (2\pi\sigma_r)\sqrt{\frac{\pi}{2}}\exp\left(\frac{\sigma_r^2}{2}\right)\mathrm{erf}\left(\frac{\sigma_r}{\sqrt{2}}\right).$$

We recover the formula given in [45], which completes the proof. □

Combining all these results, the parameter $\lambda_H(\sigma_r)$ is determined by the inverse of $\mathcal{G}_H(\lambda_H\sigma_r)$ (see eq.8) such that

$$\lambda_H(\sigma_r) = \frac{1}{\sigma_r}\mathcal{G}_H^{-1}(\varphi). \tag{12}$$

Hence, whilethe traditional *quantile function* is independent of the standard deviation we have obtained means to directly derive the parameter λ_H that exploits the underlying geometry of the hyperbolic space (see subsection 2.2, remark ii.). The HLoOP algorithm is summarized in Algorithm 1.

Algorithm 1 The procedure of HLoOP algorithm

Input: The data set $X = \{x^i\}_{i=1}^m$ where $x^i = (x_1^i, x_2^i, \cdots, x_n^i) \in \mathbb{R}^n$.
Pre-determined threshold φ, parameter k and hyperbolic distance $d_H(p,q)$;

(1) **Determine** the context set $\mathcal{S}$ of the data point x^i from kNN algorithm;
(2) **Compute** the standard distance σ_r of the context set $\mathcal{S}$;
(3) **Determine** $\mathcal{G}_H^{-1}(\varphi)$ to derive the parameter λ_H by eq.12;
(4) **Calculate** the probabilistic set distance $\mathrm{pdist}_k(x^i)$ of the data point x^i by eq.6;
(5) **Compute** the local outlier probability $\mathrm{LoOP}_k(x^i)$ of the data point x^i by eq.7;

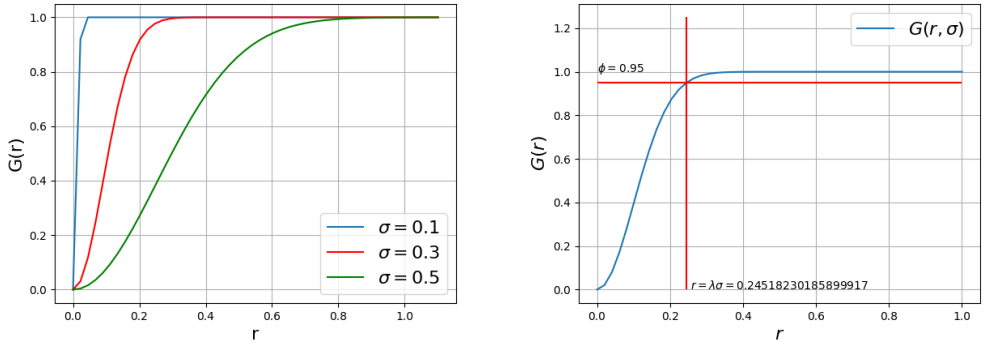
Output: Abnormal data points.

3.3 Implementation Details

Before presenting the performance of the algorithm described above, it is interesting to discuss some aspects related to the implementation of the HLoOP algorithm. Most of the steps used in the implementation of the HLoOP algorithm are directly related to the Euclidean LoOP, except that the distances are no longer Euclidean but hyperbolic. However, the computation of the significance λ cannot be computed as in the Euclidean LoOP. While an analytic expression of the Gaussian quantile function is known in the Euclidean space, the derivation of the cumulative distribution in the Poincare disk, illustrated on Figure 3a, does not lead to an analytic formulation of its inverse $\mathcal{G}_H^{-1}$. Actually, an analytical expression of $\mathcal{G}_H^{-1}$ is not needed to compute λ providing the value of $r = \lambda\sigma$ for which $\mathcal{G}_H(r, \sigma) = \varphi$ can be determined. This is equivalent to solving:

$$\mathcal{G}_H(r, \sigma) = \varphi,$$

for a given pair (σ, φ), which can be done using Newton's method. Once we have obtained r, as shown on Figure 3b, the significance can be determined by using the relation $r = \lambda\sigma$, yielding $\lambda = r/\sigma$. All the elements required to implement the HLoOP algorithm are now available. The next Section is dedicated to the assessment of the performance of this algorithm.



