Visualizing High-Dimensional Hyperbolic Data

Haoran Guo
Yunhui Guo
Stella Yu

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-127
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-127.html

May 14, 2022

Acknowledgement

I’d first like to thank Professor Stella Yu, for being a great advisor and giving me the opportunity to do interesting research, and Professor Yi Ma, for reviewing this thesis. Next,
I’d like to give a very big thanks to Yunhui Guo, for providing invaluable guidance and
mentorship in not only co-authoring this project, but throughout my degree. Thank you to
Tsung-Wei Ke as well, for his important mentorship in starting my research career.

Thank you to all my friends for being my friends and carrying me through a great experience
in school.

Thank you to my parents, whom all my good traits come from, for the unlimited support
and love.

Visualizing High-Dimensional Hyperbolic Data

by

Haoran Guo

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Stella Yu, Chair
Professor Yi Ma

Spring 2022

Visualizing High-Dimensional Hyperbolic Data

Abstract

Visualizing High-Dimensional Hyperbolic Data

by

Haoran Guo

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Stella Yu, Chair

Hierarchies can often be found in real-world data and semantics. Hyperbolic space can naturally embed hierarchical structures, making it an attractive alternative to traditional Euclidean or spherical space when learning representations. Many popular machine learning methods and architectures have been adapted to embed and use hierarchical data, leading to significant improvements over Euclidean counterparts. Additionally, high-dimensional embeddings can capture more information than low-dimensional ones, which can also lead to performance improvements. While high-dimensional hyperbolic embeddings can lead to better representations, visualizing them in a human understandable way can be challenging. Visualizations of learned embeddings are important for both understanding the representation model and characteristics of the data, so for this and other reasons, many hyperbolic models do not leverage high-dimensional embeddings. To address this problem, we propose CO-SNE which extends the Euclidean space visualization tool, t-SNE to hyperbolic space. CO-SNE is able to deflate high-dimensional embeddings into low-dimensional space without losing their hyperbolic characteristics. We present results that show CO-SNE outperforms previous methods on visualizing high-dimensional hyperbolic data, including real-life biological data and learned hyperbolic embeddings. We also show the the hierarchical nature of image segmentation by visualizing the results of hierarchical semantic segmentation. Overall, CO-SNE is able to successfully visualize high-dimensional hyperbolic data, resulting in visualizations that can present insight on the data and the methods generating the data.

To my mom and dad

Contents

List of Figures

List of Tables

Acknowledgments

I'd first like to thank Professor Stella Yu, for being a great advisor and giving me the opportunity to do interesting research, and Professor Yi Ma, for reviewing this thesis. Next, I'd like to give a very big thanks to Yunhui Guo, for providing invaluable guidance and mentorship in not only co-authoring this project, but throughout my degree. Thank you to Tsung-Wei Ke as well, for his important mentorship in starting my research career.

Thank you to all my friends for being my friends and carrying me through a great experience in school.

Thank you to my parents, whom all my good traits come from, for the unlimited support and love.

Chapter 1

Introduction

Hierarchical structures can be found in many datasets and real-world applications. Social networks [14] and complex networks [1] are representative examples of hierarchical data. Words datasets with hypernymy relations [29] and visual inputs [20, 25] also exhibit hierarchical characteristics. Euclidean and spherical spaces are common choices for embedding data, but they cannot embed hierarchical data without distortion. Thus, hyperbolic space has been widely used as an alternative in representation learning [20, 25, 29] on hierarchical data. Hyperbolic space is a non-Euclidean space with interesting characteristics, as visualized in Figure 1.1. Importantly, the hyperbolic metric can closely approximate the tree metric since hyperbolic spaces expand exponentially in volume in contrast to Euclidean space which exhibits polynomial expansion. As such, hyperbolic spaces are continuous analogues to trees, which naturally represent hierarchies [20]. Algorithms that operate directly on hyperbolic space [10, 7, 41] have been introduced, furthering the prevalence of hyperbolic embeddings. These studies have shown performance improvements over Euclidean counterparts, especially when working with hierarchical data.

Learning high dimensional embeddings can generally lead to better hyperbolic representation quality, but often many methods still stick with two-dimensional hyperbolic space [17, 29]. One main reason is for the ease of visualization, which is crucial for better understanding of the embedding space. Visualizations can elucidate underlying structures of the embedded data, such as hierarchies in particular for hyperbolic data, as well as insights on what a representation model is learning. For hyperbolic data, the Poincaré ball model is one of the most popular models among several isometrically equivalent models for representing hyperbolic space and in hyperbolic representation learning [13, 17, 29]. Two-dimensional hyperbolic embeddings based on the Poincaré ball model can be easily visualized within a Euclidean unit circle. However, for high-dimensional hyperbolic data, the task is not so straightforward as many visualization methods assume Euclidean data input and/or visualize in a Euclidean space.

Embeddings in the Poincaré ball model have two main properties that should ideally be maintained when visualized:

1. The embeddings have a global hierarchical structure such that root nodes are in the center of the ball and leaf nodes are close to the boundary of the ball.

2. The embeddings possess a local similarity structure. Sibling nodes should be close in the embedding space.

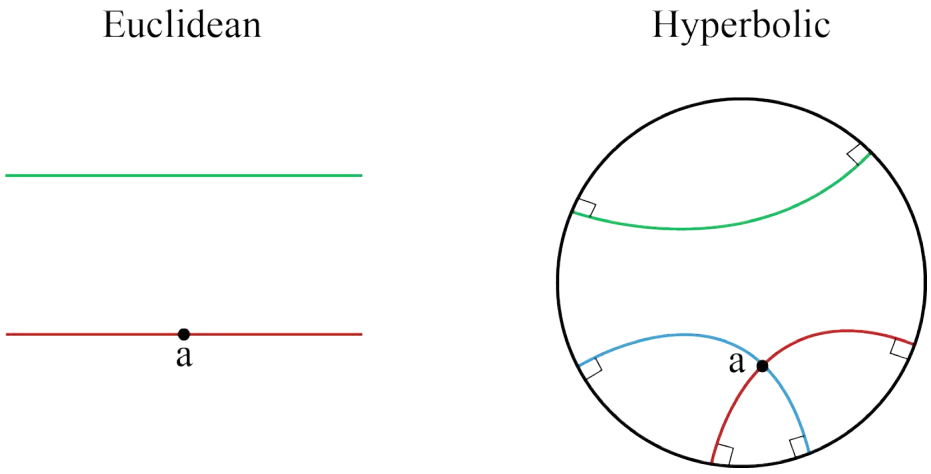


Figure 1.1: In Euclidean space, there is only one parallel line (red) to the green line that intersects point a. In hyperbolic space (visualized on the Poincaré disk), there are multiple lines (blue and red) that are parallel to the green line that intersect point a. The colored lines in the hyperbolic model are known as geodesics, a generalization of Euclidean straight lines.

One of the most popular visualization methods for high-dimensional Euclidean embeddings is t-SNE [40]. Since it is built for Euclidean embeddings, it does not preserve the global hierarchical structure of hyperbolic embeddings. HoroPCA [5] is a recently proposed extension of PCA [18] to hyperbolic space that does specifically cater to hyperbolic data. However, it can struggle to preserve the local similarity structure of the data. Therefore, we propose a new method, CO-SNE, which aims to fill the need for a visualization method suitable for high-dimesional hyperbolic embeddings (see Figure 1.2). CO-SNE extends the standard t-SNE method's ability to maintain local similarity to hyperbolic space by adopting hyperbolic versions of the normal and Cauchy distributions. To maintain the global hierarchical structure we use a distance loss function which seeks to preserve the individual distances of high-dimensional hyperbolic embeddings to the Origin in low-dimensional hyperbolic space.

This thesis is based on the original CO-SNE work [12]. We will first introduce some related work in Chapter 2. Then, Chapter 3 will present a more detailed look into hyperbolic space, the Poincaré ball model, and t-SNE. In Chapter 4, CO-SNE and its development from HT-SNE is introduced. Experiments and results of CO-SNE as compared with relevant

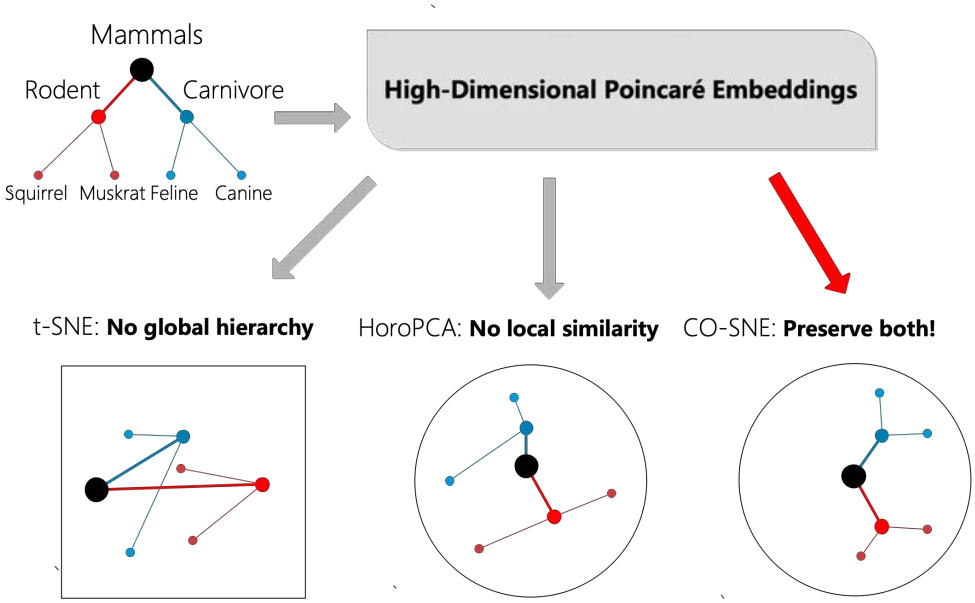


Figure 1.2: Hyperbolic space is suitable for hierarchical data such as the mammal subtree of WordNet visualized above. The visualizations from t-SNE, HoroPCA, and CO-SNE on the high-dimensional hyperbolic Poincaré embeddings of the mammal subtree shows the ability of CO-SNE to better preserve the global hierarchy with the root node in the center and leaf nodes toward the boundary. Additionally, the local similarities are maintained better since the sibling nodes are closer together.

baseline methods are presented in Chapter 5. The conclusion and future work discussion follow in Chapter 6.

Chapter 2

Related Work

This chapter surveys previous work, focusing on hyperbolic learning methods that could be better supported and extended by visualizations from CO-SNE as well as previous data visualization and dimensionality reduction methods.

2.1 Hyperbolic Learning

Many traditionally Euclidean learning methods have been extended to hyperbolic space. These methods generally introduce hyperbolic analogues to Euclidean operations and distributions, providing frameworks for learning and working with hyperbolic embeddings.

Large-margin Classification

Large-margin Classification [7] in hyperbolic space was proposed by adapting Support Vector Machines (SVM) [15] from Euclidean geometry to hyperbolic. They find the analogous set of possible hyperbolic decision rules, which are decision hyperplanes not unlike the linear decision boundaries familiar in the Euclidean SVM. For the hierarchically structured datasets they present, the hyperbolic embeddings outperform the Euclidean counterparts and the hyperbolic SVM they present matches or outperforms the Euclidean SVM. These datasets include a biological gene function prediction task and social networks data, which have natural hierarchical structure. Another work provides a robust large-margin classification [41] method in hyperbolic space. By adapting learning via adversarial examples [6] to hyperbolic space, they give the first theoretical guarantees for learning a hyperbolic classifier.

Hyperbolic Neural Networks (++)

Hyperbolic Neural Networks (HNNs) [10] is a work that extends several familiar neural network architectures and layers such as multinomial logistic regression (MLR), fully connected layers and Recurrent Neural Networks to hyperbolic space. The use gyrovector space operations [39] to parallel the Euclidean vector operations. The hierarchical datasets they present

are mostly word and sentence based datasets, including the WordNet noun hierarchy we experiment on. Another work [20] further proves the efficacy of hyperbolic neural network layers by appending them to the end of several computer vision architectures. They find this setup can outperform Euclidean neural networks on tasks such as few-shot classification and person re-identification.

In a follow-up work, Hyperbolic Neural Networks++ [36], first reformulates the hyperbolic MLR from [10] by reducing the number of parameters necessary for linear functions, $\langle \mathbf{a}, \mathbf{x} \rangle - b$. Previously, [10] had required almost twice as many parameters in their formulation of the hyperbolic MLR. This improvement lead to their version of hyperbolic fully connected layers as well as their extension to hyperbolic convolutional layers. They also introduce some ideas for building hyperbolic attention mechanisms which are also explored in the previous work introduced below.

Hyperbolic Attention Networks

Hyperbolic attention networks [11] are proposed by extending attention operations to hyperbolic space in a manner similar to [10]. The core computation for attention [3] can be broken down into a matching and aggregation phase. The matching computes attention weights, while the aggregation phase involves computing a weighted average using the matching weights. For hyperbolic attention matching, [11] takes advantage of the hyperbolic metric. Aggregation is the done through the Einstein midpoint, which is an extension of the weighted midpoint to hyperbolic space posed by Ungar [38]. They show efficacy for representing graphs and show comparable performance to Euclidean transformers on a neural machine translation task.

