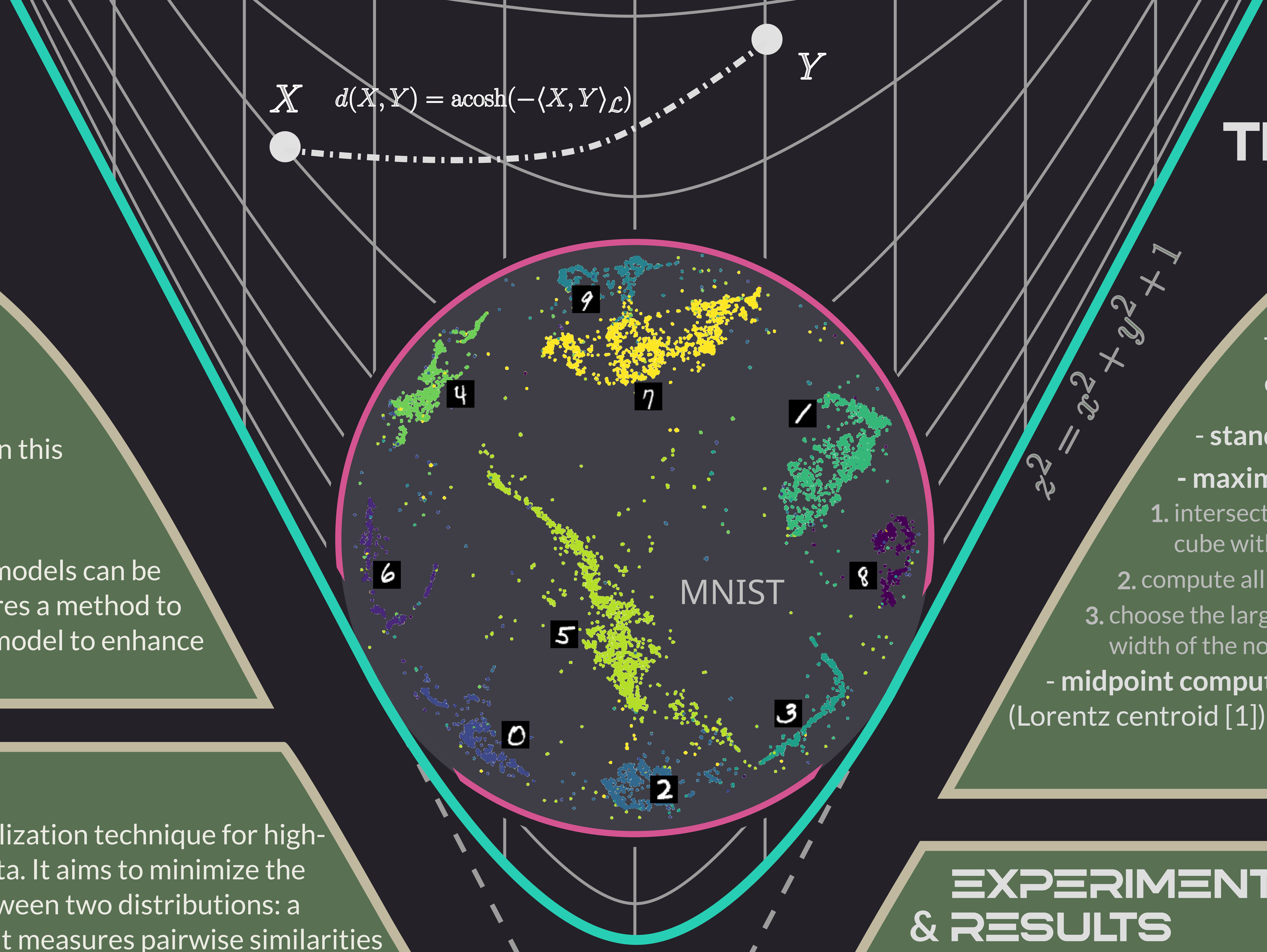
ACCELERATING T-SNE USING THE LORENTZ MODEL

AUTHOR: DANIEL PETER (d.peter@student.tudelft.nl)
SUPERVISORS: M. SKRODZKI
E. EISEMANN

TU Delft — Delft University of Technology



INTRODUCTION

Visualizing high-dimensional data is crucial for unlocking insights into complex phenomena. This would require embedding onto lower-dimensional planes to facilitate human interpretation and analysis.