(a) Cumulative univariate Gaussian model associated with the distribution Eq. (3).

(b) Newton's method to determine $r = \mathcal{G}^{-1}(\varphi)$.

Figure 3: Cumulative distribution $\mathcal{G}_H$ in the Poincare disk and resolution of $\mathcal{G}_H(r, \sigma) = \varphi$.

4 Results

4.1 Performance of the HLoOP algorithm on a toy dataset

The HLoOP method is first used to detect the outliers in a toy dataset, with a reduced number of points. The dataset was generated as follows: first, some vectors were generated uniformly in two circular areas located in the Poincaré Disk (clusters **A** and **B**). Then, each area is filled with 40 points whose positions are pulled from the normal distribution $\mathcal{N}(\cdot, R\boldsymbol{I}_2)$ where $\boldsymbol{I}_2$ is the 2×2 identity matrix. Five points located outside these areas (cluster **C**) constitute the outliers of the toy dataset, which is finally composed of $2 \times 40 + 5 = 85$ points. The HLoOP algorithm is applied to this dataset and its performance is compared to that of HLOF. As a first test, we compute the HLOF and HLoOP values of each point of the embedding for $k = 15$ and, for HLoOP a treshold $\varphi = 95\%$. Figures 4 and 5 show the different points that are surrounded by a circle whose radius is proportional to the HLoOP or HLOF value. We observe that for both methods (HLoOP or HLOF), the outliers (cluster **C**) have a score higher than the inliers (clusters **A** and **B**). For the HLoOP, this correspond to the probability of a point to be an outlier, while for the HLOF, the interpretation of the score is less straightforward. It is also interesting to note that cluster **A** highlights a weakness of HLOF : like LOF, it is designed for clusters of uniform density. The probability of datapoints in cluster **A** being generated by Gaussian distribution, the HLOF assigns high outlier scores while these points were in fact generated by the cluster. The HLoOP value is much more useful here : there is a clear chance the the point is an outlier, but it is also very likely it is just an outer point of the clusters normal distribution.

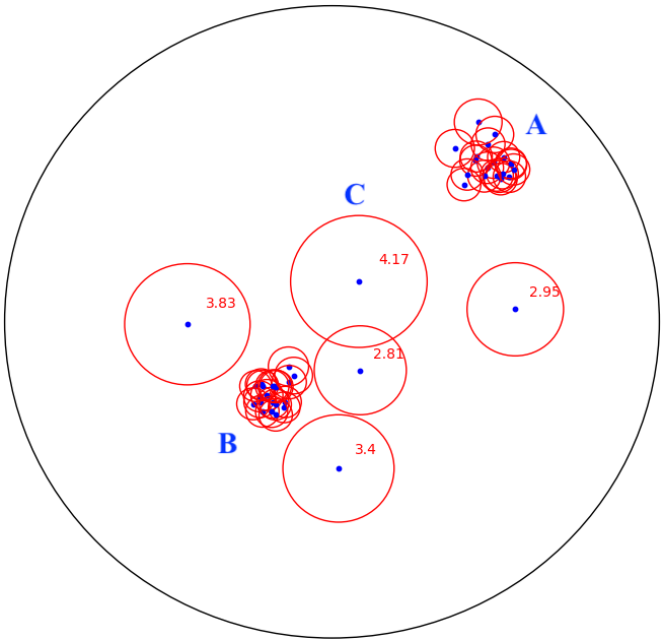


Figure 4: Embedding of the toy dataset in the Poincaré disk : HLOF values.