Poincaré Variational Autoencoders

The variational autoencoder (VAE) [21], a popular generative model, has also been extended to hyperbolic space [25]. The use hyperbolic counterparts to the traditionally Euclidean latent space distributions of the VAE. We use one of these distributions, the Riemannian normal, in CO-SNE. With a decoder architecture that specifically takes into account the hyperbolic geometry, [25] is able to yield more interpretable representations. As seen in this work, the latent space used is two-dimensional in order to easily visualize representations. Meanwhile, the best results are with higher dimensional representations. This draws attention to the need for a hyperbolic data visualization method.

Hyperbolic Segmentation Methods

Some of the previously detailed methods have been able to learn appealing representations in hyperbolic space, leading to [17] and [42] successfully leveraging them for segmentation tasks. A 3D hyperbolic VAE is introduced by [42] as the basis for their unsupervised 3D segmentation method. They demonstrate the effectiveness on biological-based datasets including

3D MRI scans. [42] uses hyperbolic embeddings to better capture hierarchical structure in segmented objects, leading to superior results than Euclidean space representations in their ablation study.

The hyperbolic learning works introduced here show results where hyperbolic methods perform better over Euclidean counterparts, usually due to some underlying hierarchical structure at play. They use hyperbolic embeddings in the intermediate representations, and for high-dimensional representations, methods for visualization are needed.

2.2 Data Visualization and Dimensionality Reduction

Data visualization of high-dimensional data involves generating low-dimensional representations that maintain the interesting structures and relative similarities of the given data. Dimensionality reduction methods can directly accomplish this task and be used for data visualization, although they may not be tuned for the specific goal of visualization.

SNE

CO-SNE's main framework structure is first introduced by Stochastic Neighbor Embedding (SNE) [16]. SNE is a probabilistic approach which computes an asymmetrical probabilities for between pairs of points. These probabilities represent the likelihood a particular point i would pick another point j as its neighbor. Two probability distributions are created in this way, one for the high-dimensional input data and one for the low-dimensional representation being created. The final low-dimensional representation is then found by gradient descent on the Kullback-Leibler divergence between these two distributions.

t-SNE

t-distributed stochastic neighbor embedding (t-SNE) [40] is one of the most widely used data visualization methods today. It is heavily based on SNE, but introduces a symmetrical probability model between points. It also uses the Student's t-distribution to model the similarities between low-dimensional points rather than a normal distribution. These adjustments allow t-SNE to improve on SNE, directly addressing some weaknesses of SNE such as the "crowding problem", which we discuss more in Section 4.2. We base CO-SNE off of t-SNE so we present more details of the methodology in Section 3.2.

UMAP

Uniform Manifold Approximation and Projection (UMAP) [26] is a more recent method that uses a manifold learning technique. It can be viewed in a similar lens as t-SNE by considering that both methods essentially construct a weighted graph representation of the

high-dimensional data and optimize a low dimensional representation to be structurally similar. UMAP uses fuzzy simplicial sets [37] to construct the weighted graph, with the "fuzzy" nature of the graph due to decreasing likelihoods of connections as distance grows. The selection of the weights of the edges are different and UMAP uses a cross-entropy based loss function. UMAP is generally better than t-SNE for preserving global structure and in terms of speed. However, it can struggle on some structures as compared to t-SNE, visualized in Figure 2.1. While the metric in UMAP is exchangeable, it does not provide any specific support for hyperbolic data.

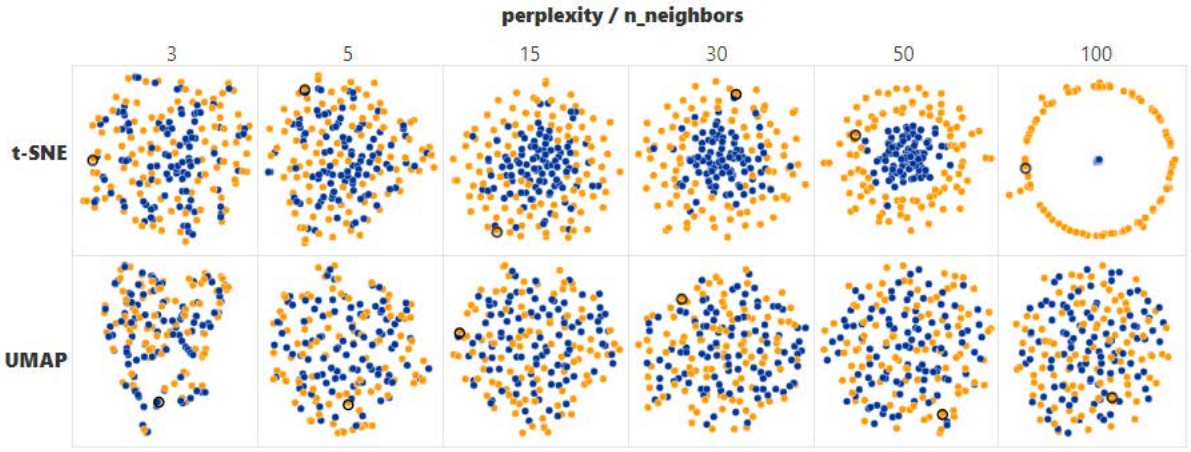


Figure 2.1: Visualizations of t-SNE and UMAP on a dense cluster inside of a wider, sparse cluster from [8]. t-SNE is able to separate the clusters, but UMAP is not. However, UMAP generally is better for preserving global structures.

PCA

Principle Component Analysis (PCA) is a well-studied technique for dimensionality reduction [18]. PCA reduces the dimensionality of data while preserving as much variation of information as possible. As a dimensionality reduction method, PCA can be used for visualization high-dimensional of data. Compared to t-SNE and UMAP, it cannot maintain local structures as well, but does provide many advantages. One main advantage is that it is much more simple to interpret - exactly how stochastic methods like t-SNE and UMAP warp high-dimensional data into low-dimensional space isn't explicit. Additionally, as a deterministic method, there are no hyperparameters to tune that can affect the results. PCA would not be suited to directly apply to hyperbolic data, so HoroPCA (below) attempts to address that problem.

HoroPCA

HoroPCA [5] introduces a hyperbolic version of PCA by generalizing the three main aspects of PCA to hyperbolic space: a sequence of affine subspaces spanned by a set of directions, a projection method onto those spaces, and an objective for selecting the set of directions. The nested sequence of affine subspaces made of the linear spans of more and more components is known as a flag in Euclidean space. The hyperbolic analogy used by HoroPCA is a nested sequence of geodesic submanifolds. Figure 2.2 shows the horospherical projections that substitute orthogonal projections in Euclidean space. Similar to orthognal Euclidean projections, the horospherical projections can preserve distance along a direction after projection. Finally, the variance objective is easily converted by replacing the Euclidean distance with hyperbolic distance. The authors show HoroPCA's efficacy on dimensionality reduction, hyperbolic data whitening, and visualization. As with PCA, however, HoroPCA should not be able to maintain local similarities as well as methods such as t-SNE.

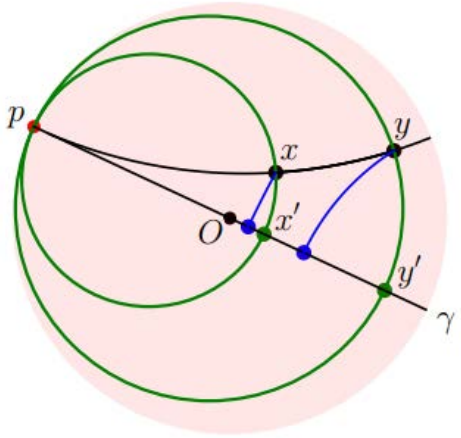


Figure 2.2: Visualizations of two points, x and y, and their horospherical projections, x' and y', onto a single direction, the geodesic γ. The blue lines show the geodesic projections.

The aforementioned data visualization and dimensionality reduction methods provide a wide suite of techniques for working with high-dimensional data with various trade-offs among them. However, many of them are not apt for hyperbolic data, with HoroPCA being the only one specifically designed for hyperbolic dimensionality reduction. HoroPCA can create interesting visualizations, but like its Euclidean counterpart, is much better at maintaining global structure than local structure. Thus, there poses a need for a data visualization method for high-dimensional hyperbolic data. In the next chapter we discuss the necessary background about hyperbolic geometry and the details of t-SNE for the development of CO-SNE.

Chapter 3

Background

Our work focuses on extending t-SNE to hyperbolic space, so this chapter presents some background about hyperbolic geometry and t-SNE to develop CO-SNE.

3.1 Hyperbolic Geometry

Non-Euclidean Geometry

Hyperbolic geometry is a non-Euclidean geometry with negative curvature, as opposed to zero and positive curvature for Euclidean and spherical geometry, respectively. It is considered a non-Euclidean geometry as it satisfies all of Euclid's postulates except the fifth, which is also known as the Parallel Postulate [32].

1. A straight line segment can be drawn joining any two points.
2. Any straight line segment may be extended to any finite length.
3. A circle may be described with any given point as its center and any distance as its radius.
4. All right angles are congruent.
5. If two lines are drawn which intersect a third in such away that the sum of the inner angles on one side is less than two right angles, then the two lines inevitably must intersect each other on that side if extended far enough.

Euclidean geometry satisfies all five postulates. The Parallel Postulate was found difficult to prove, and eventually the negations of this postulate lead to various non-Euclidean geometries that were proven to be equally consistent as Euclidean geometry. An equivalent statement to the Parallel Postulate is that there is a triangle where the sum of the angles is 180°. Figure 3.1 visualizes this statement for Euclidean, spherical, and hyperbolic geometry.

Figure 1.1 is a visualization of another equivalent statement. Due to the different characteristics of hyperbolic space and the hyperbolic metric, hyperbolic geometry can be an useful alternative to Euclidean geometry. We next introduce Riemannian geometry, which is useful for describing hyperbolic space models.

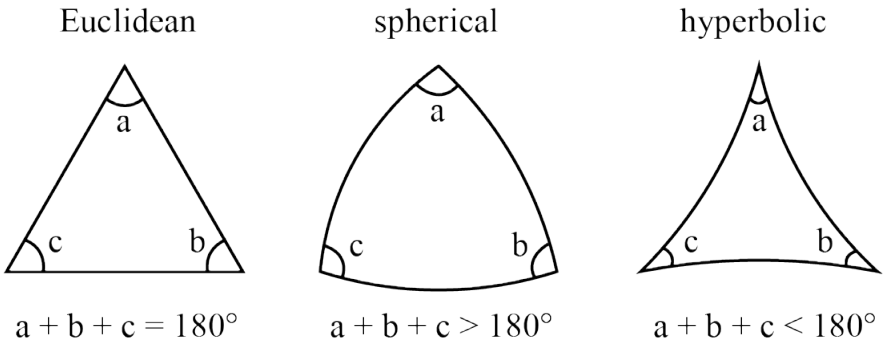


Figure 3.1: Triangles in Euclidean, spherical, and hyperbolic space which have 0, positive, and negative curvature, respectively. Only the Euclidean triangle satisfies the equivalent statement to the Parallel Postulate.

Riemannian Geometry

Riemannian geometry is a branch of differential geometry studying Riemannian manifolds which are smooth manifolds with a Riemannian metric. A hyperbolic space is a Riemannian manifold with constant negative curvature of -1, so equipping ourselves with some definitions important to Riemannian geometry will allow us to describe and work with the models for hyperbolic space presented below. Previous works [10, 32] provide some of these definitions, while [34] provides a comprehensive overview of Riemannian geometry.

An **n-dimensional manifold** $\mathcal{M}$ is a space that can locally be approximated by the Euclidean space $\mathbb{R}^n$. For example, the Earth is modeled by a sphere with positive curvature, but locally the surface of the Earth appears flat, like $\mathbb{R}^2$.

The **tangent space** of a point $x \in \mathcal{M}$ is the first order linear approximation of $\mathcal{M}$ around x and is denoted as $\mathcal{T}_x\mathcal{M}$. It is isomorphic to Euclidean space, meaning there is a structure preserving mapping between a tangent space and Euclidean space.

A **Riemannian metric** $\mathfrak{g}_x$ is a collection of smooth inner products on the tangent space associated with x — $\mathfrak{g}_x : \mathcal{T}_x\mathcal{M} \times \mathcal{T}_x\mathcal{M}$. The Riemannian metric gives rise to local notions of angles, length of curves, surface area, and volume [32]. Global quantities can be derived from these local values by integrating the local contributions [25].

A **Riemannian manifold** is then the tuple $(\mathcal{M}, \mathfrak{g})$, a manifold $\mathcal{M}$ that is equipped with a group of Riemannian metrics, $\mathfrak{g}$. Isometric models for hyperbolic space can be described in this form, with various manifold specifications and Riemannian metric tensors.

