Accelerating hyperbolic t-SNE in the Klein Disk model

Accelerating hyperbolic t-distributed Stochastic Neighbourhood Embedding approximation using a polar quadtree in the Klein Disk model

Jochem Lippes

Supervisor: Martin Skrodzki[1], Responsible Professor: Elmar Eisemann[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Jochem Lippes
Final project course: CSE3000 Research Project
Thesis committee: Martin Skrodzki, Elmar Eisemann, Gosia Migut

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

In this work we aim to implement a variaton of the acceleration of hyperbolic t-SNE done by Skrodzki et. al. [19]. This variation aims to embed the points in the Klein Disk model of hyperbolic space instead of the Poincaré Disk model using an altared version of a polar quadtree to speed up the computation in a similar fashion as the Barnes-Hut scheme for the Euclidean versino of t-SNE. We analyze our results to prove our acceleration works for the Klein Disk model and compare the efficiency of our implementation to the one for the Poincaré model in terms of quality of results and runtime.

1 Introduction

Analysis of high-dimensional data has high significance across various industries and research fields. In the process of high-dimensional analysis dimensionality reduction plays a vital role in both visualising as well as analyzing such data. This can be seen by the application of such reductions in fields such as healthcare [2], literature search [12], sports [25], targeted advertising [3] and machine learning [20]. An effective representation of this high-dimensional data in 2 - or 3 dimensional space preserves the underlying structure and thus facilitates insights through visual examination of the low-dimensional representation.

A commonly used technique for the embedding of high-dimensional data is t-distributed stochastic neighbor embedding (t-SNE). T-SNE is particularly good at preservation of local neighborhoods in the low-dimensional representation of the dataset [23]. Like most traditional dimensionality embedding algorithms (regular) t-SNE embeds the high dimensional data into flat, Euclidean space, which misses out on the potential benefits provided by other embedding spaces, one group of such spaces are the negatively curved, hyperbolic spaces [17]. Due to the negative curvature of these hyperbolic spaces geometric structures behave differently. For example, the area and circumference of circles in 2 dimensional hyperbolic space do not grow polynomially with regards to their radius like they do in Euclidean geometry, but instead they grow exponentially [9]. This geometric property makes hyperbolic space very suited to embed data into that also shows exponential growth, such as trees. Previous work has already had success in showing performance improvements in targeted advertisements systems by embedding data into hyperbolic space [18]. Hyperbolic space also provides a natural *Focus+Context* view of the data [10], which improves information foraging [15].

These benefits have led to multiple proposed methods to alter t-SNE to embed into hyperbolic space instead of Euclidean space [27], [6], [7]. Although these alterations of t-SNE have shown potential in their respective use cases, they all have to solve very costly optimization problems when compared to methods that embed into Euclidean space. This is mostly due to the fact that for the Euclidean case there exist multiple acceleration methods to speed up the computation [22], [21], [11], [14]. One of these methods is the Barnes-Hut scheme [22], it uses a quadtree data structure using equally sized quadrilaterals to form the nodes of the tree, it then uses the average of the containing points in each node as acceleration proxies. Like the other acceleration methods Barnes-Hut relies on averages and linear interpolation to speed up the computation, which hyperbolic space lacks due to the earlier mentioned exponential growth. As such there is no trivial way to translate these methods to the hyperbolic setting.

Recently however a first acceleration data structure for hyperbolic t-SNE has been proposed [19]. This acceleration method embeds high-dimensional data into the 2 dimensional Poincaré disk model of hyperbolic space [17]. It uses a modified version of Barnes-Hut with a polar quad-tree designed to operate in hyperbolic space to accelerate the computation. The paper proposing this model shows promising results for the speed up of hyperbolic t-SNE, however it does not explore whether the Poincaré Disk model is the most suitable model of hyperbolic space for this acceleration.

One of the other model for hyperbolic space is the Klein Disk model [17]. In this model geodesics, straight lines representing the shortest path between 2 points, are straight lines contained within the unit circle, unlike the the Poincaré Disk model where they are either circular arcs within the unit circle, or diameters of the unit circle. One of the drawbacks of the Klein model is that, unlike the Poincaré Disk model, it is non-conformal, meaning that angles and circles are distorted. [17] This paper mains goal will be to uncover how the performance, in terms of computational efficiency and quality of results, of the acceleration of hyperbolic t-SNE using a polar quadtree designed for the Poincaré Disk model compare to an implementation of a polar quadtree designed for the Klein Disk model.

We will do this by creating an implementation of the hyperbolic t-SNE acceleration using the Barnes-Hut scheme for the Klein model. Then comparing the runtime and quality of results of this implementation to the one for the Poincaré Disk model.

2 Background

This section will contain an overview of the relevant background that our research is based upon. Notably we will start with a discussion on hyperbolic space and more specifically the Klein Disk model. Secondly we will discuss t-distributed Stochastic Neighbour Embedding in both the Euclidean and hyperbolic space. Lastly we will discuss the Barnes-Hut acceleration method and how it has been adapted for hyperbolic t-SNE in the Poincaré Disk model.

2.1 Hyperbolic space and the Klein Disk model

When working with hyperbolic space it is first needed to choose one of the existing models like the Poincaré Disk, Klein Disk, upper half-plane or the Lorentz Hyperboloid [17] that suits the needs of the use-case. The paper our work is based upon chose to use the Poincaré Disk model of hyperbolic space, however we will adapt their implementation to instead use the Klein Disk model.

The Klein Disk model, like the Poincaré Disk model, embeds the entire hyperbolic space in the Euclidean unit circle. However the geodesics (straight lines) of the Klein Disk model

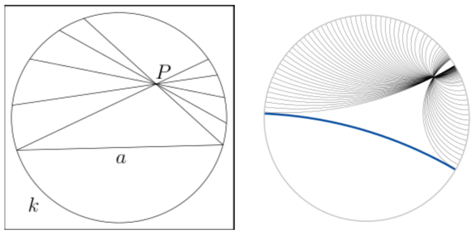

Figure 1: On the left straight lines in the Klein Disk model not intersecting with line a, on the right straight lines in the Poincaré Disk model not intersecting the blue line

are different than in the Poincaré Disk model. In the Klein Disk geodesics are represented by chords of the unit circle. This means that in the Euclidean sense the lines still appear straight, unlike the Poincaré Disk where geodesics are arcs, see fig 1 for a comparison between the two. What the Klein model wins in intuitivity by hyperbolic straight lines appearing straight in the Euclidean sense, it loses in the representation of angles and circles, as the Klein Disk model is not conformal these are distorted. Formally, the Klein Disk is the space $\mathbb{D} = \left\{ y \in \mathbb{R}^2 \,\middle|\, \|y\| < 1 \right\}$. To determine the hyperbolic distance between two points in the Klein model we use the Klein metric which is defined as follows; If you have two points P and Q to calculate the distance between P and Q first draw the chord of the unit circle through these two points. Find the two points where this chord intersects the unit circle, A and B, these are called the ideal points. Label the points such that they are A P Q B from left-most to right-most on the line, see fig 2, then the hyperbolic distance is defined by

$$d^{\mathcal{H}}(P,Q) = \frac{1}{2} \ln \frac{\|Q-A\| \, \|B-P\|}{\|P-A\| \, \|B-Q\|}. \tag{1}$$