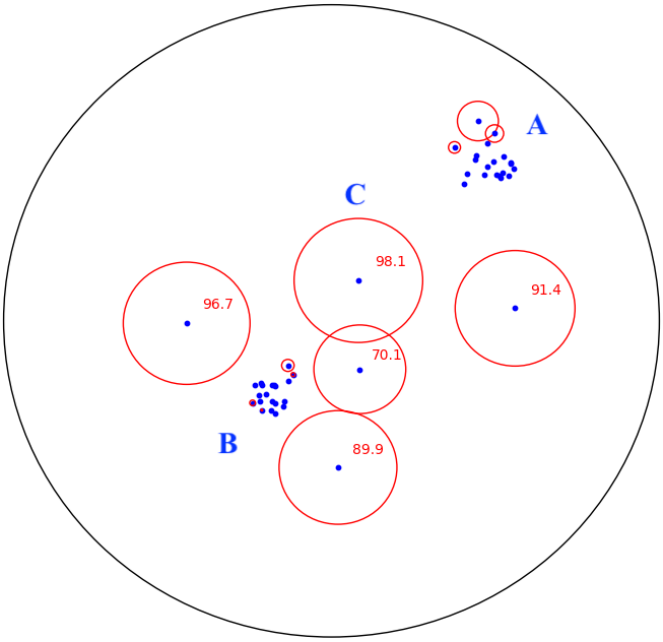


Figure 5: Embedding of the toy dataset in the Poincaré disk : HLoOP values – Some points have very small H-LoOP values and their associated circle do not appear in the figure –.

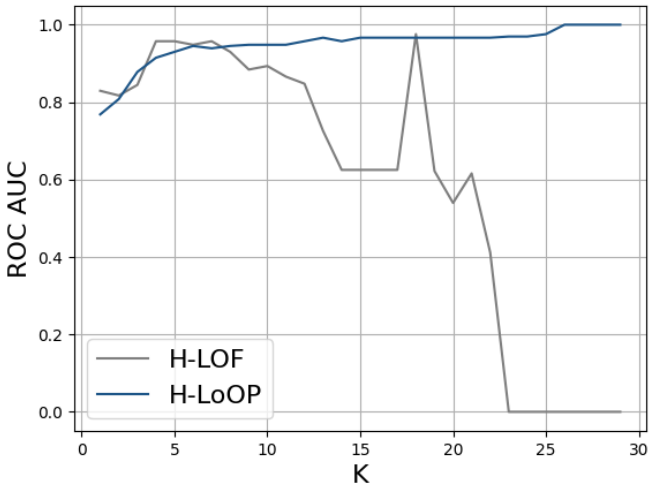


Figure 6: AUC ROC - Outlier detection in a toy dataset

The metric used to quantify the quality of the outlier detection is the Area Under the Receiving Operator Curve (AUC-ROC). We recall that value of AUC-ROC near 0 coreesponds to very poor detection performance (near 0% of the decision made by the algorithm are correct), while an AUC ROC close to 1 means that the algorithm is making very few errors. As observed in Figure 6, the HLoOP algorithm provides very good anomaly detections: for $k > 2$ (number of neighbors considered to evaluate the density of the context set $\mathcal{S}$), the number of true positives (actual outliers detected as outliers) is between 95 and 100 %, which is a very good result. In the meantime, the performance of HLOF is more contrasted and strongly dependent on the value of k. In particular, for higher values of k, the HLOF performance dramatically decreases. With such promising results, the next section aims at assessing the performances of HLoOP on a bigger dataset, containing up to 1000 points.

4.2 Evaluating the performances on the Wordnet/Mammals subgraph

This section evaluates the performance of H-LoOP on a subgraph of the WORDNET database. WORDNET is a lexical dataset composed by 117000 synsets, which corresponds to nouns, adjectives or verbs that are linked by conceptual relations. Several subgraphes are known to exist in this dataset. Among them, we decided to apply H-LOF and H-LoOP on a group of 1180 synsets from the subgraph "Mammals". The dataset was corrupted by 11 outliers corresponding to nouns of animals that are not mammals (i.e., fishes, reptiles or birds) and was embedded in the Poincaré Disk using the algorithm of Nickel et al. (2017). The values of HLoOP and HLOF were calculated for the points of this embedding. The Area Under the Receiving Operator Curve (AUC ROC) was finally calculated for both HLOF and HLoOP for several value of K. As shown in Figure 7, the performance of HLoOP is better than HLOF for all values of K, with a ROC AUC larger than 0.98, while the HLOF leads to a ROC value less than 68%. In addition to its good performance, the HLoOP algorithm leads to AUC values that are quite independent of K, which is outstanding.