A **geodesic** is the generalization of a Euclidean straight line, and is the path of minimal length between two points. Two geodesics in hyperbolic space are visualized in pink in Figure 3.2

The **exponential map** $\exp_x$ of a manifold maps a vector $v \in \mathcal{T}_x\mathcal{M}$ in the tangent space of a particular point x, to a point on the manifold $\mathcal{M}$ by moving a unit length along the geodesic starting at x in the direction of v. The **logarithmic map** is the inverse to the exponential map, projecting points on the manifold into the tangent space of other points. Figure 3.2 show two examples of the exponential map in the Poincaré ball model. Note in the rest of this work, "exp" with no subscript refers to just the normal exponential operation, not the exponential map.

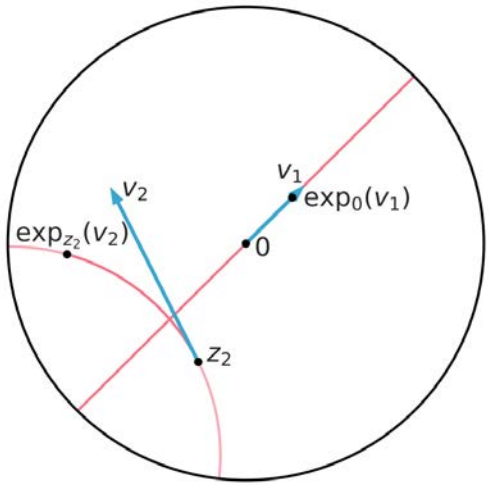


Figure 3.2: Two geodesics (pink) and two exponential mappings (blue) on the Poincaré disk, the 2D Poincaré ball.

With these definitions we can then detail models for hyperbolic space, including the Poincaré ball that is the focus in this work.

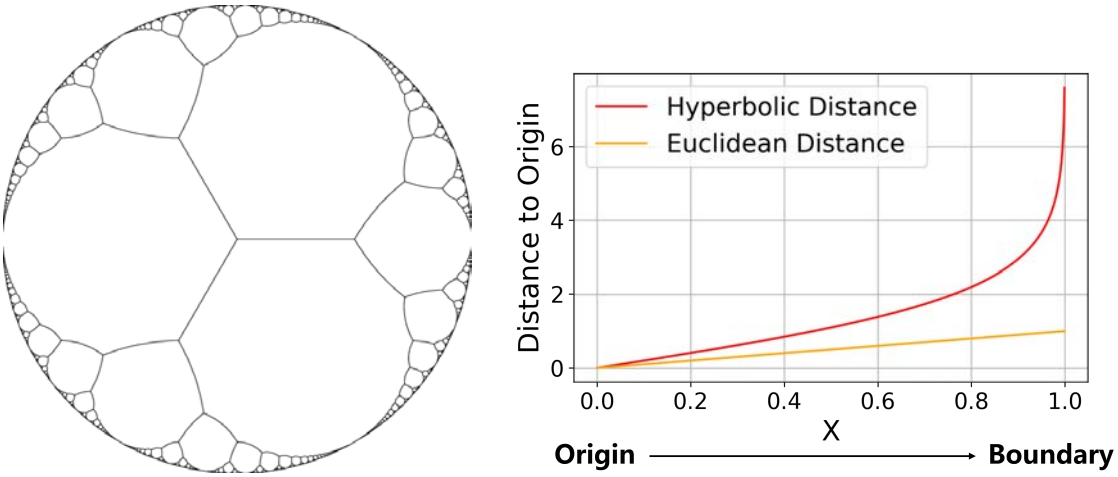
Poincaré Ball Model

In order to work with and visualize hyperbolic space, there are several isometrically equivalent models available. We choose the Poincaré ball model since it is the most commonly used one in hyperbolic representation learning [10, 29] and is suitable for visualizations. The n-dimensional Poincaré ball model is defined as $(\mathbb{B}^n, \mathfrak{g}_{\mathbf{x}})$, where $\mathbb{B}^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\}$

and $\mathfrak{g}_{\mathbf{x}} = (\gamma_{\mathbf{x}})^2 I_n$ is the Riemannian metric tensor. $\gamma_{\mathbf{x}} = \frac{2}{1-\|\mathbf{x}\|^2}$ is the conformal factor and I_n is the Euclidean metric tensor. Given two points $\boldsymbol{u} \in \mathbb{B}^n$ and $\boldsymbol{v} \in \mathbb{B}^n$, the hyperbolic distance between them is defined as,

$$d_{\mathbb{B}^n}(\boldsymbol{u}, \boldsymbol{v}) = \operatorname{arcosh}\left(1 + 2\frac{\|\boldsymbol{u} - \boldsymbol{v}\|^2}{(1 - \|\boldsymbol{u}\|^2)(1 - \|\boldsymbol{v}\|^2)}\right) \tag{3.1}$$

where arcosh is the inverse hyperbolic cosine function and $\|\cdot\|$ is the usual Euclidean norm. Compared to Euclidean distance, hyperbolic distance grows exponentially as we move the points towards the boundary of the Poincaré ball as seen in Figure 3.3. The exponential growth in volume is analogous to the exponential growth in nodes when moving from the root of a tree to the leaf level. As such, hyperbolic spaces can be see as the continuous counterpart to trees, leading to studies on their use in representing hierarchical data.



a) Poincaré disk tiling b) Distance growth comparision

Figure 3.3: a) Hyperbolic tiling of the Poincaré disk, the 2-dimensional Poincaré ball, which results in a tree structure [35]. b) Comparision of Euclidean distance and hyperbolic distance as we move towards the boundary of the Poincaré ball

.

Gyrovector Spaces and Operations

Hyperbolic neural networks [10] and many algorithms working with hyperbolic data will require vector arithmetic operations like vector addition, subtraction, and scalar multiplication. Gyrovector spaces are generalizations of Euclidean vector spaces to models of hyperbolic space via Möbius transformations. Ungar [39] introduces gyrovector spaces to study hyperbolic geometry, using them in a way analogous to vector spaces for Euclidean geometry. The aforementioned vector arithmetic operations are extended for hyperbolic space,

including vector addition, named Möbius addition:

$$\mathbf{u} \oplus \mathbf{v} = \frac{(1 + 2c\langle \mathbf{u}, \mathbf{v} \rangle + c\|\mathbf{v}\|^2)\mathbf{u} + (c\|\mathbf{u}\|^2)\mathbf{v}}{1 + 2c\langle \mathbf{u}, \mathbf{v} \rangle + c^2\|\mathbf{u}\|^2\|\mathbf{v}\|^2} \tag{3.2}$$

where c is the curvature, so $c = 0$ recovers the Euclidean addition operation. Hyperbolic distance (as in Equation 3.1) can be calculated using the Möbius addition as follows:

$$d_{\mathbb{B}^n}(\mathbf{u}, \mathbf{v}) = \frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c}\|-\mathbf{u} \oplus \mathbf{v}\|) \tag{3.3}$$

For the implementation of CO-SNE, this is how hyperbolic distances are calculated.

Riemannian Stochastic Gradient Descent

Parameters of hyperbolic algorithms that live in hyperbolic space such as learned hyperbolic embeddings or biases of hyperbolic layers of hyperbolic neural networks [10] cannot be optimized with standard gradient descent methods, as those generally operate on Euclidean data. Instead, Riemannian stochastic gradient descent [4] can be used. As a reminder, the standard stochastic gradient descent rule is:

$$\theta_{t+1} \leftarrow \theta_t - \lambda \nabla \mathcal{L} \tag{3.4}$$

where $\nabla \mathcal{L}$ is the gradient of the loss and λ is the learning rate. The update rule in Riemannian stochastic gradient descent is instead:

$$\theta_{t+1} \leftarrow \exp_{\theta_t}(-\lambda \nabla_R \mathcal{L}) \tag{3.5}$$

where $\exp \theta_t$ is the exponential map with respect to the current parameters and $\lambda \nabla_R \mathcal{L}$ is the Riemannian gradient. For the Poincaré ball model, the Riemannian gradient is a straightforward scaling of the standard Euclidean gradient. Since the exponential map is not always easy and sometimes inefficient to compute, a first order approximation called a retraction can be used in place of it. The Riemannian gradient descent update rule that we use is introduced in [29]. They use $\mathfrak{R}_\theta(v) = \theta + v$ as the retraction for the Poincaré ball model and additionally project the updated parameters like so:

$$\text{proj}(\theta) = \begin{cases} \theta/\|\theta\| - \epsilon & \text{if } \|\theta\| \geq 1 \\ \theta & \text{else} \end{cases} \tag{3.6}$$

with ϵ being a small constant for stability purposes. Putting it all together the Riemannian stochastic gradient descent update rule which we use in CO-SNE is:

$$\theta_{t+1} \leftarrow \text{proj}(\theta_t - \lambda \frac{(1 - \|\theta_t\|^2)^2}{4} \nabla \mathcal{L}) \tag{3.7}$$

We have now introduced the concept of non-Euclidean geometry and presented definitions of Riemannian geometry in order to develop the Poincaré ball model for hyperbolic geometry. We also acquainted the reader with the notion of gyrovector space and operations as well as Riemannian gradient descent for the purposes of extending the corresponding Euclidean concepts in t-SNE. The next section covers the specific algorithm of t-SNE in more detail, the moving parts of which will be replaced by the hyperbolic concepts we have covered.

3.2 t-SNE

Our method is based on t-SNE [40] which draws from Stochastic Neighbor Embeddding (SNE) [16]. SNE introduces the two step process which both our method and t-SNE replicate. First, distances in the high and low dimensional spaces are converted into probability distributions to model pairwise similarities between points. Then, a loss function is minimized, mainly a minimization of divergence between the probability distributions.

t-SNE begins by using high-dimensional distances between the input datapoints to generate the similarity values between them, forming a joint probability distribution. The joint probability p_{ij} represents the probability a point will pick another as its neighbor if neighbors are picked in proportion to the probability density of a distribution centered at that point. To define p_{ij}, t-SNE first defines the conditional probability $p_{j|i}$, the probability that the point $\mathbf{x}_i$ will pick a point $\mathbf{x}_j$ as its neighbor, using a normal distribution centered at the point $\mathbf{x}_i$. t-SNE then defines the joint probability distribution P, by setting $p_{ij} = \frac{p_{i|j}+p_{j|i}}{2m}$ as a way to increase the cost contribution of outlier points, where m is the number of high-dimensional datapoints. The conditional probability density $p_{j|i}$ is

$$p_{j|i} = \frac{\exp(-d(\mathbf{x}_i, \mathbf{x}_j)^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d(\mathbf{x}_i, \mathbf{x}_k)^2/2\sigma_i^2)} \tag{3.8}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. In the low-dimensional space, Student's t-distribution is used instead of a normal distribution (see Section 4.2) for modeling the joint probability distribution Q between embeddings, and q_{ij} is defined as

$$q_{ij} = \frac{(1 + d(\mathbf{y}_i, \mathbf{y}_j)^2)^{-1}}{\sum_{k\neq l}(1 + d(\mathbf{y}_k, \mathbf{y}_l)^2)^{-1}} \tag{3.9}$$

where $\mathbf{y}_i$ is the corresponding low-dimensional embedding of $\mathbf{x}_i$. Then, to best maintain the structures of the high-dimensional dataset, the cost function to minimize for embedding the low dimensional points is the Kullback-Leibler divergence between the probability distributions P and Q:

$$\mathcal{C} = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3.10}$$

The KL divergence is minimized through gradient descent with a couple of optimization tricks. They have an "early compression" loss in the early stages of training in order to

keep the points clusters close together for a certain amount of iterations. In this time, it is easier for the clusters to move around due to the lower distance between them. They also use "early exaggeration" which exaggerates the probabilities for the high-dimensional space distribution P in early iterations. This forces the low-dimensional embeddings further apart, creating more space in the map. The overall structure of the algorithm is given in in Algorithm 1. The standard t-SNE uses Euclidean distance and distributions not suited for hyperbolic space. Therefore, we makes some replacements and extensions to develop CO-SNE, as described in the next chapter.

Chapter 4

CO-SNE

This chapter presents CO-SNE by first introducing a direct extension of t-SNE that we call HT-SNE and then providing the explanation and details for adjustments and changes to create the final algorithm.

4.1 HT-SNE

Our first direct extension of t-SNE, HT-SNE, replaces a couple aspects of t-SNE with hyperbolic parallels to directly handle hyperbolic data. Algorithm 1 gives the main process for generating low-dimensional embeddings of t-SNE, which HT-SNE mirrors. To compute the pairwise affinities in high-dimensional space in line 1 of the algorithm, we replace the normal distribution of t-SNE with the hyperbolic normal distribution. Then, in line 4, we replace the Student's t-distribution in t-SNE with the hyperbolic Student's t-distribution. Additionally, the Euclidean gradient of t-SNE in line 5 is changed to the Riemannian gradient. Similar, the standard gradient descent update rule in line 6 is substituted for the Riemannian gradient descent update.

Algorithm 1: t-SNE and HT-SNE algorithm

Input: High-dimensional data, $\mathcal{X} = \{x_1, x_2, ..., x_n\}$
Result: Low-dimensional data embeddings, $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$
1 compute pairwise affinities of $\mathcal{X}$, p_{ij}, obtaining distribution P;
2 initialize low-dimensional embeddings $\mathcal{Y}^{(0)}$;
3 **for** t *in range* $(1, T)$ **do**
4 | compute pairwise affinities of $\mathcal{Y}^{(t-1)}$, $q_{ij}^{(t-1)}$, obtaining distribution $Q^{(t-1)}$;
5 | compute gradient of cost function based on P and Q;
6 | update $\mathcal{Y}^{(t)}$ using gradient descent update rule;
7 **end**

The hyperbolic normal and and hyperbolic Student's t-distributions are introduced below.