This can be rewritten to

$$D^{\mathcal{H}}(p,q) = \sinh^{-1}\left(\frac{\sqrt{|(D_{pq})^2 - (A_{pq})^2|}}{\sqrt{1-(D_p)^2} \cdot \sqrt{1-(D_q)^2}} \right) \tag{2}$$

where D_{pq} is the Euclidean distance between the two points,

$$D_{pq} = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2},$$

$$D_p = \sqrt{x_p^2 + y_p^2}$$

and

$$D_q = \sqrt{x_q^2 + y_q^2}$$

are the Euclidean distances from the origin to the two points respectively, and

$$A_{pq} = x_p y_q - x_q y_p.$$

2.2 T-distributed Stochastic Neighbour Embedding

T-SNE is a dimensionality reduction technique that aims to embed high-dimensional data into low dimensional space while trying to preserve local neighbourhoods of the high-dimensional space [23]. It tries to achieve this by viewing

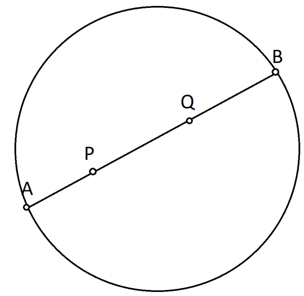

Figure 2: A figure showing the way we construct the points for the distance in the Klein Disk model

the high dimensional data input $\{x_1, ..., x_n\} \subseteq \mathbb{R}^d$ as (conditional) probabilities that are given by the formula

$$p_{j|i} = \frac{\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{2\sigma_i^2}\right)}, \quad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2}, \tag{3}$$

where $P_{i|i} = 0$ and σ_i is the variance of the Gaussian centered on point $\mathbf{x_i}$.

Besides the high-dimensional probability t-SNE also uses a corresponding probability distribution for the low-dimensional embedding $Q = \{y_1, ..., y_n\} \subseteq \mathbb{R}^{d'}$, this distribution is given by

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq \ell} (1 + \|y_k - y_\ell\|^2)^{-1}}. \tag{4}$$

To perform the embedding t-SNE begins by creating a set of low-dimensional embedded points using principal component analysis (PCA) [8]. It then alters these points using gradient descent to optimize the Kullback-Leibler divergence between the high- and low-dimensional probabilities. This divergence is given by

$$C = KL(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{5}$$

the corresponding gradient is given by

$$\frac{\delta C}{\delta \mathbf{y}_i} = 4 \sum_{j \neq i} (p_{ij} - q_{ij}) \left(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2\right)^{-1} (\mathbf{y}_i - \mathbf{y}_j). \tag{6}$$

Without any acceleration methods t-SNE has a run time complexity of $\mathcal{O}(n^2)$. This can be seen when analyzing the following rewritten version of Eq. 6

$$\frac{\delta C}{\delta \mathbf{y}_i} = 4 \left(\sum_{j \neq i} p_{ij} q_{ij} Z(\mathbf{y}_i - \mathbf{y}_j) - \sum_{j \neq i} q_{ij}^2 Z(\mathbf{y}_i - \mathbf{y}_j) \right), \tag{7}$$

where $Z = \sum_{k \neq \ell} (1 + \|\mathbf{y}_k - \mathbf{y}_\ell\|^2)^{-1}$. Analyzing this equation we see that, if the Guasians in Eq. 3 are truncated, it can be computed efficiently [22]. The second part of this equation however can not be computed efficiently and leads to the $\mathcal{O}(n^2)$ runtime.

2.3 Hyperbolic t-SNE

To adapt t-SNE to embed into hyperbolic space changes are needed to the essential equations that make up the algorithm. Firstly, the high dimensional probability distribution P_{ij} is still given by Eq. 3. The low-dimensional probabilities however do change. As the space we are embedding into is hyperbolic space instead of Euclidean space the geometric calculations within this space need to be adapted as well. Instead of using the Euclidean distances we have to use the hyperbolic distances. As such the low-dimensional probability distribution q_{ij} is now given by

$$q_{ij}^{H} = \frac{(1 + (d_{ij}^{H})^2)^{-1}}{\sum_{k \neq \ell}(1 + (d_{ij}^{H})^2)^{-1}}, \tag{8}$$

where d_{ij}^{H} is the hyperbolic distance between points i and j. As the gradient descent also relies on Euclidean distances in the embedding space it will also differ when we want to embed into hyperbolic space. The rewritten version of the gradient from Eq. 7 in the hyperbolic case becomes

$$\frac{\delta C^{\mathcal{H}}}{\delta \mathbf{y}_i} = 4\left(\sum_{j \neq i} p_{ij} q_{ij}^{\mathcal{H}} Z^{\mathcal{H}} \frac{\delta d_{ij}^{\mathcal{H}}}{\delta \mathbf{y}_i} - \sum_{j \neq i} \left(q_{ij}^{\mathcal{H}}\right)^2 Z^{\mathcal{H}} \frac{\delta d_{ij}^{\mathcal{H}}}{\delta \mathbf{y}_i}\right), \tag{9}$$

where $Z^{\mathcal{H}} = \sum_{k \neq \ell}\left(1 + \left(d_{ij}^{\mathcal{H}}\right)^2\right)^{-1}$.