5 Conclusion and perspectives

This paper has presented extensions of the Local Outlier Factor (LOF) and Local Outlier Probability (LoOP) algorithms, respectively refered to as Hyperbolic LOF (HLOF) and Hyperbolic LoOP (HLoOP). Rather than working in the Euclidean space, these extensions work in a specific model of the hyperbolic space, namely the Poincaré Disk. Both algorithms are density based and compare the density of a point's neighborhood with the density of others's neighborhood. On one hand, the HLOF compute the density based on a deterministic distance (called reachibility distance) while the HLoOP introduces the notion of probabilistic distance and returns for each point its probability of outlierness. Simulations results conducted on a toy dataset have shown that the HLoOP algorithm allows a better

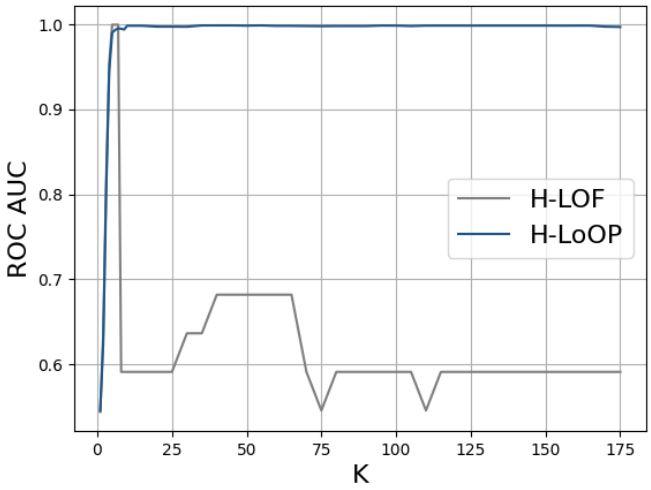


Figure 7: AUC ROC - Outlier detection in our corrupted subgraph of Wornet/Mammals

distinction of outliers and inliers when compared to HLOF. While the HLoOP directly provides the probability of each point to be an outlier, the HLOF returns a score whose interpretation is not straightforward and depends on the dataset under study. Evaluations of the areas under the receiver operational characteristics of data in the Poincaré disk have confirmed a better detection performance of HLoOP when compared to HLOF. The results obtained with this dataset have also shown that the HLoOP performance seems to be less sensitive to the number of neighbors taken into account in the computation of the density of the context set than for HLOF. Given these promising results, we have embedded the *mammals* subset of the Wordnet dataset in the Poincaré disk after introducing artificial outliers. The HLOF and HLoOP values and the areas under the receiver operational characteristics HLOF and HLoOP algorithms confirm the results obtained with the previous dataset.
Future work includes the extension of these two algorithms to the Lorentz's disk, i.e., to another model of the hyperbolic space. Indeed, it has been shown that the Poincaré disk presents some numerical instabilities that are not observed in the Lorentz model. Moreover, it would be interesting to apply the HLoOP and HLOF algorithms to more complex datasets, with more points and more attributes. For instance, given the growing interest of the hyperbolic geometry in the computer vision domain, it could be worthy to try using the HLoOP and HLOF to detect outliers in a set of images. Finally, the hyperbolic geometry could be used to derive new outlier detection algorithms based on isolation forest or one-class support vector machines.

References

[1] Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric deep learning: going beyond Euclidean data. *CoRR* **2016**, *abs/1611.08097*.

[2] Anderson, J. W. *Hyperbolic geometry*; Springer Science & Business Media, 2006.

[3] Peng, W.; Varanka, T.; Mostafa, A.; Shi, H.; Zhao, G. Hyperbolic Deep Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2022**, *44*, 10023–10044.

[4] Nickel, M.; Kiela, D. Poincaré Embeddings for Learning Hierarchical Representations. Advances in Neural Information Processing Systems. 2017.

[5] Sarkar, R. Low distortion Delaunay embedding of trees in hyperbolic plane. *In International Symposium on Graph Drawing* **2011**, 355–366.

[6] Sala, F.; De Sa, C.; Gu, A.; Re, C. Representation Tradeoffs for Hyperbolic Embeddings. Proceedings of the 35th International Conference on Machine Learning. 2018; pp 4460–4469.

[7] Ganea, O.; Bécigneul, G.; Hofmann, T. Hyperbolic entailment cones for learning hierarchical embeddings. *International Conference on Machine Learning* **2018**, 1646–1655.