Hyperbolic Normal Distribution

To define the conditional probability in the high-dimensional hyperbolic space, we need to generalize the normal distribution to hyperbolic space. One natural generalization is called the Riemannian normal distribution which is the maximum entropy probability distribution given an expectation and a variance [33]. Given the Fréchet mean $\boldsymbol{\mu} \in \mathbb{B}_c^n$ and a dispersion parameter $\sigma > 0$, the Riemannian normal distribution is defined as,

$$\mathcal{N}_{\mathbb{B}^n}(\boldsymbol{x}|\boldsymbol{\mu}, \sigma^2) = \frac{1}{\boldsymbol{Z}} \exp(-\frac{d_{\mathbb{B}^n}(\boldsymbol{\mu}, \boldsymbol{x})^2}{2\sigma^2}) \tag{4.1}$$

where $\boldsymbol{Z}$ is the normalization constant. There are other generalizations of the normal distribution in hyperbolic space [25]. We choose to use the Riemannian normal distribution for simplicity. Thereafter, we refer to Riemannian normal distribution as hyperbolic normal distribution.

Hyperbolic Student's t-Distribution

One way to define the Student's t-distribution is to express the random variable t as,

$$t = \frac{u}{\sqrt{v/n}} \tag{4.2}$$

where u is a random variable sampled from a standard normal distribution and v is a random variable sampled from a χ^2-distribution of n degrees of freedom. In particular, t-SNE adopts a Student's t-distribution with one degree of freedom and the probability density function is defined as

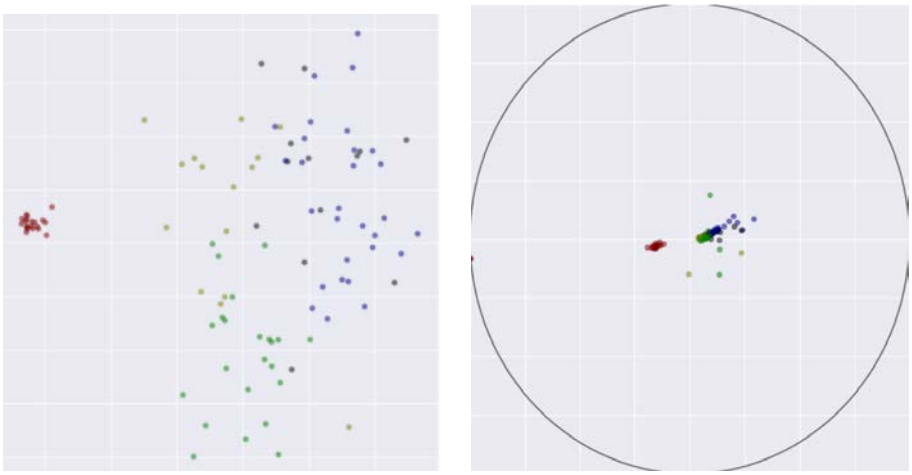
$$f(t; t_0) = \frac{1}{\pi(1 + (t - t_0)^2)} \tag{4.3}$$

To extend the Student's t-distribution to hyperbolic space, we derive the probability density function as

$$f_{\mathbb{B}^n}(t; t_0) = \frac{1}{\pi(1 + d_{\mathbb{B}^n}(t, t_0)^2)} \tag{4.4}$$

HT-SNE Results

Using the hyperbolic normal distribution to define conditional probabilities in high-dimensional space and the hyperbolic Student's t-distribution in the low-dimensional space, we obtain our first implementation of a hyperbolic extension of t-SNE, HT-SNE. However, as shown in Figure 4.1, we see that HT-SNE seems to suffer from something that appears to match the description of the "crowding problem" mentioned in [40], with points crowded near the center. Additionally, points in clusters have almost no separation between them. In the next section, we examine and address these problems.



a) t-SNE Visualization b) HT-SNE Visualization

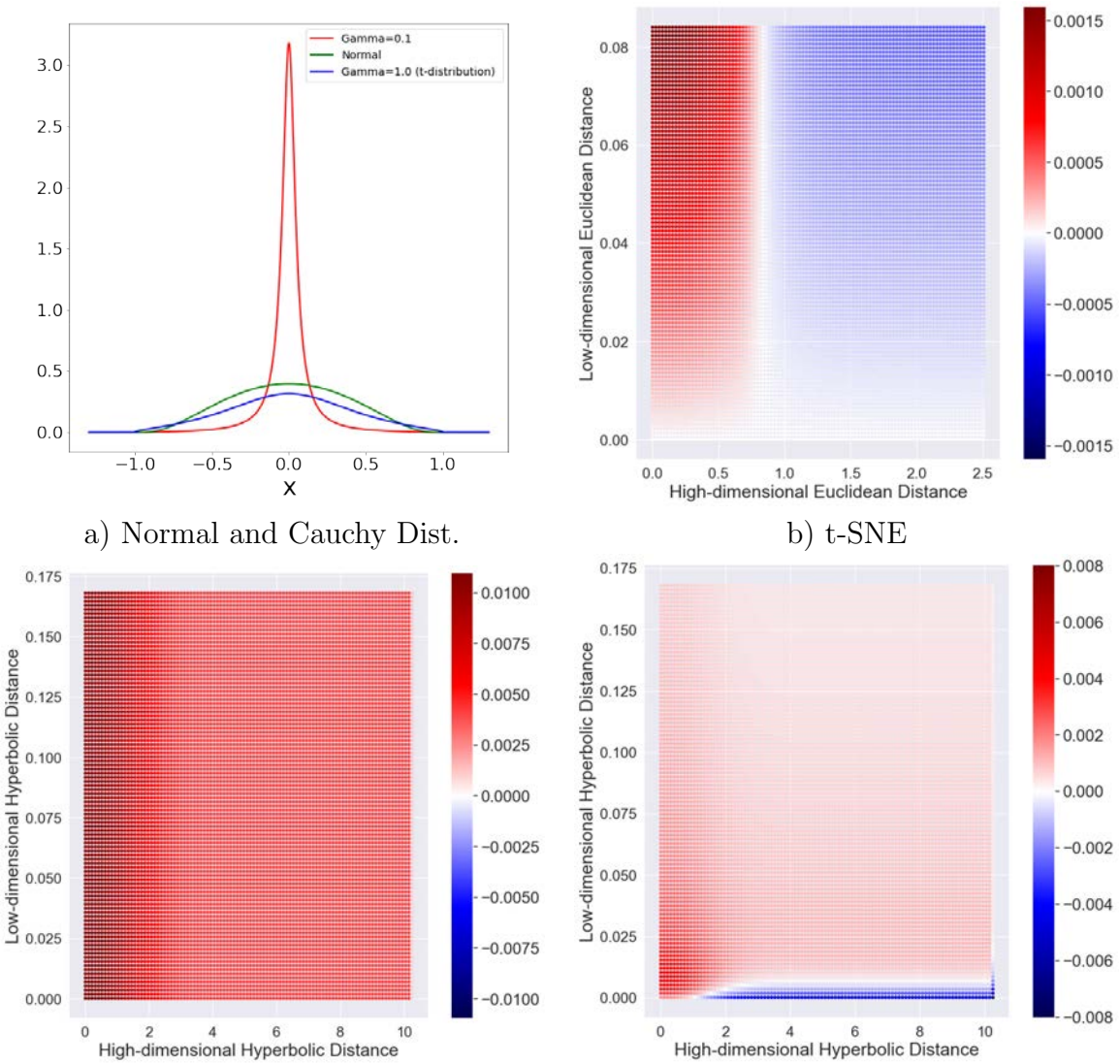
Figure 4.1: An initial result of HT-SNE. a) Clusters visualized by t-SNE have meaningful separation between points in the green, yellow, black, and blue clusters. b) Clusters visualized by HT-SNE show crowding, where the green, yellow, black, and blue clusters are crushed together near the center. The visualized data is five hyperbolic normal distribution clusters of dimension five.

4.2 From HT-SNE to CO-SNE

Addressing Crowding

One of t-SNE's main contributions is addressing the "crowding problem" [40] in SNE. The problem arises from the greater area available in high-dimensional space, which scales with r^n for an n-dimensional sphere. As such, as we move further from a particular point, the relative amount of space available in the low-dimensional space is much smaller. Consider then a particular cluster with many moderately distanced points around it. In high-dimensional space, these points can be spaced around in the moderately distanced area from the cluster. In low-dimensional space, there is not nearly as much volume in the moderated distanced area, so many of these points have to be placed very far away. They are so far away, that a large amount of small attractive forces act on them from the points in the cluster, pulling them towards the center. To address this, t-SNE uses the Student's t-distribution in the low-dimensional space to model distances as opposed to the normal distribution for high-dimensional space. The heavier tails of the Student's t-distribution allow for points to be placed further away in the low-dimensional map, since the larger probability in the tails makes them match up with the probability of moderately distanced points in the normal distribution.

We use the hyperbolic Student's t-distribution to mirror t-SNE's solution, but as seen



a) Normal and Cauchy Dist.

b) t-SNE

c) HT-SNE

d) CO-SNE

Figure 4.2: a) The probability density function of the hyperbolic normal distribution, hyperbolic Cauchy distribution with $\gamma = 0.1$ and hyperbolic Cauchy distribution with $\gamma = 1.0$. Hyperbolic Student's t-distribution (hyperbolic Cauchy distribution with $\gamma = 1.0$) does not have much heavier tails. b) The gradients of the t-SNE as a function of low-dimensional and high-dimensional Euclidean distance. c) The gradients of the HT-SNE as a function of low-dimensional and high-dimensional hyperbolic distance. d) The gradients of CO-SNE as a function of low-dimensional and high-dimensional hyperbolic distance. There is strong repulsion when dissimilar high-dimensional datapoints are projected close.

in the HT-SNE results, there is still crowding towards the center, as well as points in a cluster stacking together in a very small area. We first observe that the hyperbolic Student's t-distribution is does not have much heavier tails than the hyperbolic normal distribution. Then, we look into the gradient response between two points as a function of their low-dimensional distances and high-dimensional distances. Figure 4.2 visualizes this using Euclidean distance for t-SNE and hyperbolic distance for HT-SNE, with positive values signifying attraction and negative values signifying repulsion. It is clear that HT-SNE lacks repulsion forces that t-SNE has in this resolution, attracting even dissimilar high-dimensional points that are embedded close together in the low dimensional space. This suggests that rather than heavier tails in the low-dimension distribution, we actually require heavier peaks to encourage dissimilar points in the high-dimensional space to repel each other if they are placed close together in the low-dimensional space.

To alleviate this problem, we consider the hyperbolic Cauchy distribution, which has the probability density function

$$f(t; t_0, \gamma) = \frac{1}{\pi\gamma}[\frac{\gamma^2}{d_{\mathbb{B}^n}(t, t_0)^2 + \gamma^2}] \tag{4.5}$$

where γ is the scale parameter. Notice that the Student's t-distribution is a special case of the Cauchy distribution with $\gamma = 1.0$. With a small γ, the Cauchy distribution has a higher peak as seen in part a) of Figure 4.2. By using the Cauchy distribution with a small γ for the low-dimensional probability mapping, high dimensional points that are placed close together in the low-dimensional embedding space, but are dissimilar in the high-dimensional space, will incur a higher probability in the low-dimensional probability conversion than the high-dimensional conversion. This results in a repulsion force as the gradient (detailed later in Equation 4.8) will become negative.

Distance Loss

As mentioned before, hyperbolic space can aptly be used for embedding tree structures which are hierarchical in nature. The exponential growth in volume away from the origin of the Poincaré ball is analogous to the exponential growth in nodes away from the root of a tree. Therefore, the origin of the Poincaré ball is analogous to the root of a tree, and the points near the boundary are analogous to the leaf nodes. The depth or level of a node in the tree is then represented by the norm of a datapoint in the Poincaré ball. The cost function of t-SNE does not consider this structure, so we attempt to keep the norm invariant through adding an additional loss. This leads to the following loss function which minimizes the difference between the norms of the high-dimensional datapoint $\mathbf{x}_i$ and the corresponding low-dimensional embedding $\mathbf{y}_i$.

$$\mathcal{H} = \frac{1}{m}\sum_{i=1}^{m}(\|\mathbf{x}_i\|^2 - \|\mathbf{y}_i\|^2)^2 \tag{4.6}$$

With the distance loss, we can preserve the global hierarchy of the high-dimensional hyperbolic embeddings.