When performing gradient descent it is needed to take two extra steps that are not present in the Euclidean version of t-SNE. Firstly, when performing the gradient descent-step to change a point $\mathbf{y}_i$ it has to be ensured that it goes along a geodesic of the model of hyperbolic space that it is acting in, that is, it is straight in the Hyperbolic sense. For example, in the Poincaré Disk model this means to account for the curvature of the geodesics in this model, see [19] for an example of this step for the Poincaré Disk model. The second extra step is to ensure that after a gradient descent-step the points are still within the bounds of the hyperbolic space of the model they are embedded in. For both the Poincaré - and Klein Disk models this means that after each step it is needed to ensure that points are still confided by the unit circle, thus after each step, for each point, the following projection is performed

$$proj(y_i) = \begin{cases} y_i/\|y_i\| - \epsilon & \text{if } \|y_i\| \geq 1 \\ y_i & \text{otherwise} \end{cases}, \tag{10}$$

With these changes it is possible to embed into the hyperbolic space using t-SNE. However, like in the Euclidean space, naive approaches to the optimization problem given by Eq. 9 still have a runtime of $\mathcal{O}(n^2)$, as the second part of the sum will take quadratic calculations in the hyperbolic case also.

2.4 Barnes-Hut acceleration for t-SNE

One of the proposed acceleration methods for t-SNE is the Barnes-Hut scheme. Barnes-Hut uses a quadtree data structure build in the embedded space, sometimes also called a Barnes-Hut tree. In this data structure the nodes consist of equally sized quadrilaterals containing the embedded points. It speeds up the computation of the second part of the sum in Eq. 7 by replacing it with a summary of a group of points

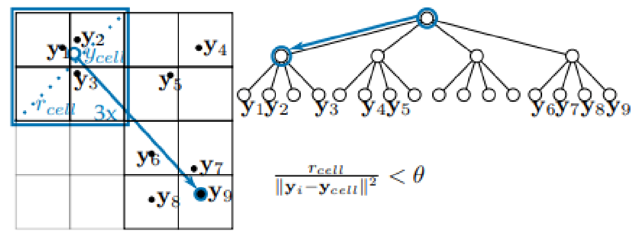

Figure 3: Visualization showcasing how the Barnes-Hut tree works. On the left we see that the points grouped by the top left cell of the quadtree are sufficiently far away from query point y_9 and as such will be summarized using the midpoint of the cell y_{cell}

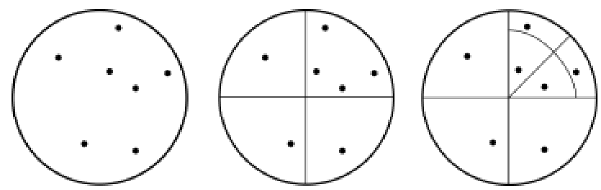

Figure 4: A showcase on how the building of a polar quadtree works, it starts out with the full circle, then splits into 4 slices, and then splits one of them along the angular and radial direction. [19]

in a quadtree node when the midpoint of these points is sufficiently far away from the querying point $\mathbf{y_i}$. To do this, when calculating the gradient for each point we traverse the quadtree and check for each node whether the following inequality holds

$$\frac{r_{cell}}{\|\mathbf{y}_i - \mathbf{y}_{cell}\|} < \theta, \tag{11}$$

where r_{cell} is the length of the diagonal of the quadtree cell (node), y_{cell} is the arithmetic midpoint of the grouped points contained in the quadtree cell. See fig. 3 for a visualization of this. θ is a parameter set by the user, usually within a range of 0.2 and 0.8 [22], that steers the approximation. If Eq. 11 holds we do not do traversal of its children and summarize the points contained in the cell for the evaluation of the gradient. It does this by weighing the midpoint y_{cell} by the amount of points contained within the quadtree cell.

2.5 Polar quadtree

A regular quadtree would not work in Hyperbolic spaces that are contained within the unit disk, like the Klein - and Poincaré Disk models. This is because quadrilaterals of the quadtree can contain areas of the space that are beyond the boundaries of the unit disk, which are undefined in these models. One translation of a quadtree adjusted for these models is the polar quadtree [24]. For this polar quadtree the root cell is not a rectangle that contains all points, which is the case for the Euclidean quadtree, instead it is a circle containing all input points in the embedded space. It then forms the polar quadrilaterals that make up the nodes of the polar quadtree by splitting along the angular and radial directions. Denote ϕ as the angular direction, the splitting of a cell is done at $\text{mid}_\phi = (\max_\phi + \min_\phi)/2$, where $\max_\phi$ denotes the maxi-

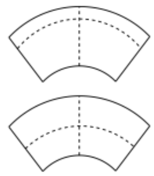

Figure 5: Depection of the 2 different splitting criteria, top shows if we split a quadrilateral by equal (hyperbolic) area, bottom shows if we split by equal length

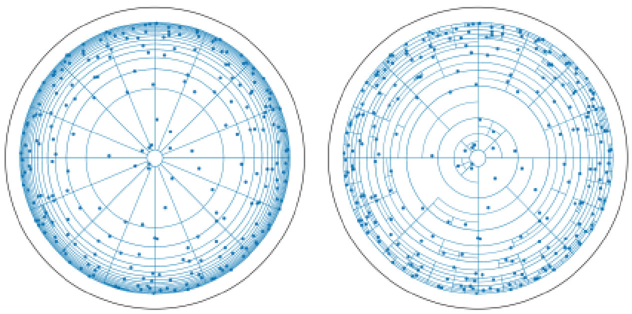

Figure 6: Visualization of the effect of splitting choice, on the left equal area and on the right equal length.

mum - and $\min_\phi$ the minimum angular value of the current cell. To ensure each of the four sub-cells have equal area a split in the radial direction r is performed at

$$\text{mid}_r = acosh\left(\frac{\cosh \max_r + \cosh \min_r}{2}\right). \tag{12}$$

See fig. 4 for a visualization of this process. In [24] it is shown the insertion time for a new point is $\mathcal{O}(\log n)$ where n denotes the amount of nodes present in the tree.