[8] Liu Q, K. D., Nickel M Hyperbolic graph neural networks. *Advances in Neural Information Processing Systems* **2019**,

[9] Chami, I.; Ying, R.; Ré, C.; Leskovec, J. Hyperbolic Graph Convolutional Neural Networks. *Proceedings of the 33rd International Conference on Neural Information Processing Systems* **2019**,

[10] Dai, J.; Wu, Y.; Gao, Z.; Jia, Y. A hyperbolic-to-hyperbolic graph convolutional network. *Computer Vision and Pattern Recognition* **2021**,

[11] Cetin, E.; Chamberlain, B.; Bronstein, M.; Hunt, J. J. Hyperbolic Deep Reinforcement Learning. *arXiv* **2022**,

[12] Atigh, M. G.; Schoep, J.; Acar, E.; van Noord, N.; Mettes, P. Hyperbolic image segmentation. *Computer Vision and Pattern Recognition* **2022**, 4453–4462.

[13] Gao, Z.; Wu, Y.; Jia, Y.; Harandi, M. Curvature generation in curved spaces for few-shot learning. *International Conference on Computer Vision* **2021**, 8691–8700.

[14] Suris D, V. C., Liu R Learning the predictability of the future. *Computer Vision and Pattern Recognition* **2021**, 12602–12612.

[15] Dengxiong X, K. Y. Generalized open set recognition via hyperbolic side information learning. *Winter Conference on Applications of Computer Vision* **2023**, 3992–4001.

[16] Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised vision transformers. *International Conference on Computer Vision* **2021**, 9650–9660.

[17] Kleinberg, R. Geographic Routing Using Hyperbolic Space. IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications. 2007; pp 1902–1909.

[18] Cvetkovski, A.; Crovella, M. Hyperbolic Embedding and Routing for Dynamic Graphs. IEEE INFOCOM 2009. 2009; pp 1647–1655.

[19] Cassagnes, C.; Tiendrebeogo, T.; Bromberg, D.; Magoni, D. Overlay addressing and routing system based on hyperbolic geometry. 2011 IEEE Symposium on Computers and Communications (ISCC). 2011; pp 294–301.

[20] Lv, S.; Li, H.; Wu, J.; Bai, H.; Chen, X.; Shen, Y.; Zheng, J.; Ding, R.; Ma, H.; Li, W. Routing Strategy of Integrated Satellite-Terrestrial Network Based on Hyperbolic Geometry. *IEEE Access* **2020**, *8*, 113003–113010.

[21] Higgott, O.; Breuckmann, N. P. Subsystem Codes with High Thresholds by Gauge Fixing and Reduced Qubit Overhead. *Physical Review X* **2021**, *11*.

[22] Higgott, O.; Breuckmann, N. P. Constructions and performance of hyperbolic and semi-hyperbolic Floquet codes. 2023.

[23] Mettes, P.; Atigh, M. G.; Keller-Ressel, M.; Gu, J.; Yeung, S. Hyperbolic Deep Learning in Computer Vision: A Survey. *arXiv 2305.06611* **2023**,

[24] Nielsen, F.; Okamura, K. Information measures and geometry of the hyperbolic exponential families of Poincaré and hyperboloid distributions. 2022.

[25] Su, S.; Xiao, L.; Ruan, L.; Gu, F.; Li, S.; Wang, Z.; Xu, R. An Efficient Density-Based Local Outlier Detection Approach for Scattered Data. *IEEE Access* **2019**, *7*, 1006–1020.

[26] Alghushairy, O.; Alsini, R.; Soule, T.; Ma, X. A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. *Big Data Cogn. Comput.* **2020**, *5*.

[27] Souiden, I.; Brahmi, Z.; Toumi, H. A Survey on Outlier Detection in the Context of Stream Mining: Review of Existing Approaches and Recommadations. *Intelligent Systems Design and Applications - 16th International Conference on Intelligent Systems Design and Applications* **2016**,

[28] Campos, G. O.; Zimek, A.; Sander, J.; Campello, R. J. G. B.; Micenková, B.; Schubert, E.; Assent, I.; Houle, M. E. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery* **2016**, *30*, 891–927.

[29] Wang, S. A Comprehensive Survey of Data Mining-Based Accounting-Fraud Detection Research. 2010 International Conference on Intelligent Computation Technology and Automation. 2010; pp 50–53.