Optimization

The final criterion of CO-SNE is then composed of the KL-divergence for maintaining local similarity as seen in Equation 3.10 and the distance loss of Equation 4.6 for maintaining the global hierarchy:

$$\mathcal{L} = \lambda_1 \mathcal{C} + \lambda_2 \mathcal{H} \tag{4.7}$$

where λ_1 and λ_2 are hyperparameters. With this loss rather than just a KL divergence loss and the hyperbolic Cauchy distribution to model the low-dimensional probabilities, we move from HT-SNE (our more direct hyperbolic makeover of t-SNE) to arrive at our method which we call CO-SNE. A comparison of the t-SNE, HT-SNE, and CO-SNE in terms of metric of point similarity, distributions used to model point attraction, and loss function for optimization is presented in Table 4.1.

	Metric	High/low-dimensional Dist.	Losses
t-SNE	Euclidean	Normal/t-distribution	KL-div
HT-SNE	hyperbolic	hyperbolic normal/hyperbolic t-distribution	KL-div
CO-SNE	hyperbolic	hyperbolic normal/hyperbolic Cauchy	KL-div + Distance

Table 4.1: Our CO-SNE extends t-SNE by adopting hyperbolic normal distribution and hyperbolic Cauchy distribution. Compared with t-SNE, CO-SNE assumes the high-dimensional and low-dimensional space are hyperbolic. t-SNE cannot maintain the global hierarchy of the hyperbolic embeddings. Compared with HT-SNE, CO-SNE adopts hyperbolic Cauchy distribution and an additional distance loss. HT-SNE cannot push dissimilar high-dimensional points away in the low-dimensional space.

To train CO-SNE, the cost function is optimized with respect to the low-dimensional embeddings. The gradient of the KL-divergence $\mathcal{C}$ with respect to $\mathbf{y}_i$, a particular low-dimensional embedding, is given by

$$\begin{aligned} \frac{\delta \mathcal{C}}{\delta \mathbf{y}_i} &= \sum_j \frac{\delta \mathcal{C}}{\delta d_{\mathbb{B}^n}(\mathbf{y}_i, \mathbf{y}_j)} \frac{\delta d_{\mathbb{B}^n}(\mathbf{y}_i, \mathbf{y}_j)}{\delta \mathbf{y}_i} \\ &= 2 \sum_j (p_{ij} - q_{ij})(1 + d_{\mathbb{B}^n}(\mathbf{y}_i, \mathbf{y}_j)^2)^{-1} \frac{\delta d_{\mathbb{B}^n}(\mathbf{y}_i, \mathbf{y}_j)}{\delta \mathbf{y}_i} \end{aligned} \tag{4.8}$$

The partial gradient of the distance with respect to the low dimensional embedding $\mathbf{y}_i$ is given by

$$\frac{\delta d_{\mathbb{B}^n}(\mathbf{y}_i, \mathbf{y}_j)}{\delta \mathbf{y}_i} = \frac{4}{\beta \sqrt{\gamma^2 - 1}} \left(\frac{||\mathbf{y}_j||^2 - 2\langle \mathbf{y}_i, \mathbf{y}_j \rangle + 1}{\alpha^2} \mathbf{y}_i - \frac{\mathbf{y}_j}{\alpha} \right) \tag{4.9}$$

where $\alpha = 1-||\mathbf{y}_i||^2$, $\beta = 1-||\mathbf{y}_j||^2$, $\gamma = 1+\frac{2}{\alpha\beta}||\mathbf{y}_i-\mathbf{y}_j||^2$. We use the Riemannian stochastic gradient descent [4], detailed in Section 3.1, for the KL-divergence $\mathcal{C}$. The gradient of the distance loss $\mathcal{H}$ with respect to $\mathbf{y}_i$ is computed as

$$\frac{\delta\mathcal{H}}{\delta\mathbf{y}_i} = -4(\|\mathbf{x}_i\|^2 - \|\mathbf{y}_i\|^2)\mathbf{y}_i \tag{4.10}$$

We constrain the embeddings to the Poincaré ball after each update as in [29].

We find that the a two-stage training process produces the best results. In the first stage of 500 iterations, we only train with the local similarity loss of Equation 3.10. Then, we add the distance loss as in Equation 4.7. This seems to work better in practice as moving points locally when they are closer to the boundary is tough due to the exponentially growing distances away from the origin. With our method developed, we compare it with baselines and show visual results in the next chapter.

Chapter 5

Visualizing Hyperbolic Data with CO-SNE

In this chapter, we experiment with and present the results of CO-SNE on high-dimensional hyperbolic data. First, we compare the visualizations of CO-SNE on several hyperbolic datasets with other data visualization methods. Next, we show the ability of CO-SNE to visualize hierarchical semantic segmentation. Finally, we perform an ablation experiment to show the effects of adjusting the cost function hyperparameters and CO-SNE's robustness to hyperparameter selection.

5.1 Hyperbolic Data Visualizations and Comparisons

In this section, we test CO-SNE on several hyperbolic datasets and compare with 4 baseline algorithms:

- t-SNE [40]: this is the standard t-SNE which adopts Euclidean distance for computing the similarities in the high-dimensional space and the low-dimensional space. We use the t-SNE implementation from sci-kit learn [31] in our experiments and initialize the low-dimensional embedding using a normal distribution with mean 0.01 and unit variance. The training follows the standard setups as in [31]

- Principal Component Analysis (PCA) [18]: PCA is a commonly used dimensionality reduction method which attempts to maintain the maximal variation of the data. However, as a linear dimensionality method, PCA cannot reduce high-dimensional data to two dimensions in a meaningful way [2]. Again, we use the implementation from sci-kit learn [31], running with default hyperparameters.

- HoroPCA [5]: HoroPCA is a recently proposed extension of PCA on hyperbolic space. HoroPCA proposed to parameterize geodesic subspaces by ideal points in the Poincaré ball. HoroPCA can be used as a data whitening method for hyperbolic data and as

a visualization method as well. We use the implementation provided by the authors with default hyperparameters here: `https://github.com/HazyResearch/HoroPCA` .

- UMAP [26]: UMAP is a recently proposed dimensionality reduction and visualization method based on the ideas from Riemannian geometry and topology. UMAP is competitive with t-SNE and can better preserve the global structure of the data. We use the implementation provided in [27] with default hyperparameters.

We implement CO-SNE based on t-SNE by modifying the way of computing similarities and the optimization procedure as described in the previous chapter. We initialize the low dimensional embeddings the same as t-SNE above, using a normal distribution with mean 0.01 and unit variance. The scaling parameter for the hyperbolic Cauchy distribution, γ, is set to 0.1. The hyperparameter λ_1 is usually set to 10.0 and the hyperparameter λ_2 is usually set to 0.01.

We experiment on the 5 following datasets which are described in more detail in the following sections: 1) synthetic dataset sampled from a mixture of hyperbolic normal distributions, 2) biological dataset of cellular differentiation, 3) embedded hierarchical word taxonomies, 4) supervised hyperbolic neural network embeddings, and 5) unsupervised hyperbolic variational autoencoder embeddings.

Synthetic Point Clusters

We first use a synthetic dataset to validate the efficacy of CO-SNE. We randomly generate five point clusters of twenty points each in the five-dimensional hyperbolic space. Each cluster follows a hyperbolic normal distribution with the unit variance and the mean located on a different axis. The first and second means are close to the origin, at [0.1, 0, 0, 0, 0] and [0, -0.2, 0, 0, 0] respectively, whereas the third and fourth means are far from and equidistant to the origin, at [0, 0, 0.9, 0, 0] and [0, 0, 0, -0.9, 0] respectively. The last mean is right at the origin [0, 0, 0, 0, 0]. Figure 5.1 shows the 2D visualization results of these 5D points by different methods.

CO-SNE produces a much better visualization than the baselines. With CO-SNE, the projected two-dimensional hyperbolic embeddings can preserve both the local similarity structure and the global hierarchical structure of high-dimensional datapoints well. Also, CO-SNE can prevent the high-dimensional datapoints from being projected too close which usually happens with Euclidean distance based methods. Note that HT-SNE does not have enough repulsion between points, resulting in crowding and collapsing of clusters. The drawbacks of each baseline are as follows:

- Standard t-SNE: Euclidean distances are used for computing similarities in the high-dimensional space. Hyperbolic distances grow much faster than Euclidean distances. For high-dimensional hyperbolic datapoints which are close to boundary of the Poincaré ball, the standard t-SNE wrongly underestimates the distance between them. As a consequence, the standard t-SNE takes dissimilar high-dimensional data points as neigh-

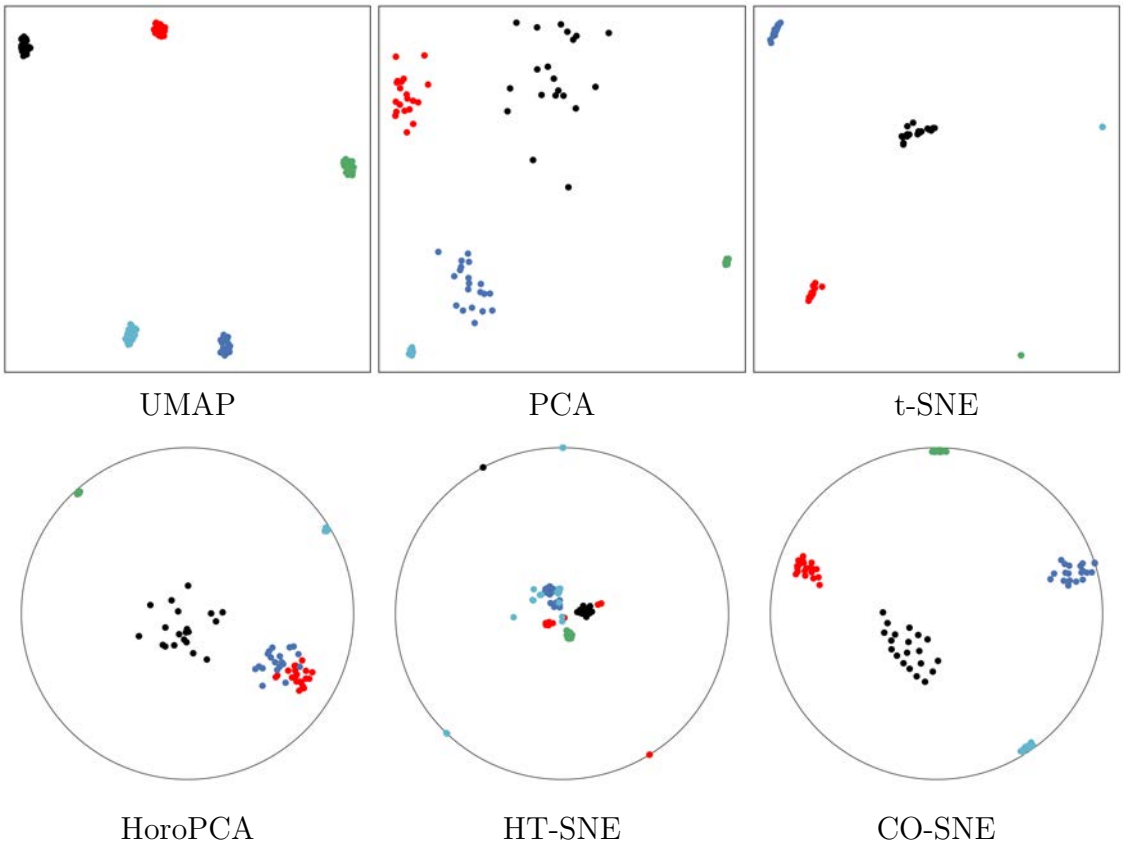
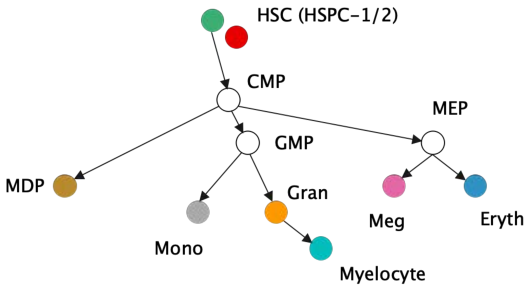


Figure 5.1: The projection of high-dimensional hyperbolic datapoints sampled from a mixture of hyperbolic normal distributions in a two-dimensional space with different methods. CO-SNE produces two-dimensional hyperbolic embeddings which preserve the hierarchical and similarity structure of the high-dimensional hyperbolic datapoints.

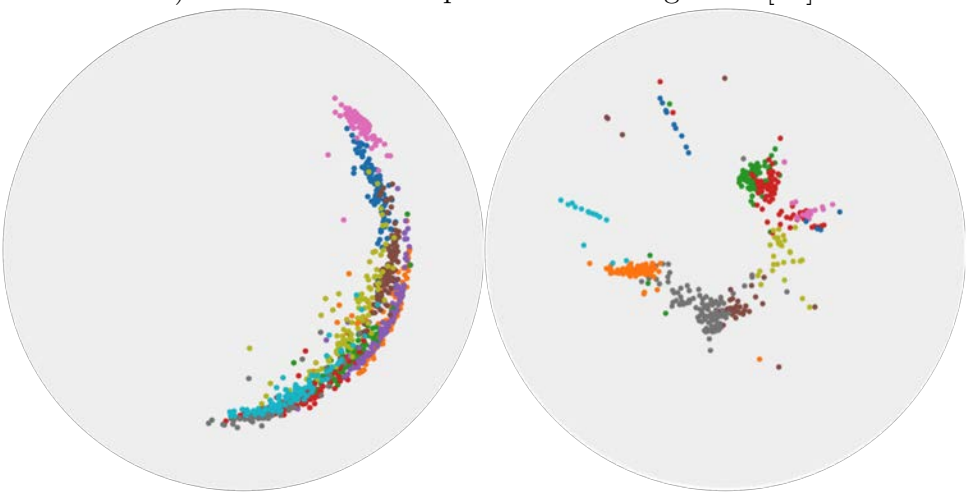
bors. The resulting low-dimensional embeddings collapse into one point which leads to poor visualization.