2.6 Barnes-Hut for hyperbolic t-SNE in the Poincaré Disk model

The acceleration data structure proposed in [19] adapts the Barnes-Hut tree to a modified version of a polar quadtree [24] in the Poincaré Disk Model. However, when splitting the quadrilaterals it shows that instead of splitting in the angular direction (Eq. 12) it produces better results when splitting in the radial direction instead, see fig 6. Thus the splitting of the cells occurs by

$$\text{mid}_r = \frac{\max_r + \min_r}{2}. \tag{13}$$

In the Euclidean space the midpoint of a cell is given by the arithmetic mean of the points contained within the cell. In hyperbolic space however such an arithmetic mean does not exist. As such the paper proposes to use the following closed form formula to approximate the midpoint with an error rate of 7% [24]

$$m(\{v_j\}) = \sum_j \left(\frac{\gamma(v_j)}{\sum_\ell \gamma(v_\ell)}\right) v_j \tag{14}$$

where $\gamma(v_j) = 1/\sqrt{1 - \|v_j\|^2}$ and v_j are the coordinates of the point in the Klein model. This approximation is used because it allows us to compute the mid-points as a rolling average, meaning the runtime of inserting a new point and updating all midpoint information remains $\mathcal{O}(\log n)$. [19]
in Eq. 8 and Eq. 9 it uses the Poincaré Disk model's distance formula and its gradient for the calculations 2. It also makes use of the projection step from Eq. 10. It aims to speed up the computation of the second sum of Eq. 9 following the same concept as the euclidean case, that is, it traverses the tree for every query point y_i checking at every node whether

$$\frac{r_{cell}}{d^{\mathcal{H}}(\mathbf{y}_i, \mathbf{y}_{cell})} < \theta \tag{15}$$

holds. This is the hyperbolic version of 11. If this holds, just like in the Euclidean case, we stop the traversal of the subtrees and replace the summands by

$$-N_{cell}\left(q^{\mathcal{H}}_{i,cell}\right)^2 Z^{\mathcal{H}} \frac{\delta d^{\mathcal{H}}(\mathbf{y}_i, \mathbf{y}_{cell})}{\delta \mathbf{y}_i}, \tag{16}$$

where N_{cell} is the number of points contained in the cell, y_{cell} is the midpoint of the cell by 14 and

$$q^{\mathcal{H}}_{i,cell} Z^{\mathcal{H}} = \left(1 + d^{\mathcal{H}}(\mathbf{y}_i, \mathbf{y_{cell}})^2\right)^{-1}.$$

When performing the gradient descent-step it is also necessary to account for the curvature of geodesics in the Poincaré model. For this we use the exponential map of the Poincaré Disk model [19]. It is shown within the paper that the asymptotic runtime of this acceleration is $\mathcal{O}(n^\alpha$ where $\alpha = \frac{\log(t_{i+1} - \log(t_i)}{\log(n_{i+1} - log(n_i)}$, $(n_i, n_{i+1}$ a pair of input sizes and corresponding average iteration times $(t_i, t_{i+1}$.
We build our implementation for a polar quadtree in the Klein Disk model upon this data structure. Although we do not expect a better asymptotic runtime, we do hope to find speed ups in two areas of computation. Firstly, when calculating the midpoint of a cell, the points in the polar quadtree need to be converted to points in the Klein Disk model and then the result of the Eq. 14 needs to be converted back to the Poincaré Disk model. By adopting the implementation to the embed directly in the Klein Disk model we hope to find a speed up by making these conversions unnecessary.
Secondly, we hope to find a speed up due to the lack of a need for an exponential map in the Klein Disk model after each gradient descent-step. This is because, as mentioned earlier, geodesics in the Klein Disk model are also straight lines in the Euclidean sense.

3 Related work

T-SNE is the main focus of our research [23]. T-SNE is a dimensionality reduction algorithm, these algorithms are classified by whether they obtain their embedding linearly or non-linearly and whether they aim to preserve local neighbourhoods or global distances. T-SNE falls under the non-linear locally preserving methods. For our motivation to choose t-SNE as an embedding algorithm we refer you to a recent survey on this topic that finds that t-SNE "perform[s] the best in cluster identification and membership identification." [26]. In

[5] it is explained in detail how one can interpret the 2 parts of the sum from 7 as positive and repulsive forces working upon a point in the embedding. Finding embeddings into the Klein Disk model in scientific literature is quite difficult, as most embeddings embed into the Poincaré Disk model as this is generally more suited due to the conformality of the Poincaré model. One successful embedding into the Klein model is [1]. However in this paper it is also noted that "the Klein model which is less well-behaved (i.e., more scaled and anisotropic) than the Poincare model". As such although we do expect a speed up by not having to do conversions for the midpoint calculations and no exponential map, we expect worse results when compared to the implementation for the Poincaré Disk.

4 methodology

In this section we will discuss and prove our method for adapting the Barnes-Hut acceleration for t-SNE in the Klein Disk model. We will do so by discussing how we perform gradient descent in the Klein Disk model and how we have changed the Barnes-Hut approach for the Poincaré Disk model from [19] to instead work for the Klein Disk model.

4.1 Performing gradient descent in the Klein Disk model

To perform the gradient descent in the Klein Disk model we remain as close to possible as the original proposed method by [19]. To this end we will also need the following 2 partial derivatives of the Klein Disk model's distance equation, for the x direction this gives

$$\frac{\delta d^{\mathcal{H}}(\mathbf{p},\mathbf{q})}{\delta x_p} = -\frac{x_p^2\rho + x_p x_q^2\alpha - x_p\beta + x_q\alpha\beta}{\zeta}, \tag{17}$$

where $\rho = x_q - y_p x_q y_q$, $\alpha = y_p^2 - 1$ and $\beta = y_p y_q$, for the y direction we this gives

$$\frac{\delta d^{\mathcal{H}}(\mathbf{p},\mathbf{q})}{\delta y_p} = -\frac{y_p\gamma + \omega + y_p^2\mu}{\zeta}, \tag{18}$$

where

$$\gamma = (x_p^2(x_q^2 - y_q^2) - 2x_p x_q + y_q^2 + 1)$$

,

$$\omega = \left(x_p^2 - 1\right) y_q \left(x_p x_q - 1\right)$$

and

$$\mu = y_p^2\left(y_q - x_p x_q y_q\right)$$

In both equations

$$\zeta = \kappa^{3/2}\sqrt{\varphi}\sqrt{\frac{\eta^2}{\nu\tau}}\sqrt{-\psi^2 + \upsilon}$$

where
$\kappa = -x_p^2 - y_p^2 + 1$,
$\varphi = -x_q^2 - y_q^2 + 1$,
$\eta = (x_p x_q + y_p y_q - 1)$,
$\nu = x_p^2 + y_p^2 - 1$,
$\tau = x_q^2 + y_q^2 - 1$,
$\upsilon = (x_p - x_q)^2 + (y_p - y_q)^2$
and $\psi = x_p y_q - y_p x_q$.