Dimensionality reduction techniques play an essential role in this process, preserving important metrics to reveal underlying structures of the data.

In order to preserve data hierarchy information, hyperbolic models can be used to represent the embedding space. This research explores a method to accelerate hyperbolic t-SNE using the Lorentz Hyperboloid model to enhance efficiency and quality.

HYPERBOLOID

Hyperbolic space is the unique, complete, simply connected Riemannian manifold with constant negative sectional curvature. There exist multiple equivalent models for hyperbolic space, such as the Poincare ball and half-plane model, the Klein model and the Hyperboloid (or Lorentz) model.

The Lorentz model of 2-dimensional hyperbolic space is defined as the Riemannian manifold $\mathcal{L}^2 = (\mathcal{H}^2, g^{\mathcal{L}})$, where

$$\mathcal{H}^2 = \{x \in \mathbb{R}^3 : x_0^2 = x_1^2 + x_2^2 + 1, x_0 > 0\}$$

is the upper sheet of a two-sheeted 2-dimensional hyperboloid [4].

T-SNE

t-SNE is a visualization technique for high-dimensional data. It aims to minimize the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding space [3].

RESEARCH QUESTION

Can the Lorentz Hyperboloid model be used as the embedding space, in conjunction with an octree acceleration data-structure, to improve the **quality of the embeddings** of Ht-SNE, while maintaining a **similar run-time efficiency** with the accelerated version?.

METHODOLOGY

The data-structure used for acceleration is a modified version of an **octree** (see bottom of poster). This data-structure is applied on the Lorentz coordinates of each embedded data point. The octree has the following:

- **standard octree splitting criterion**: center of cube
- **maximum width computation**:
  1. intersect all the segments of the cube with the hyperboloid
  2. compute all pairwise distances
  3. choose the largest one as the maximum width of the node
- **midpoint computation** (Lorentz centroid [1]): $c(\mathrm{y}_j) = \dfrac{\frac{1}{N}\sum_j^N \mathrm{y}_j}{\left\|\frac{1}{N}\sum_j^N \mathrm{y}_j\right\|_{\mathcal{L}}}$

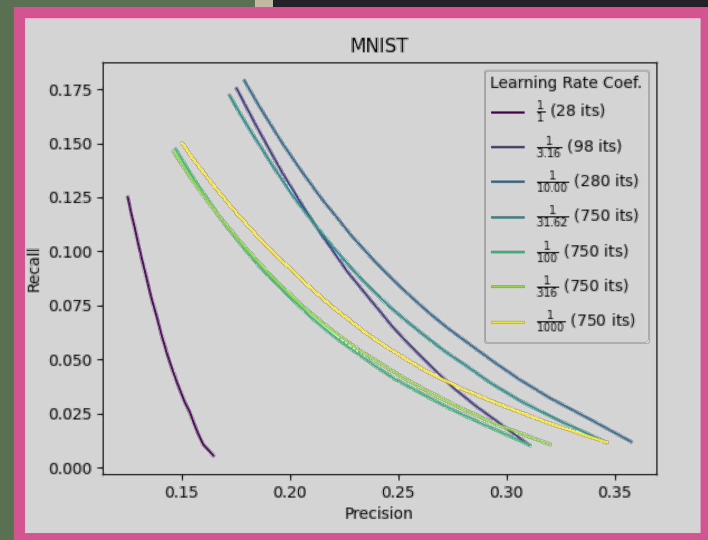
Algorithm 1 Summary of a Gradient Descent Step at y_i