- PCA and HoroPCA: as mentioned above, PCA and HoroPCA are linear dimensionality reduction methods which are not generally suitable for visualization in a two-dimensional space. Both PCA and HoroPCA cannot preserve local similarity of the hyperbolic data.
- UMAP: UMAP suffers from the same issue as t-SNE since Euclidean distance is used for computing high-dimensional similarities.

For the rest of the results, we compare CO-SNE with HoroPCA since HoroPCA is the only baseline specifically designed for hyperbolic data.



a) Canonical hematopoietic cell lineage tree [22]



b) HoroPCA c) CO-SNE

Figure 5.2: Visualization of the high-dimensional biological datapoints of the mouse myelopoiesis dataset. a) The hierarchy of the original data from [22]. b) The two-dimensional embeddings generated by HoroPCA. c) The two-dimensional embeddings generated by CO-SNE. Both the local similarities and hierarchical structure is captured.

Biological Cellular Differentiation

Biological data can reveal naturally occurring hierarchies, such as in single cell RNA sequencing data [22]. In [22], they analyze cellular differentiation data, the transition of immature cells to specialized types. The immature cells can be viewed as the root of the tree and can branch off into several different types of cells, creating hierarchical data of cells in different states of progress in the transition process. One dataset we adapt from [22] is the mouse myelopoiesis dataset presented by [30], where there are 532 cells of 9 types. Two of the types, HSPC-1 and HSPC-2, form the root of the hierarchy, while megakaryocytic (Meg), erythrocytic (Eryth), monocyte-dendritic cell precursor (MDP), monocytic (Mono), and myelocyte (myelocytes and metamyelocytes) type cells are states father from the root. Granulocytic (Gran) cells are a precursor to myelocytes and multi-lineage primed (Multi-Lin) cells are in

an intermediate state.

The data has originally 382 dimensions (noisy) and like [22] we first reduce it to 20 dimensions via PCA to reduce the noises. We then scale the data to fit within the Poincaré ball and run CO-SNE and HoroPCA to produce two-dimensional hyperbolic embeddings. Noted that this dataset is not centered. The results are shown in Figure 5.2. The proposed low-dimensional embeddings by CO-SNE capture the hierarchical structure in the original data.

Hierarchical Word Embeddings

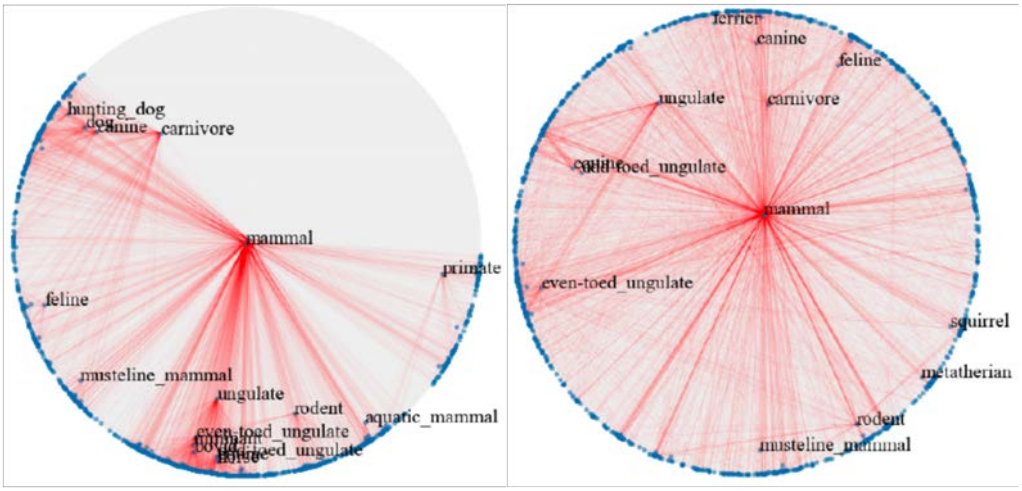
Hyperbolic space has been used to embed hierarchical representations of symbolic data. In [29], the authors adopt hyperbolic space for embedding taxonomies, in particular, the transitive closure of WordNet noun hierarchy [28]. As shown in [29], higher-dimensional hyperbolic embeddings often lead to better representations, but they are harder to visualize. Following [29], we embed the hypernymy relations of the mammals subtree of WordNet in hyperbolic space (more details in the embedding methodology is available in Section 5.2). We use the open source implementation provided by [29] to train the ten-dimensional embeddings. We use HoroPCA and CO-SNE to visualize the learned embeddings in a two-dimensional hyperbolic space.

Figure 5.3 shows that compared with HoroPCA, CO-SNE can better preserve the hierarchical and similarity structure of the high-dimensional datapoints. For example, the word *feline* and *canine* are close to *carnivore* in the CO-SNE embeddings, which is not the case in HoroPCA. The embeddings produced by CO-SNE also more resemble the two-dimensional embeddings as shown in Figure 2b of [29].

Features of Hyperbolic Neural Networks

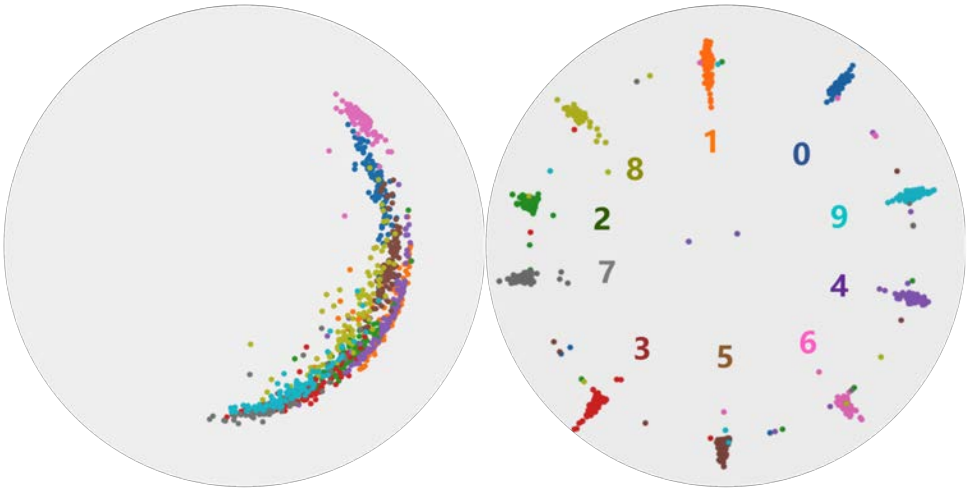
We next visualize the embeddings produced by hyperbolic neural networks for supervised image classification. First, we train a hyperbolic neural network (HNN) [10] with feature clipping [13] on MNIST, using clipping value of 1.0 and feature dimension of 64. Then, we use HoroPCA and CO-SNE to reduce the dimensionality of the features to two. The full test set of MNIST cannot be used due to the out-of-memory issue of HoroPCA. Therefore, we randomly sample 100 images for each class and visualize the embeddings.

Figure 5.4 shows the two-dimensional embeddings generated by HoroPCA and CO-SNE. The visualization produced by CO-SNE is significantly better the visualization produced by HoroPCA. In CO-SNE, the classes are well separated and have a clear hierarchical structure. We further train hyperbolic classifiers [10] on the frozen two-dimensional features. We use a learning rate of 0.001 and the number of epoch is 10. The classification accuracy of the features generated by HoroPCA is 30.2% while the accuracy of the features generated by CO-SNE is 61.2%. This implies that low-dimensional features generated by CO-SNE are more separated and respect the structure of original high-dimensional embeddings.



a) HoroPCA b) CO-SNE

Figure 5.3: Visualization of high-dimensional Poincaré word embeddings of the WordNet mammals subtree. a) The two-dimensional Poincaré word embeddings generated from HoroPCA. b) The two-dimensional Poincaré word embeddings generated from CO-SNE. In the embeddings generated by CO-SNE, the word *feline* and *canine* are close to *carnivore*, which is not the case in HoroPCA.



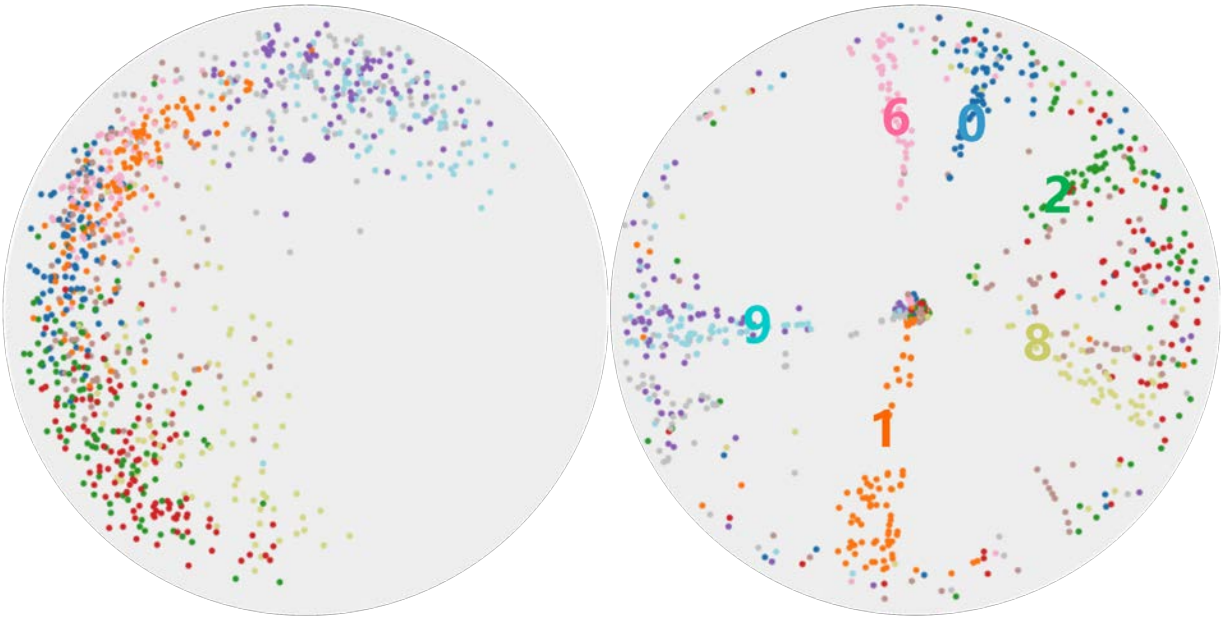
a) HoroPCA b) CO-SNE

Figure 5.4: CO-SNE produces better visualization of hyperbolic neural networks' (HNNs) features than HoroPCA. a) The visualization produced by HoroPCA. b) The visualization produced by CO-SNE. In CO-SNE, the classes are well separated and have a clear hierarchical structure.

Latent Space of Poincaré VAE

The variational autoencoder (VAE) [21] is a popular architecture for unsupervised learning. Standard VAEs assume the latent space is Euclidean. [25] extends standard VAEs by assuming the latent space is hyperbolic, creating Poincaré VAEs. Compared to standard VAEs, Poincaré VAEs can embed tree-like structures more efficiently. For using CO-SNE to visualize the latent space of Poincaré VAEs, we train a Poincaré VAE with a latent dimension of five on the MNIST dataset [23], following the procedure from [25]. We further generate the latent space representations of 1000 randomly sampled images with the encoder.

Figure 5.5 shows the visualizations produced by HoroPCA and CO-SNE. Clearly, CO-SNE produces much better visualization than HoroPCA. In particular, we can easily observe the hierarchical and clustering structures in the latent space which are totally distorted in the visualization produced by HoroPCA. Thus, CO-SNE can be used to understand the latent space of Poincaré VAEs and facilitate the development of better unsupervised hyperbolic learning methods.



a) HoroPCA b) CO-SNE

Figure 5.5: Visualization of the high-dimensional latent space representations generated by the encoder of the Poincaré VAE. a) The two-dimensional latent Poincaré embeddings generated from HoroPCA. b) The two-dimensional latent Poincaré embeddings generated from CO-SNE. CO-SNE can capture the hierarchical and clustering structures of the high-dimensional latent representations while HoroPCA cannot.

Discussion

Through our visualization experiments, we can see that CO-SNE is able to accomplish the two main features we would like to maintain when embedding high-dimensional hyperbolic data into low-dimensional space: the local similarities between points and their hierarchical global structure. Previous methods such as t-SNE [40] and UMAP [26] which use Euclidean distance to compute similarities are not able to visualize the global hierarchical structure. CO-SNE is able to do so by keeping the norm of the datapoints mostly invariant. Compared to HoroPCA [5], which does cater to hyperbolic data, CO-SNE is able to create visualizations that have better local similarity results for the most part.