These variations are needed as the derivative of the hyperbolic distance in Eq. 9. Due to the complexity of the calculation for these gradients and the limited time available for this research we have decided to obtain these gradients with the use of a derivative calculator and verifying the correctness of the steps instead of doing the derivation by hand, see section 6 for further explanation on how this was done.
The steps for the gradient descent remain the same as in section 2.3. For the hyperbolic distance in Eq. 8 we use the Klein Disk model's distance defined in Eq. 2.
Like mentioned earlier, we hope to find a speed up by not the lack to need to perform an exponential map step for the gradient descent in the Klein Disk model, as the geodesics of this model are not curved. We do still need to do the projection from Eq. 10 to make sure points remain within the bounds of the unit disk.

4.2 Accelerating t-SNE in the Klein Disk model

We aim to use the same polar quadtree data structure as is used for the Poincaré Disk model but alter it to fit to the Klein model instead. As the Klein Disk model has the same domain as the Poincaré Disk model the changes required to this data structure lie mostly in the way it deals with distances. One of the required changes is how we decide whether to summarize by Eq. 15 we now use the hyperbolic distance for the Klein Disk model, given in Eq. 2. We also have to alter the way we compute the longest diagonal of a cell by using the distance for the Klein Disk model here as well.
Furthermore, we can skip the step in the midpoint calculation where the implementation from [19] converts the points from the Poincaré Disk model to the Klein Disk model for the computation and then converts the result back to the Poincaré Disk model. As our polar quadtree is already in the Klein model no conversion between these models is necessary for this step. Here we also expect to see a speed up when comparing the two implementations.
The acceleration works the same as in the Poincaré Disk model, that is we aim to summarize groups of points sufficiently far away from a query point y_i. The formula of the summarized points remains as in Eq. 16 using Eq. 17 and Eq. 18 for the gradient of the hyperbolic distance. The rest of the acceleration remains analogous to the one for the Poincaré Disk model. As such the application to other hyperbolic t-SNE schemes proven in [19] remains true.

5 Experimental Setup and Results

In this section we will discuss what experiments we have conducted, how they were setup and what the results of these experiments were. The experiments conducted include analysis of the performance, in terms of both run time and quality of results, of the embedding with a focus on comparing it to the results of the same dataset embeddings but with the implementation of the Barnes-Hut scheme for the Poincaré disk. Furthermore we will look into the effects of θ on both the quality of results and the runtime of the algorithm. We will also compare our acceleration to the exact version for hyperbolic t-SNE in the Klein disk model.

5.1 Setup of the experiments

We set up our experiments according to mostly the same setup proposed in [19] as much of the same principles apply to the Klein Disk model the same as they do for the Poincaré Disk model when performing gradient descent. As such we also start by doing PCA to speed up computations by reducing the dimensionality of the data to 50 dimensions. Then we follow the same strategies employed by regular t-SNE by performing an early exeggeration step, where the positive forces p_{ij} are multiplied by a factor to emplify them [22]. Like [19] we decided to use an exegeration factor of 12. Then we perform up to 750 iterations with regular p_{ij}, the same as regular t-SNE [22]. For the learning rate we deviate from the regular case of t-SNE, as we embed into hyperbolic distance where distances between points close to the edge of the disk are rather small, we adjust our learning rate as such. That is why for the learning, like in [19], we take

$$\mu = \frac{n}{12 \cdot 1000}. \tag{19}$$

This is smaller then the recommended learning rate of $\mu = n/12$ [4] for Euclidean t-SNE. We use this smaller learning rate because with a higher learning rate the points tend to end up close to the boundary of the Klein Disk. For the last parameters for the gradient descent optimization we turn to [19] and use the same parameters. That is, we use momentum and gains for the gradient descent as in the Euclidean setting [22], allowing us to use a smaller learning rate that grows with momentum and gains. We use the recommended parameters for the Euclidean case, which means we take a momentum of 0.5 for the early exaggeration and a momentum of 0.8 for the non-exaggerated gradient descent. For the perplexity we stay within the recommended range [4] and take a value of 30. All accelerations are run with $\theta = 0.5$ unless specified otherwise. We run all the experiments on equal machinery on the CPU. For the analysis of our results we turn to the same precision/recall graphs used in [19] [13]. For this, like in previous work, we take a maximum neighborhood size $k_{max} = 30$. We compute the number of true positives for each of the values of $k \in \{1, ...k_{max}\}$ as $TP_k - N_{k_{max}}(X) \cap N_k(Y)$, this means that it is considered a true positive if a point is both in the higher and embedded low dimensional neighborhood, given the respective distances. We then obtain the precision by $PR_k = |TP_k|/k$, the recall is obtained by $RC_k = |TP_k|/k_{max}$. As such an ideal situation would be for the precision to always be 1 and the recall to grow as k/k_{max}, data sets however do not necessarily show this property, nor does t-SNE find such a solution [19]. As such the goal of our method is not to have perfect precision recall, but rather show that our acceleration does not significantly decrease the quality of results when compared to an exact solution.

For the experiments we mainly used 3 datasets, the Planaria [16] and C.Elegans datasets are experimentally obtained gene expression atlases containing flatworm cell data. We also use the MNIST dataset for visualization purposes. We chose these datasets due to their size and use in previous work [19], [7]. To see visual representations of the embedding of the Planaria and MNIST datasets see fig. 13 and fig. 12

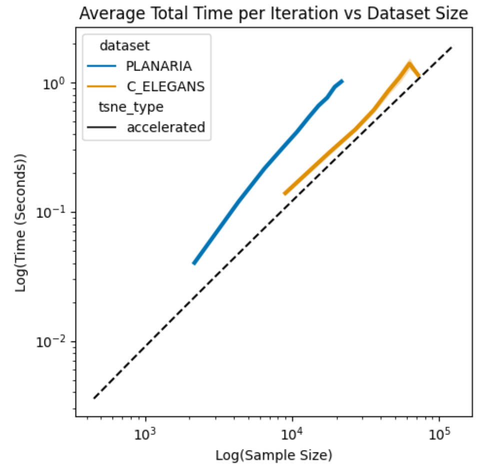

Figure 7: Graph showing the average total time per iteration compared to the size of the dataset sample.

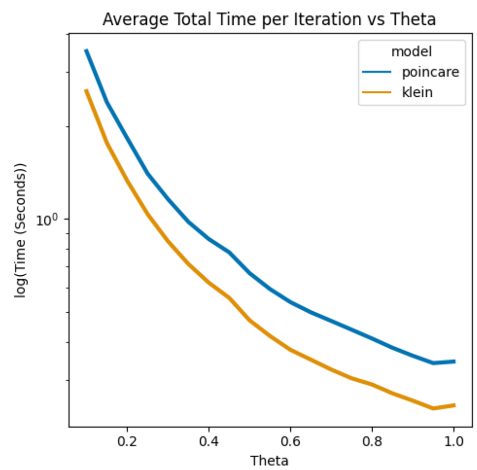

Figure 8: Graph showing the average time per iteration per value of theta for the Planaria dataset. Note that the Klein implementation outperforms the Poincaré implementation for all values of theta.