Require: $Y \subseteq \mathcal{H}, \mathrm{y}_i \in Y$
- ▷ *Compute positive forces* ◁
- $f_{pos} \leftarrow \sum_{j\neq i} p_{ij}\, q^{\mathcal{L}}_{ij}\, Z^{\mathcal{L}}\, \nabla_{y_i} d^{\mathcal{L}}_{ij}$
- $f_{neg} \leftarrow 0$
- $octree \leftarrow$ build octree on Y
- **for all** $\mathrm{y}_j \in Y$ **do**
  - ▷ *Perform depth-first traversal of the octree* ◁
  - cell $\leftarrow$ root of $octree$ ▷ *Start with root cell*
  - **while** $r_{\text{cell}}/d^{\mathcal{L}}(\mathrm{y}_i, mid_{\text{cell}}) \geq \theta$ **do**
    - cell $\leftarrow$ child of $tree$ that contains y_j
  - ▷ *Compute negative force using the summarization* ◁
  - $f_{neg} \leftarrow f_{neg} - N_{\text{cell}}\,(q^{\mathcal{L}}_{ij})^2\, Z^{\mathcal{L}}\, \nabla_{\mathrm{y}_i} d^{\mathcal{L}}(\mathrm{y}_i, mid_{\text{cell}})$
- ▷ *Gradient in ambient space* ◁
- $\nabla^{\mathbb{R}^3}_{\mathrm{y}_i} C^{\mathcal{L}} \leftarrow f_{pos} + f_{neg}$
- ▷ *Gradient in tangent space* ◁
- $\nabla^{\mathcal{L}}_{\mathrm{y}_i} C^{\mathcal{L}} \leftarrow \nabla^{\mathbb{R}^3}_{\mathrm{y}_i} C^{\mathcal{L}} + \langle \mathrm{y}_i, \nabla^{\mathbb{R}^3}_{\mathrm{y}_i} C^{\mathcal{L}}\rangle_{\mathcal{L}} \cdot \mathrm{y}_i$
- ▷ *Project on the hyperboloid and update* ◁
- $\mathrm{y}_i \leftarrow \exp_{\mathrm{y}_i}(-\alpha \cdot \nabla^{\mathcal{L}}_{\mathrm{y}_i} C^{\mathcal{L}})$

EXPERIMENTS & RESULTS

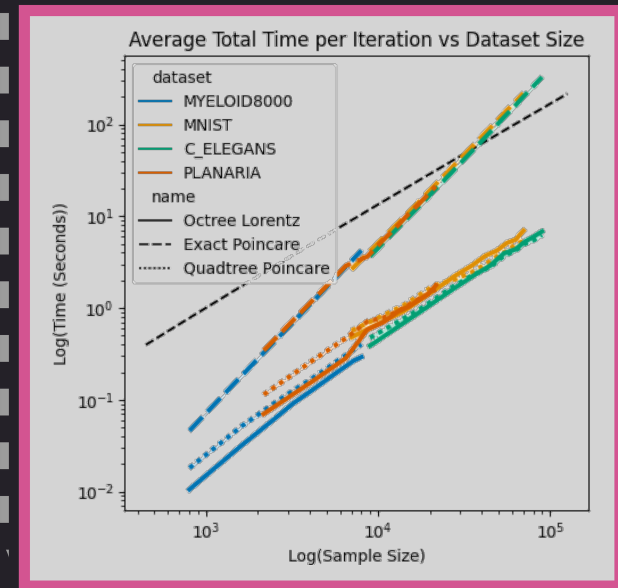
1. Embedding Quality vs Other Implementations:

Lorentz Ht-SNE performs better than both Quadtree Poincare [2] and exact versions (see prec.-vs-rec. graphs on bottom).



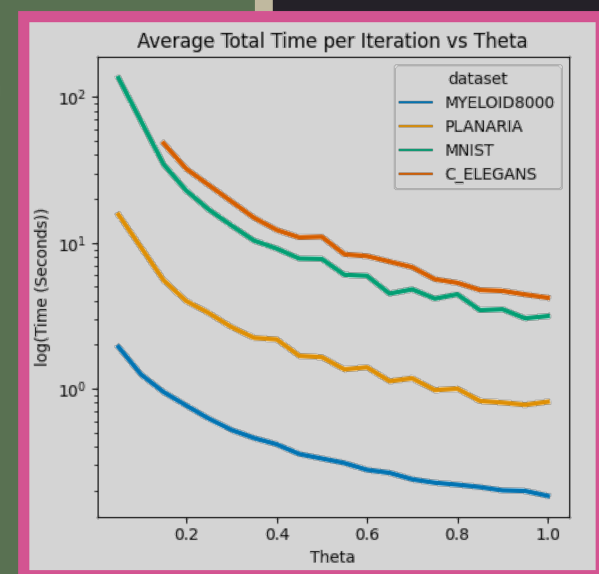
2. Embedding Quality per Learning Rate:

Tested an interval from $\frac{n}{12}$ to $\frac{n}{12000}$ logarithmically. Lorentz Ht-SNE performs best with values between $\frac{n}{120}$ and $\frac{n}{36}$. With greater values, the algorithm sends the points over the float max value.