CO-SNE does still suffer from some weaknesses just as the original t-SNE method does. The curse of dimensionality is still inevitable as it is impossible to fully represent the intrinsic structure of high-dimensional data in lower dimensions. The non-convex cost function leads to natural challenges in optimization and stopping rules. These weaknesses are visible in some of the visualizations. For example, there are some points that are far away from any other points as in Figure 5.4 and away from the cluster of their ground truth label that is not seen in the HoroPCA visualization. Despite some weaknesses, CO-SNE is overall a good visualization method for working with hyperbolic data, including features from hyperbolic learning methods. In the next section, we show the hierarchical nature of visual segmentation through [19] and CO-SNE.

5.2 Visualizing Unsupervised Hierarchical Semantic Segmentation

Hierarchical relations can be found in visual data, especially in the task of unsupervised semantic segmentation. In semantic segmentation, the goal is to assign a semantic category to every pixel in the given images. If supervision is available in the the form of pixel groupings, then the problem can be reduced to image recognition. If there are image level semantic category labels, then pixels labels can be predicted through classification. In unsupervised semantic segmentation, the problem is a matter of finding the groupings that best capture object-invariance and view-invariance of categories. Groupings have an intrinsic characteristic of granularity, which affects the resulting segmentation from the groups. At higher granularity, all the pixels on a human may be grouped together. Meanwhile, at lower granularity, groupings may be granular to each body part, such as a torso. A natural hierarchy can be formed between groupings as a less granular grouping may be made up of more fine-grained groups.

While many methods for unsupervised semantic segmentation do not address this ambiguity of granularity of groupings directly, [19] presents a hierarchical unsupervised segmentation method takes advantage of the notion of granularity. The authors perform hierarchical groupings at multiple levels of granularity, and improve the learned features using these hierarchies. To do this, they first learn pixel-level features. The most granular groupings

are then clusters of these features. From there, they create increasingly less granular groupings through further clustering. Additionally, they make use of clustering transformers that enforce consistency across levels of grouping granularity. The less granular groupings are formed by the merging of the previous level of more granular groupings. Figure 5.6 shows this process. This forms a natural hierarchy that can be embedded into hyperbolic space.

To generate the hyperbolic embeddings, we first extract the hierarchical relations from [19] by extracting granular feature clusters and their relations to the next level of clusters. Then, using [29], we embed these relations as into five-dimensional hyperbolic space. The main procedure of [29] is to minimize a loss function with respect to the generated embeddings $\boldsymbol{\theta}_i$ as so:

$$\Theta' \leftarrow \arg\min_{\Theta'} \mathcal{L}(\Theta) \quad \text{s.t.} \, \forall \, \boldsymbol{\theta}_i \in \Theta : \|\boldsymbol{\theta}_i\| < 1 \tag{5.1}$$

The constraint for $\boldsymbol{\theta}_i$ assures the embeddings to be within the Poincaré ball. The loss function to be optimized is task dependent. In this task, we use the same one as used for the WordNet transitive closure embeddings in [29], since the relations in both datasets form a directed acylic graph. The loss function is given as

$$\mathcal{L}(\Theta) = \sum_{(u,v)\in\mathcal{D}} \log \frac{\exp(-d_{\mathbb{B}^n}(\boldsymbol{\theta}_u, \boldsymbol{\theta}_v))}{\sum_{v'\in\mathcal{N}(u)} \exp(-d_{\mathbb{B}^n}(\boldsymbol{\theta}_u, \boldsymbol{\theta}_{v'}))} \tag{5.2}$$

where $\mathcal{D}$ contains the relations between the related objects (in our case, segments of various levels of granularity) and $d_{\mathbb{B}^n}$ is the hyperbolic distance function on the Poincaré ball as in Equation 3.1. $\mathcal{N}(u)$ is defined as $\mathcal{N}(u) = \{v|(u,v) \notin \mathcal{D}\} \cup \{u\}$ which is the set of negative examples for u where there are no relation connections. This loss is a soft ranking loss, moving related objects closer together than ones without. The objects that are low in the hierarchy, representing the leaf nodes, are thus spread apart within clusters that huddle close to the object in the next level of the hierarchy that they are close to. This naturally pushes the objects high in the hierarchy, the root nodes, closer to the center. For training, $\mathcal{N}(u)$ is formed by random sample of 10 negative examples. The Riemannian stochastic gradient descent method is used for optimization, using the rule in Equation 3.7.

We do the above procedure for images in the Pascal VOC2012 dataset [9], and embed their hierarchically organized segments in five-dimensional hyperbolic space. We then map these embeddings to two-dimensional space using CO-SNE. Figure 5.7 shows a visualization of the image from Figure 5.6 and its segments, plotting the segments in the Poincaré ball at their embedded two-dimensional location. CO-SNE is able to maintain the global hierarchical structure, with the higher hierarchical level, less granular segments closer to the origin than the lower hierarchical level, more granular segments. Furthermore, the most fine-grained objects (visualized as points in Figure 5.7, which are the first level clusterings of the pixel-level features, are clustered close together near their next level parent segment. This shows the ability for CO-SNE to maintain local similarities as well.

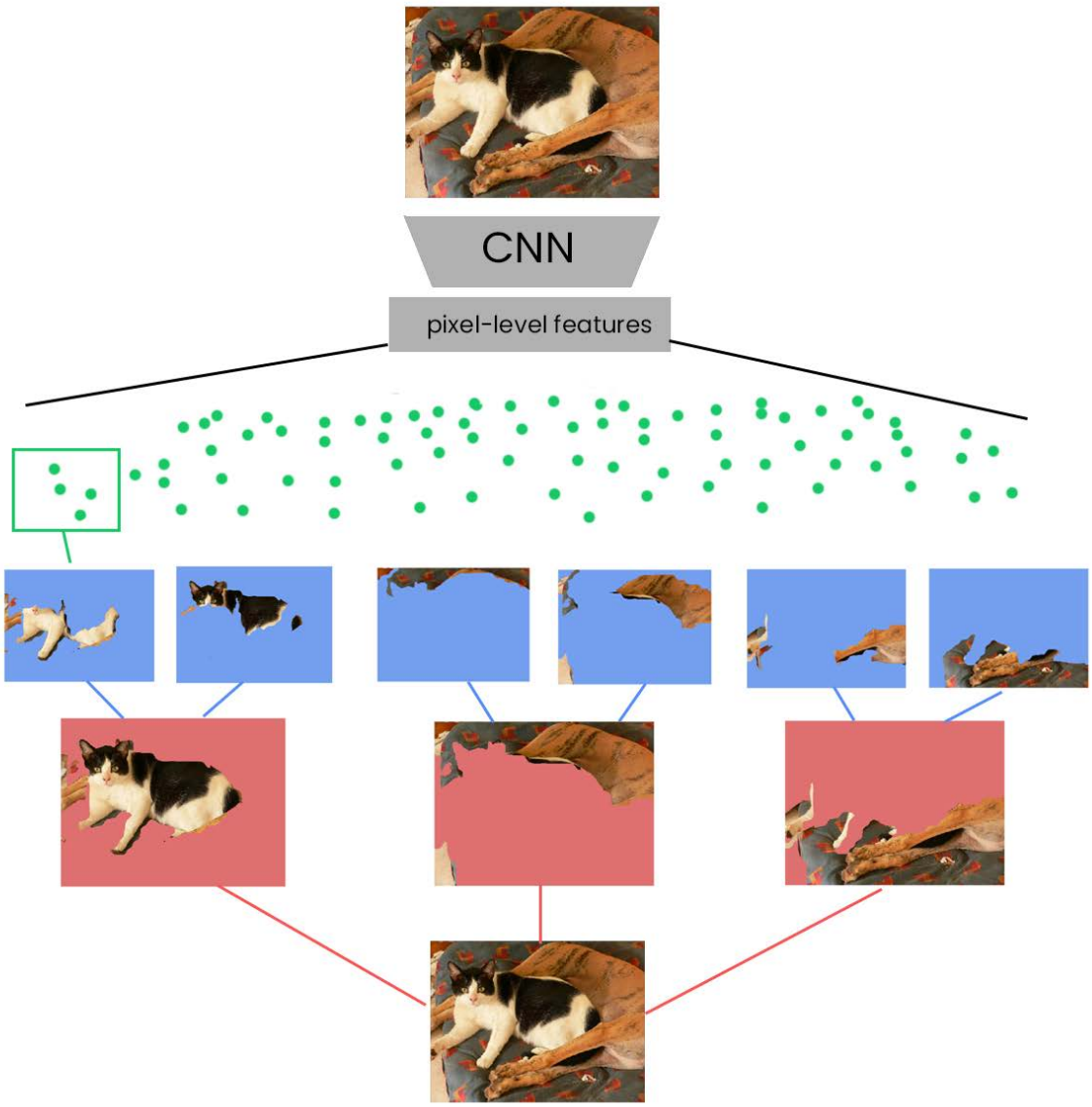


Figure 5.6: The generation of different levels of hierarchical groupings. The pixel-level features generated by a convolutional neural network are clustered into initial cluster (green dots), then into fine segments (blue), then coarse segments (red), which make up the whole image. The hyperbolic embedding of these various entities are visualized in Figure 5.7.

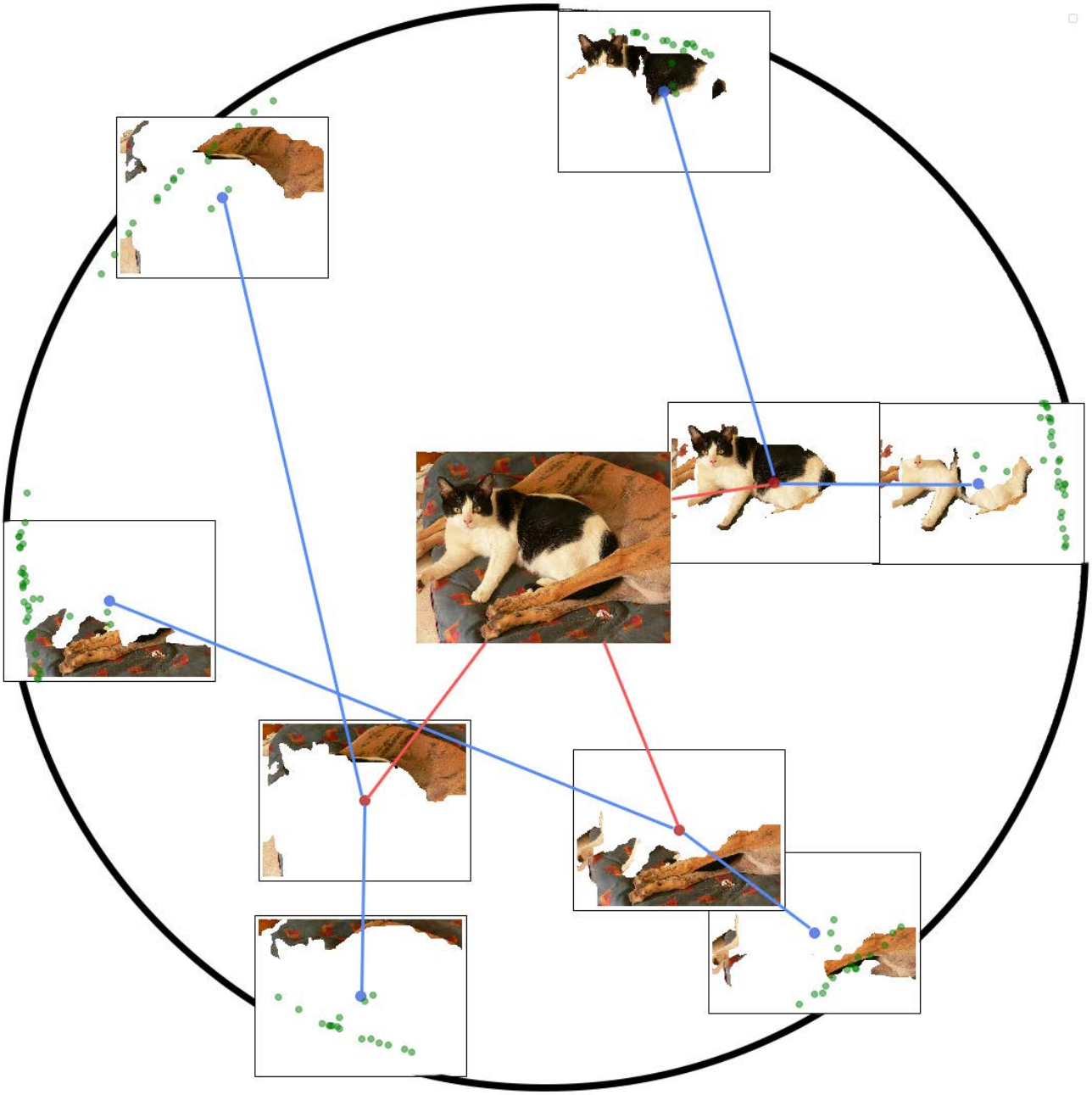


Figure 5.7: The various hierarchically leveled segments and clusters of a sample VOC2012 image, segmented by [19] and embedded into five-dimensional hyperbolic space via [29], and visualized by CO-SNE. The full image is placed at the center, with red lines connecting it to the most coarse image segments. Blue lines connect these most coarse segments to finer grained segments. The green dots near the blue segments are pixel-feature clusters, the most granular entity we visualize. The global hierarchy is clear, with the less granular elements a level closer to the origin than the most granular elements. The local similarity is also good at the most granular level, evidenced by the clusters of pixel-feature clusters (green dots).