5.2 Asymptotic runtime

The argumentation for the time gained by acceleration is analogue to the one given in [19]. This is because we use the same concepts, only swapping the hyperbolic distance functions, which do not lead to a higher run-time complexity. As such we can follow the same argumentation for the asymptotic runtime. Thus we obtain an asymptotic runtime as mentioned in 2.6. By looking at the graph in 7 we can see that the runtime indeed does not show quadratic growth with regards to the sample size of the dataset, just like for the Poincaré implementation from [19].

5.3 Further time gain

Like expected we see a faster computation time for the Klein Disk model when compared to the Poincaré Disk. In fig. 8 it can be seen that for every value of θ we see a shorter total time

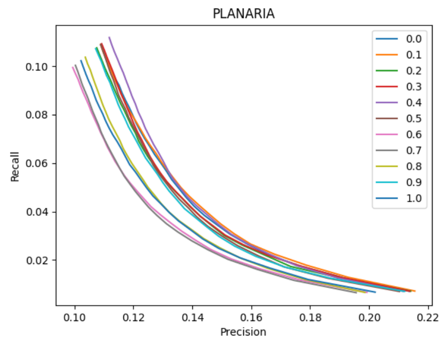

Figure 9: Graph showing the precession recal curves for values of θ between 0.0 and 1.0 for our implementation.

per iteration for the implementation for the Klein Disk model when compared to the one for the Poincaré Disk model. This speed up can be attributed to the two factors previously mentioned, namely the lack of a need to converse between models when calculating the midpoints and the lack of a need for an exponential map. Due to these speed ups our model outperforms the one for the Poincaré Disk model in terms of run time.

5.4 Effect of θ

To see the effect of θ on the embedding we have conducted an experiment on the Planaria dataset, running the embedding for all values of theta between 0.0 and 1.0 taking steps of 0.1. In fig. 9 we see the precision recall curves for different values of θ, note that $\theta = 0.0$ gives the exact version of hyperbolic t-SNE without any approximation. For all values of θ we observe the precision recall curves do not change significantly, especially for values of $\theta < 0.5$ we observe that there is almost no loss in quality of results, proving our acceleration does not significantly impact the quality of results from the embedding for the Planaria dataset. The run time however does significantly increase the lower the value of θ, which can be seen in graph fig. 8. This is analogue to the behaviour seen for the implementation for the Poincaré Disk model [19].

5.5 Impact of splitting criterion

Just like with the Poincaré Disk model implementation we explored the 2 options for the splitting criterion. As expected, the split by length performs better as is the case for the Poincaré implementation from [19], this can be seen for the PLANARIA dataset from the graph in fig. 10.

5.6 Quality of results

The quality of results of our implementation follow similar trends as the one from [19] as can be seen by looking at the graph in fig. 11 we can see that the quality of results does not significantly decrease when using our acceleration method, analogue to the behaviour seen in the Poincaré Disk model's implementation. Comparing the quality of results of our implementation the the one for the Poincaré Disk, we soon find that the quality is significantly worse. When looking at fig.

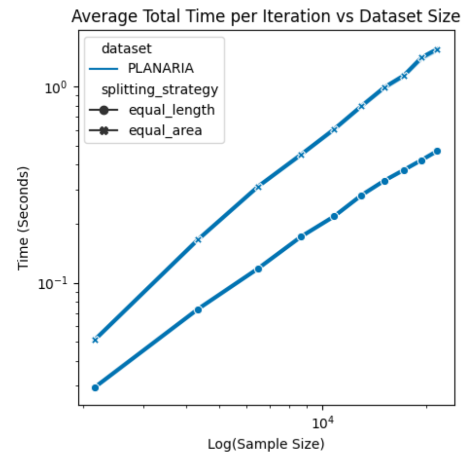

Figure 10: A comparison for the runtime when using different splitting strategies.

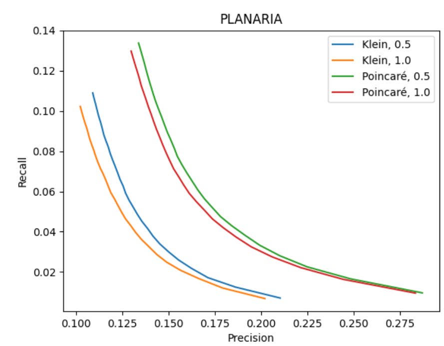

Figure 11: Graph comparing Klein and Poincaré precision recalls, for both exact and accelerated with $\theta = 0.5$ for the Planaria dataset. Note that the goal is for both to be as high as possible, therefore Poincaré outscores Klein on all values.

11 we can see that the Poincaré disk model outperforms our model.

6 Responsible Research

This section will discuss the responsible research conduction relating to our research. We will start wth a discussion on the datasets that were used. Secondly we will cover how we obtained our methodology and conducted our experiments. Lastly we will discuss some societal concerns with regards to our research.

6.1 Datasets used

For our research we decided to use the same datasets as the original paper proposing the acceleration method for hyperbolic t-SNE in the Poincaré Disk model [19]. We will briefly discuss each dataset and show how we ensured it was ethically appropriate.

Firstly, the MNIST dataset contains handwritten digits. As this data does not include personal or sensitive information,

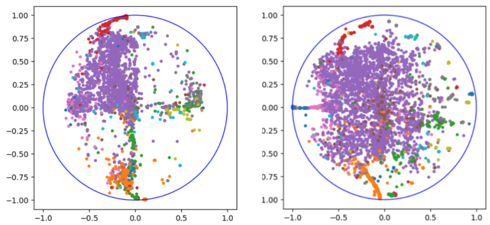

Figure 12: Embedding of the Planaria dataset, left Klein and right Poincaré

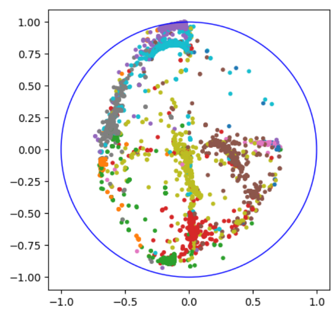

Figure 13: Klein model embedding of the MNIST dataset

we can safely use it without ethical concerns.
We also made use of datasets containing cell data, including the Planaria data set [16] and C.Elegans. All of these are originally obtained through gene expression atlases. Both the Planaria dataset and C.Elegans solely contain cell data on flatworms, as such they do not pose any ethical concerns related to personal or sensitive information.