[30] Lin, J.; Keogh, E.; Fu, A.; Van Herle, H. Approximations to magic: finding unusual medical time series. 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05). 2005; pp 329–334.

[31] Portnoy, L. Intrusion detection with unlabeled data using clustering. 2000.

[32] García-Teodoro, P.; Díaz-Verdejo, J.; Maciá-Fernández, G.; Vázquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security* **2009**, *28*, 18–28.

[33] Yeung, D.-Y.; Ding, Y. Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition* **2003**, *36*, 229–243.

[34] Phua, C.; Lee, V. C.-S.; Smith-Miles, K.; Gayler, R. W. A Comprehensive Survey of Data Mining-based Fraud Detection Research. *ArXiv* **2010**, *abs/1009.6119*.

[35] Thiprungsri, S.; Vasarhelyi, M. A. Cluster Analysis for Anomaly Detection in Accounting Data: An Audit Approach 1. *The International Journal of Digital Accounting Research* **2011**, *11*, 69–84.

[36] Bolton, R. J.; Hand, D. J. Unsupervised Profiling Methods for Fraud Detection. 2002.

[37] Bansal, R.; Gaur, N.; Singh, S. N. Outlier Detection: Applications and techniques in Data Mining. 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence). 2016; pp 373–377.

[38] Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; Sander, J. LOF: identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD international conference on Management of data* **2000**, 93–104.

[39] Guo, M.; Pan, S.; Li, W.; Gao, F.; Qin, S.; Yu, X.; Zhang, X.; Wen, Q. Quantum algorithm for unsupervised anomaly detection. *Physica A: Statistical Mechanics and its Applications* **2023**, *625*, 129018.

[40] Kriegel, H.-P.; Kröger, P.; Schubert, E.; Zimek, A. LoOP: local outlier probabilities. *Proceedings of the 18th ACM conference on Information and knowledge management* **2009**, 1649–1652.

[41] Cho, S.; Lee, J.; Park, J.; Kim, D. A Rotated Hyperbolic Wrapped Normal Distribution for Hierarchical Representation Learning. *ArXiv* **2022**, *abs/2205.13371*.

[42] Nickel, M.; Kiela, D. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. *CoRR* **2018**, *abs/1806.03417*.

[43] Nagano, Y.; Yamaguchi, S.; Fujita, Y.; Koyama, M. A Wrapped Normal Distribution on Hyperbolic Space for Gradient-Based Learning. *International Conference on Machine Learning* **2019**,

[44] Barbaresco, F. Lie Group Machine Learning and Gibbs Density on Poincaré Unit Disk from Souriau Lie Groups Thermodynamics and SU(1,1) Coadjoint Orbits. Geometric Science of Information. Cham, 2019; pp 157–170.

[45] Said, S.; Bombrun, L.; Berthoumieu, Y. New Riemannian Priors on the Univariate Normal Model. *Entropy* **2014**, *16*, 4015–4031.

[46] Pennec, X. Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *Journal of Mathematical Imaging and Vision* **2006**, *25*, 127–154.

[47] Mathieu, E.; Lan, C. L.; Maddison, C. J.; Tomioka, R.; Teh, Y. W. Continuous Hierarchical Representations with Poincaré Variational Auto-Encoders. Neural Information Processing Systems. 2019.

[48] Ovinnikov, I. Poincaré Wasserstein Autoencoder. *CoRR* **2019**, *abs/1901.01427*.

[49] Cho, S.; Lee, J.; Kim, D. GM-VAE: Representation Learning with VAE on Gaussian Manifold. *arXiv preprint arXiv:2209.15217* **2022**,

[50] Petersen, P. *Riemannian Geometry*; Graduate Texts in Mathematics; Springer New York, 2006.

[51] Nielsen, F.; Nock, R. Hyperbolic Voronoi Diagrams Made Easy. Proceedings of the 2010 International Conference on Computational Science and Its Applications. USA, 2010; pp 74–80.

[52] Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **1967**, *13*, 21–27.

[53] Hu, L.-Y.; Huang, M.-W.; Ke, S.-W.; Tsai, C.-F. The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus* **2016**, *5*, 1304.