5.3 Ablation Experiment

We perform an ablation experiment on the hyperparameters, λ_1 and λ_2, of the CO-SNE loss function given in Equation 4.7. λ_1 weights the KL divergence loss, given by Equation 3.10, between the pairwise affinity distributions. λ_2 weights the distance loss in Equation 4.6. Figure 5.8 shows the ablation results on a mixture of five clusters of hyperbolic normal distribution samples in five-dimensional hyperbolic space. The results show us how the two losses affect the generated low-dimensional embeddings in different ways and a general idea of the best hyperparameters to choose.

The top left corner in Figure 5.8 shows the initialization of the low-dimensional embeddings. The first column of visualizations shows the effect of the KL divergence loss on the embeddings without the distance loss. The local similarities seem to be maintained for the most part, but the global hierarchy is not, with all the points being pushed to the boundary in most of the visualizations. The first row then shows the effect of the distance loss without the KL divergence loss. The distance from the origin of the points is maintained, so we see that the global hierarchy is visualized with no local similarities. The combination of the two losses in the remaining figures are able to visualize both the local and global structures of the data. We see a trade off between the two hyperparameters in terms of local and global structures, but overall CO-SNE is fairly robust to the selection of these hyperparameters. One matter to note is that the magnitude of the gradient of the distance loss is much larger than magnitude of the gradient the KL divergence loss, so λ_1 should be larger than λ_2. We chose $\lambda_1 = 10$ and $\lambda_2 = 0.1$ for most of the visualizations in the previous sections.

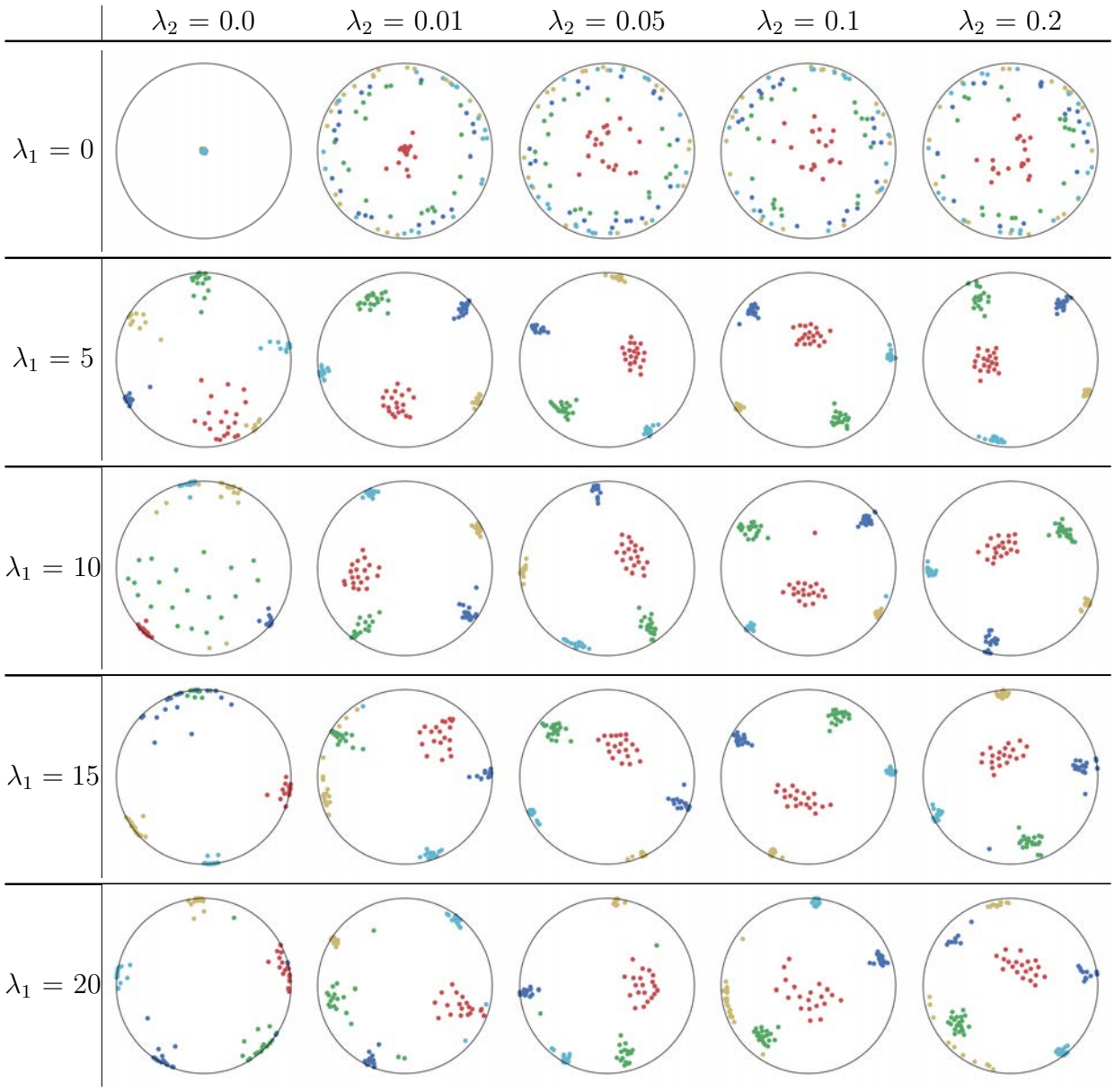


Figure 5.8: The effect of λ_1 and λ_2 visualizations of CO-SNE on a mixture of five hyperbolic normal distributions in a five-dimensional hyperbolic space. We can observe that λ_1 is responsible for preserving the local similarity structure and λ_2 is responsible for preserving the global hierarchical structure.

Chapter 6

Conclusion and Future Work

Hyperbolic geometry has gained traction for its use in embedding hierarchical data, with the hyperbolic metric being apt as a continuous approximation of the tree metric. High-dimensional hyperbolic representations can often outperform low-dimensional counterparts, but hyperbolic data lacks the solid visualization methods that are readily available for Euclidean data. In this work, we address this problem by introducing CO-SNE.

CO-SNE extends t-SNE to hyperbolic space and is able to maintain the local similarities as well as global hierarchical structure of high-dimensional hyperbolic data in low-dimensional space. Our development process begins with a direct extension of t-SNE to hyperbolic space, HT-SNE. To alleviate the shortcomings of HT-SNE, we substitute the hyperbolic Cauchy distribution to model similarities between the low-dimensional embeddings and introduce a distance loss. We demonstrate the effectiveness of CO-SNE by visualizing several relevant hyperbolic datasets such as real-life biological data and learned representations of hyperbolic neural networks. We also show the hierarchical nature of visual data by using CO-SNE to visualize the embedded hierarchies of unsupervised hierarchical semantic segmentation [19].

We hope that future work will be able to leverage CO-SNE as a tool in working with high-dimensional hyperbolic data. Visualizing features is an important step in tasks such as representation learning to gain a better understanding of the data and the method being developed. Hyperbolic geometry has already shown significant gains over Euclidean geometry in several previous methods [7, 10, 20, 24], especially on datasets with strong hierarchical structure. There is also room for further work on data visualization and dimensionality reduction of hyperbolic data. We discussed some weaknesses of CO-SNE in Section 5.1, which may be improved through exploration of different distributions to model similarities and other potential losses. Additionally, dimensionality reduction to dimensions greater than the 2 in the visualizations provided could be explored for hyperbolic data. HoroPCA [5] provides one example of this dimensionality reduction. Improvements in visualization and dimensionality reduction tools can in turn greatly improve our understanding of complicated learning systems and representations.

Bibliography

[1] Gregorio Alanis-Lobato, Pablo Mier, and Miguel A Andrade-Navarro. "Efficient embedding of complex networks to hyperbolic space via their Laplacian". In: *Scientific reports* 6.1 (2016), pp. 1–10.

[2] Sanjeev Arora, Wei Hu, and Pravesh K Kothari. "An analysis of the t-sne algorithm for data visualization". In: *Conference On Learning Theory*. PMLR. 2018, pp. 1455–1462.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[4] Silvere Bonnabel. "Stochastic gradient descent on Riemannian manifolds". In: *IEEE Transactions on Automatic Control* 58.9 (2013), pp. 2217–2229.

[5] Ines Chami et al. "HoroPCA: Hyperbolic Dimensionality Reduction via Horospherical Projections". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 1419–1429.

[6] Zachary Charles et al. "Convergence and margin of adversarial training on separable data". In: *arXiv preprint arXiv:1905.09209* (2019).

[7] Hyunghoon Cho et al. "Large-margin classification in hyperbolic space". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1832–1840.

[8] Andy Coenen and Adam Pearce. *Understanding UMAP*. URL: `https://pair-code.github.io/understanding-umap/`.

[9] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[10] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. "Hyperbolic neural networks". In: *arXiv preprint arXiv:1805.09112* (2018).

[11] Caglar Gulcehre et al. "Hyperbolic attention networks". In: *arXiv preprint arXiv:1805.09786* (2018).

[12] Yunhui Guo, Haoran Guo, and Stella Yu. "CO-SNE: Dimensionality Reduction and Visualization for Hyperbolic Data". In: *arXiv preprint arXiv:2111.15037* (2021).

[13] Yunhui Guo et al. "Free Hyperbolic Neural Networks with Limited Radii". In: *arXiv preprint arXiv:2107.11472* (2021).

[14] Mangesh Gupte et al. "Finding hierarchy in directed online social networks". In: *Proceedings of the 20th international conference on World wide web*. 2011, pp. 557–566.

[15] Marti A. Hearst et al. "Support vector machines". In: *IEEE Intelligent Systems and their applications* 13.4 (1998), pp. 18–28.

[16] Geoffrey E Hinton and Sam Roweis. "Stochastic neighbor embedding". In: *Advances in neural information processing systems* 15 (2002).

[17] Joy Hsu et al. "Learning Hyperbolic Representations for Unsupervised 3D Segmentation". In: *arXiv preprint arXiv:2012.01644* (2020).

[18] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.

[19] Tsung-Wei Ke et al. "Unsupervised Hierarchical Semantic Segmentation with Multiview Cosegmentation and Clustering Transformers". In: *arXiv preprint arXiv:2204.11432* (2022).

[20] Valentin Khrulkov et al. "Hyperbolic image embeddings". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6418–6428.

[21] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[22] Anna Klimovskaia et al. "Poincaré maps for analyzing complex hierarchies in single-cell data". In: *Nature communications* 11.1 (2020), pp. 1–9.

[23] Yann LeCun. "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/* ().

[24] Qi Liu, Maximilian Nickel, and Douwe Kiela. "Hyperbolic graph neural networks". In: *arXiv preprint arXiv:1910.12892* (2019).

[25] Emile Mathieu et al. "Continuous Hierarchical Representations with Poincar\'e Variational Auto-Encoders". In: *arXiv preprint arXiv:1901.06033* (2019).

[26] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[27] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *The Journal of Open Source Software* 3.29 (2018), p. 861.

[28] George A Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[29] Maximilian Nickel and Douwe Kiela. "Poincar\'e embeddings for learning hierarchical representations". In: *arXiv preprint arXiv:1705.08039* (2017).

[30] Andre Olsson et al. "Single-cell analysis of mixed-lineage states leading to a binary cell fate choice". In: *Nature* (2016).

[31] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[32] Wei Peng et al. "Hyperbolic deep neural networks: A survey". In: *arXiv preprint arXiv:2101.04562* (2021).

[33] Xavier Pennec. "Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements". In: *Journal of Mathematical Imaging and Vision* 25.1 (2006), pp. 127–154.

[34] Peter Petersen. *Riemannian geometry*. Vol. 171. Springer, 2006.

[35] François Sausset et al. "Bootstrap Percolation and Kinetically Constrained Models on Hyperbolic Lattices". In: *Journal of Statistical Physics* 138 (July 2009). DOI: 10.1007/s10955-009-9903-1.

[36] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. "Hyperbolic neural networks++". In: *arXiv preprint arXiv:2006.08210* (2020).

[37] David I Spivak. "Metric realization of fuzzy simplicial sets". In: *Preprint* (2009), p. 4.

[38] Abraham A Ungar. *Analytic hyperbolic geometry: Mathematical foundations and applications*. World Scientific, 2005.

[39] Abraham Albert Ungar. "A gyrovector space approach to hyperbolic geometry". In: *Synthesis Lectures on Mathematics and Statistics* 1.1 (2008), pp. 1–194.

[40] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[41] Melanie Weber et al. "Robust large-margin learning in hyperbolic space". In: *arXiv preprint arXiv:2004.05465* (2020).

[42] Zhenzhen Weng et al. "Unsupervised Discovery of the Long-Tail in Instance Segmentation Using Hierarchical Self-Supervision". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2603–2612.