6.2 Transparency and Reproducibility

Unfortunately due to a lack of time we were unable to conduct thorough experiments with a wide array of datasets, instead opting to mainly base our initial conclusion on thorough testing with the Planaria dataset, some testing with the C.Elegans data set and visualizations through the MNIST dataset. To draw full conclusions it would be necessary to run the experiments for other datasets as well to see whether the same behaviour is witnessed across different datasets, we can note however that we do expect this to be the case, as this is the case in [19] and our results show much similarity.
Our experiments were all run on equal machinery on the CPU. The code used to run the experiments can be found in the GitHub mentioned in the supplemental materials, including documentation on how one can setup an environment and run the experiments, or use our implementation for other purposes on their machinery.
For our methodology we needed a derivative of the distance function for the Klein Disk model. Due to the complexity of the calculations needed to compute this derivative, in combination with the short amount of time allocated for our research, we decided to use WolframAlpha (https://www.wolframalpha.com/) to compute the derivative and verify the correctness ourselves. We did this by giving the following input
"find the derivative ArcSinh[Sqrt[(a-c)^ 2+(b-d)^2-(a*d-c*b)^ 2]/Sqrt[1 - (a^ 2+b^ 2)]*Sqrt[1 - (c^ 2 +d^ 2)]]"
where $a = x_p$, $b = y_p$, $c = x_q$ and $y = y_q$ when looking at the regular formula, the formula can also be found in Eq. 2. This substitution was done for ease of input. We converted the result it gave us back to use our representation of the variables.

6.3 Societal impact

It should be noted that the results produced by t-SNE can vary in their quality from run to run. We observed during our research that sometimes the embeddings were worse. This poses a societal risk where an end user might put too much faith in the results obtained through our algorithm without verifying their correctness. To mitigate this the result should always be analyzed on the quality of it. It can also be recommended to run the embedding multiple times to find the optimal one.
Furthermore, t-SNE falls in line with other efficient data embedding and dimensionality reduction techniques and offers the same ethical concerns as these. As these techniques become more powerful and readily available they bring a range of societal ethical concerns that need to be addressed.
One concern raising from these techniques is their misuse in surveillance and privacy-intrusive applications. Usage of these techniques to analyze personal data at a granular level can potentially lead to privacy breaching and unauthorized profiling. It is of importance to implement clear guidelines and ethical norms to combat the misuse of these techniques and enforce that these techniques are only used in ways that respect both individual privacy and data protection laws.
Furthermore, the problem of data bias can also be of ethical concern in the use of the proposed technique. Data bias in the data fed into the hyperbolic t-SNE algorithm can get even more exaggerated in the data embedding. Biases like these can lead to discrimination or unfair treatment of certain groups in applications such as hiring, law enforcement or financial services. These biases are hard to detect and hard to deter, as such it is mostly up to the user of the techniques to ensure that the data they are putting into the algorithm is (mostly) free of (unfair) bias.
Lastly, the complexity of our data embedding technique can lead to distrust in the end-users of applications of the technique as it is difficult to understand and thus trust the results. To combat this mistrust it is important to keep the technique as transparent as possible and make sure there is clear and complete documentation so the results are verifiable and explainable to the end-users, leading to higher trust in the results obtained using this technique.

7 Conclusion and Future Work

7.1 Conclusion

Unfortunately due to a lack of time we were unable to conduct enough experiments to come to a full conclusion on our implementation of the acceleration data structure for the Klein Disk model and how exactly it compares to the Poincaré Disk model. Although we believe we can conclude that our acceleration method works for the Klein Disk model. From the experiment we conducted with the Planaria dataset we can see that our acceleration gives results similar in quality to an exact implementation for the Klein Disk model, see fig 9, while providing faster computation times, see fig. 8 further analysis is necessary to draw full conclusions on the embedding into the Klein model. For the parameters and datasets used we can conclude that our implementation performs worse in terms of quality of results, but better in terms of run time. As a preliminary conclusion we would not recommend using the Klein Disk model implementation over the one for the Poincaré Disk model, as the gain in runtime does not merit the loss in quality of results.

7.2 Future work

For further conclusions to be made about the t-SNE embedding into the Klein Disk model it is necessary to conduct more experiments using our implementation. Using different parameters and datasets. One thing that could be explored is with different learning rate, due to a lack of time we were unable to conduct further experimenting on this.
There are also some other avenues that can be explored regarding t-SNE embedding into the Klein Disk model. One such potential avenue is to replace the polar quadtree used in our acceleration with a modified regular quad tree. This quad tree would have to implement a way to handle cells containing areas that reach outside the unit disk or on the unit disk, as these points are not defined in the Klein Disk model. Similarly, exploring the use of grid squares instead of the polar quadtree to perform the acceleration would also be interesting.
For both of these methods it would be interesting to see whether the Klein Disk model or Poincaré Disk model offers better performance. As the Klein model has more similar geometric properties in regards to geodesics (straight lines) to Euclidean space when compared to the Poincaré Disk model, it would be interesting to see whether this property makes it more suited for an acceleration method that also makes use of straight lines to separate the cells.
Overall these remaining research directions in the field of hyperbolic t-SNE have the potential to further improve the efficiency and capabilities of hyperbolic t-SNE, which can benefit its broader use in scientific - and industrial domains.

Supplemental materials

The data sets used in our experiments are avialbe as follows: Planaria (https://shiny.mdc-berlin.de/psca/), MNIST (https: //yann.lecun.com/exdb/mnist/), C.Elegans (https: //github.com/Munfred/wormcells-data/releases). The implementation of the acceleration data structure and the code to run the experiments can be found at https://gitlab.ewi.tudelft.nl/cse3000/2023-2024-q4/Skrodzki_Eisemann/shared-approximating-nearest-neighbors-in-hyperbolic-spa

Figure credits

Figure 1 left side was taken from https://en.wikipedia.org/wiki/Beltrami%E2%80%93Klein_model, right side was taken from https://en.wikipedia.org/wiki/Poincar%C3%A9_disk_model both are taken from wikipedia and are thus part of the public domain. Figures 3,4,5 and 6 were taken from [19]

References

[1] Noam Aigerman and Yaron Lipman. Hyperbolic orbifold tutte embeddings. *ACM Trans. Graph.*, 35(6):217–1, 2016.