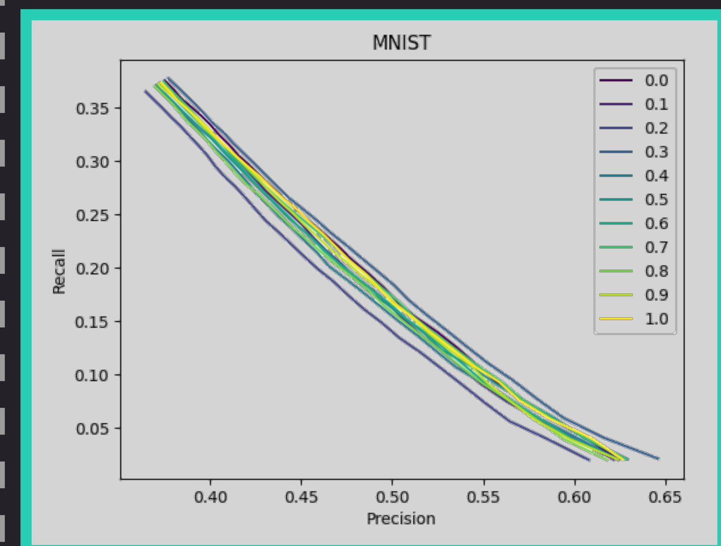
3. Absolute Run-time Comparison:

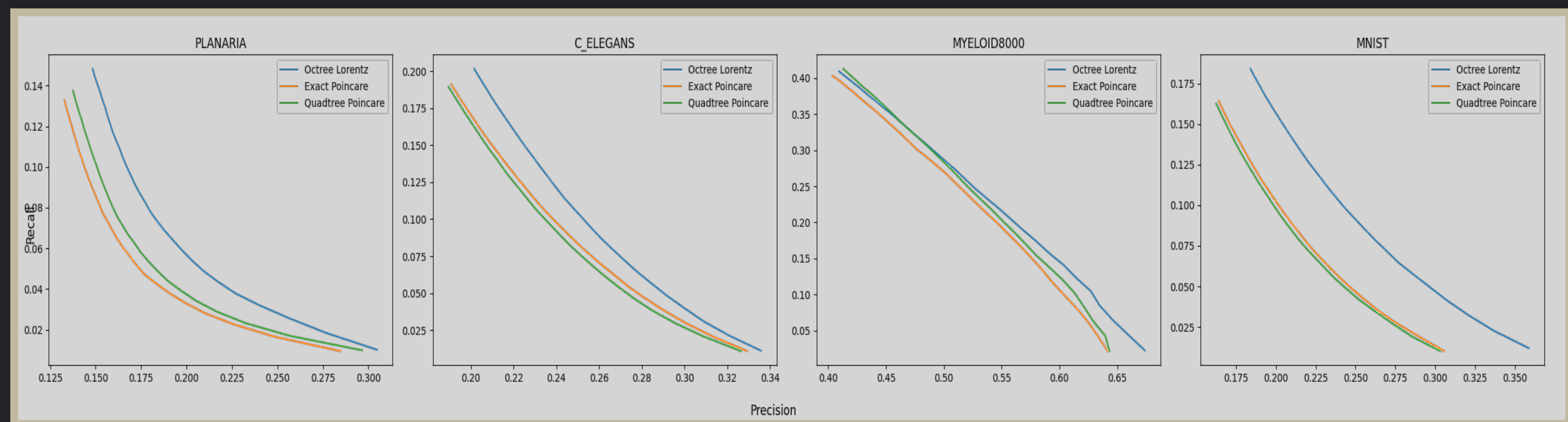
Our version maintains the speedup from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. Compared to the Poincare version, it has a similar performance, even surpassing it slightly.



4. Embedding Quality and Efficency per θ value:

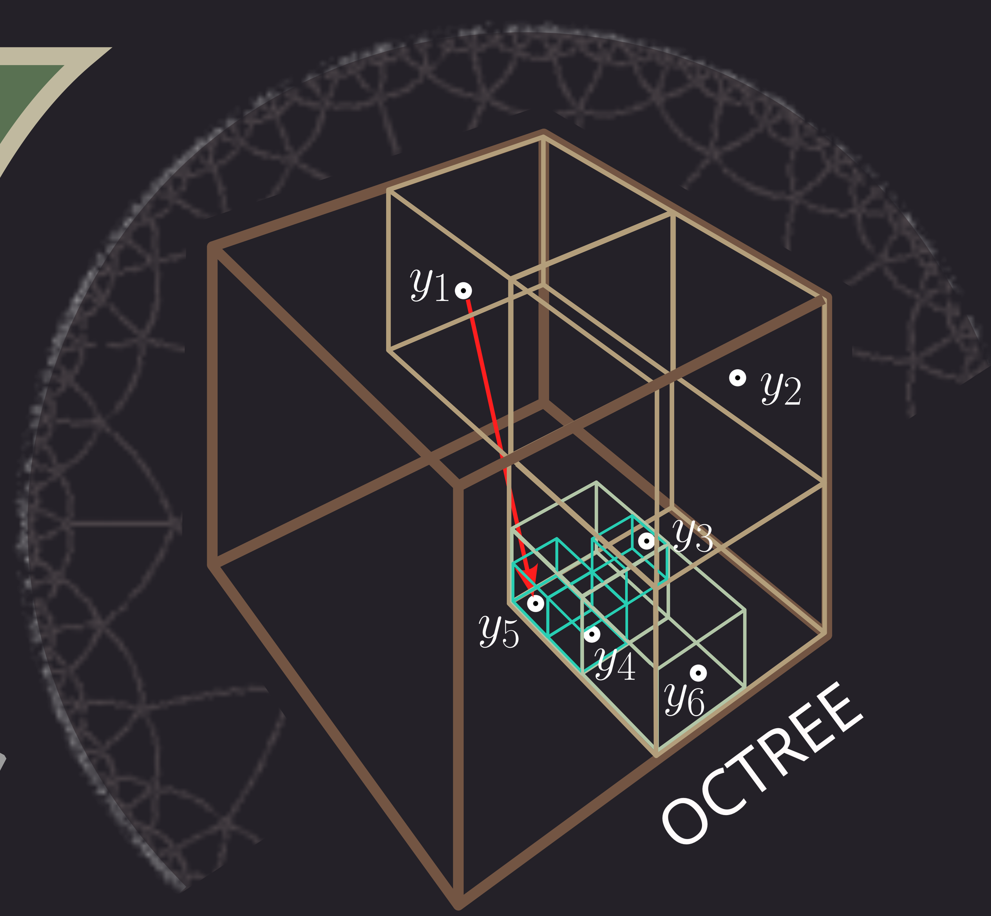
Increasing theta value reduces run-time significantly, while maintaining a similar quality.





CONCLUSION

The Lorentz Hyperboloid model improves the quality of hyperbolic t-SNE, making it a viable alternative for visualizing high-dimensional datasets.
Our method also maintains the acceleration factor from using an acceleration data-structure.



REFERENCES

[1] Marc Teva Law, Renjie Liao, Jake Snell, and Richard S. Zemel. Lorentzian distance learning for hyperbolic representations. In International Conference on Machine Learning, 2019.
[2] Martin Skrodzki, Hunter van Geffen, Nicolas F. Chaves de Plaza, Thomas Hollt, Elmar Eisemann, and Klaus Hildebrandt. Accelerating hyperbolic t-sne. IEEE Transactions on Visualization and Computer Graphics, pages 1–13, 202.4
[3] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(86):2579–2605, 2008.
[4] James W. Anderson. Hyperbolic geometry. 1999.