[2] El-Ad David Amir, Kara L Davis, Michelle D Tadmor, Erin F Simonds, Jacob H Levine, Sean C Bendall, Daniel K Shenfeld, Smita Krishnaswamy, Garry P Nolan, and Dana Pe'er. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature biotechnology*, 31(6):545–552, 5 2013. doi: 10.1038/nbt.2594. URL https://www.nature.com/articles/nbt.2594#citeas.

[3] Wenjia Ba, J. Michael Harrison, and Harikesh S. Nair. Advertising media and target audience optimization via high-dimensional bandits, 2022.

[4] Anna C. Belkina, Christopher O. Ciccolella, Rina Anno, Richard Halpert, Josef Spidlen, and Jennifer E. Snyder-Cappione. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature communications*, 10(1), 11 2019. doi: 10.1038/s41467-019-13055-y. URL https://www.nature.com/articles/s41467-019-13055-y#citeas.

[5] Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Attraction-repulsion spectrum in neighbor embeddings, 2022.

[6] Yunhui Guo, Haoran Guo, and Stella Yu. Co-sne: Dimensionality reduction and visualization for hyperbolic data, 2022.

[7] Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature communications*, 11(1), 6 2020. doi: 10.1038/s41467-020-16822-4. URL https://www.nature.com/articles/s41467-020-16822-4#citeas.

[8] Dmitry Kobak and George C. Linderman. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature biotechnology*, 39(2):156–157, 2 2021. doi: 10.1038/s41587-020-00809-z. URL https://www.nature.com/articles/s41587-020-00809-z#citeas.

[9] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3), September 2010. ISSN 1550-2376. doi: 10.1103/physreve.82.036106. URL http://dx.doi.org/10.1103/PhysRevE.82.036106.

[10] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 401–408, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0201847051. doi: 10.1145/223904.223956. URL https://doi.org/10.1145/223904.223956.

[11] George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nature methods*, 16(3):243–245, 2 2019. doi: 10.1038/s41592-018-0308-4. URL https://www.nature.com/articles/s41592-018-0308-4#citeas.

[12] Arpit Narechania, Alireza Karduni, Ryan Wesslen, and Emily Wall. Vitality: Promoting serendipitous discovery of academic literature with transformers visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):486–496, 2022. doi: 10.1109/TVCG.2021.3114820.

[13] N. Pezzotti, T. Höllt, B. Lelieveldt, E. Eisemann, and A. Vilanova. Hierarchical Stochastic Neighbor Embedding. *Computer graphics forum*, 35(3):21–30, 6 2016. doi: 10.1111/cgf.12878. URL https://onlinelibrary.wiley.com/doi/10.1111/cgf.12878.

[14] Nicola Pezzotti, Julian Thijssen, Alexander Mordvintsev, Thomas Höllt, Baldur Van Lew, Boudewijn P.F. Lelieveldt, Elmar Eisemann, and Anna Vilanova. Gpgpu linear complexity t-sne optimization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1172–1181, 2020. doi: 10.1109/TVCG.2019.2934307.

[15] Peter Pirolli, Stuart K. Card, and Mija M. Van Der Wege. Visual information foraging in a focus + context visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, page 506–513, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133278. doi: 10.1145/365024.365337. URL https://doi.org/10.1145/365024.365337.

[16] Mireya Plass, Jordi Solana, F. Alexander Wolf, Salah Ayoub, Aristotelis Misios, Petar Glažar, Benedikt Obermayer, Fabian J. Theis, Christine Kocks, and Nikolaus Rajewsky. Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics. *Science*, 360(6391):eaaq1723, 2018. doi: 10.1126/science.aaq1723. URL https://www.science.org/doi/abs/10.1126/science.aaq1723.

[17] Jürgen Richter-Gebert. *Hyperbolic Geometry*, pages 483–503. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-17286-1. doi: 10.1007/978-3-642-17286-1_25. URL https://doi.org/10.1007/978-3-642-17286-1_25.

[18] Kyuyong Shin, Young-Jin Park, Kyung-Min Kim, and Sunyoung Kwon. Multi-manifold learning for large-scale targeted advertising system, 2020.

[19] Martin Skrodzki, Hunter van Geffen, Nicolas F. Chaves de Plaza, Thomas Höllt, Elmar Eisemann, and Klaus Hildebrandt. Accelerating hyperbolic t-sne, 2024.

[20] Alexander Strehl and Joydeep Ghosh. Relationship-Based clustering and Visualization for High-Dimensional Data Mining. *INFORMS journal on computing*, 15(2):208–230, 5 2003. doi: 10.1287/ijoc.15.2.208.14448. URL https://doi.org/10.1287/ijoc.15.2.208.14448.

[21] Mark van de Ruit, Markus Billeter, and Elmar Eisemann. An efficient dual-hierarchy t-sne minimization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):614–622, 2022. doi: 10.1109/TVCG.2021.3114817.

[22] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.*, 15(1):3221–3245, jan 2014. ISSN 1532-4435.

[23] Laurens van der Maaten, TiCC, and Geoffrey Hinton. Visualizing Data using t-SNE. Technical report, 11 2008. URL https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

[24] Moritz Von Looz, Henning Meyerhenke, and Roman Prutkin. *Generating Random Hyperbolic Graphs in Subquadratic Time*. 1 2015. doi: 10.1007/978-3-662-48971-0\{_}40. URL https://link.springer.com/chapter/10.1007/978-3-662-48971-0_40#citeas.

[25] Jiang Wu, Dongyu Liu, Ziyang Guo, Qingyang Xu, and Yingcai Wu. Tacticflow: Visual analytics of ever-changing tactics in racket sports. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):835–845, 2022. doi: 10.1109/TVCG.2021.3114832.

[26] Jiazhi Xia, Yuchen Zhang, Jie Song, Yang Chen, Yunhai Wang, and Shixia Liu. Revisiting dimensionality reduction techniques for visual cluster analysis: An empirical study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):529–539, 2022. doi: 10.1109/TVCG.2021.3114694.

[27] Yuansheng Zhou and Tatyana O. Sharpee. Hyperbolic geometry of gene expression. *iScience*, 24(3):102225, 3 2021. doi: 10.1016/j.isci.2021.102225. URL https://www.cell.com/iscience/fulltext/S2589-0042(21)00193-0?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2589004221001930%3Fshowall%3Dtrue.