Hyperbolic Encoding in Neural Circuits

Sheng Yang[*]
John A. Paulson School of Engineering and Applied Science
Harvard University
Cambridge, MA 02138
shengyang@g.harvard.edu

Abstract

Hyperbolic spaces offer low distortion embedding of natural hierarchical information. This paper provides a general framework of measuring hyperbolicity to validate the hyperbolic assumption of data distribution and empirically verify the better performance in bringing a lower distortion with comparable reconstruction of olfactory and image dataset in hyperbolic embedding tasks than its Euclidean counterpart in a biologically plausible context.

1 Introduction

The n-dimensional hyperbolic space $\mathbb{H}^n$ is the unique simply-connected Riemannian manifold with a constant sectional curvature -1. One can embed finite trees into $\mathbb{H}^n$ with arbitrarily small distortion [Sarkar, 2011]. This offers biologists and neuroscientists a new perpective in processing natural information such as genes, odor, and images.

It has been suggest that various biological information are tree-like: [Klimovskaia et al., 2020] demonstrates that cell types are inherently dendrogram-like and an embedding into a low-dimensional hyperbolic spaces beats conventional dimensional reduction techniques in preserving distortions; [Zhou et al., 2018, Zhou and Sharpee, 2021] employ statisical testing to show that odors and gene expressions are inherently hyperbolic; [Khrulkov et al., 2020] argues that hyperbolic spaces offer better alternatives than Euclidean and spherical spaces to perform image-related downstream evluation such as classifications.

However, previous works use different evaluations and architectures to make the case for hyperbolic admissibility and bags the question of biological plausibility (i.e. whether such hyperbolic processing could be implemented through mammal brains). In our work, we would like to address these two issues by a unifying architecture and quantify such biological plausibility through a designed metric. We use the *relative hyperbolicity* δ_{rel} defined by Gromov's hyperbolicity to measure the similarity of a dataset to a tree and employ a modified *autoencoder* to obtain a faithful representation of the input dataset in the latent space as well as preserving the hyperbolic structure. These unified attempts largely align with previous claims that odor and image dataset are hyperbolic and that hyperbolic spaces are useful in preserving pairwise distances of hyperbolic input dataset.

This paper is organized in the following structure. Section 2 introduces necessary information of hyperbolic spaces. Section 3 covers the measurement of hyperbolicity abd Section 4 discusses the modified *autoencoder*. Section 5 provides empirical verifications through the lens of hyperoblicity and modified *autoencoder*. Finally, we discuss possible extensions in Section 6.

[*] Github: yangshengaa

Preprint. Under review.

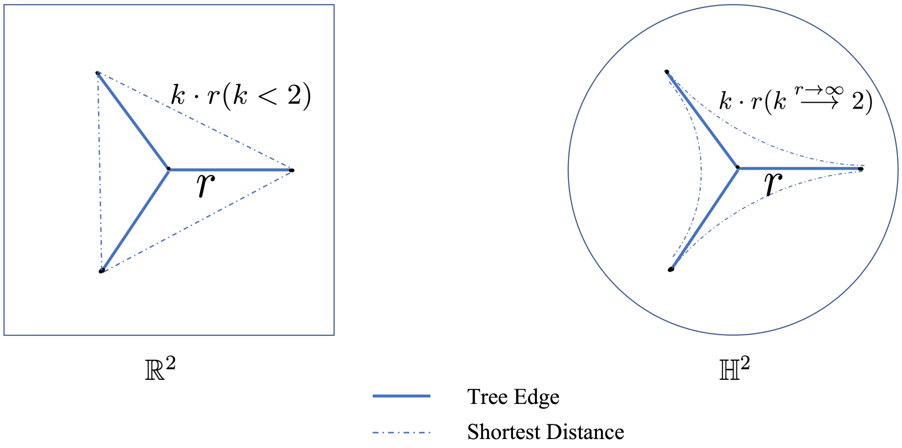
2 Preliminary: Hyperbolic Space

2.1 Hyperbolic Spaces Are Analogs of Trees

Hyperbolic spaces are the continuous analog of trees in that metrics in hyperbolic spaces are more-tree like than their Euclidean counterpart. The pairwise distances among three points that are not colinear in the Euclidean space, regardless of its dimension, is governed by the triangle inequality, meaning that the third length is always less than the sum of the other two; hyperbolic spaces alter this rule when two of the three points are sufficiently distanced: the third distance matches approximately as the addition of the other two. This behavior closely resembles a tree structure, where the distance between two point is the summation of the shortest path between them.

Such an analog is illustrated by Figure 1: thick lines demarcates the tree edges and dotted lines are shortest distances in respective spaces. When the leaf nodes are pushed towards the boundary, the ratio between the distance of the outer two points and the distance to center becomes exactly 2.

Figure 1: 2D illustration: Hyperbolic Spaces Are More Tree Like



2.2 Lorentz Model

Different hyperbolic spaces are available and are pairwise isometric [Peng et al., 2021]. It thus theoretically suffices to select and study one of them[2].

One popular choice is the Lorentz model $\mathbb{L}^n$, a submanifold in $\mathbb{R}^{n+1}$ defined by

$$\mathbb{L}^n := \{x \in \mathbb{R}^{n+1} | \; x_0 > 0, [x, x] = -1\} \tag{1}$$

where $[x, y] = -x_0 y_0 + \sum_{i=1}^{n} x_i y_i$ for all $x, y \in \mathbb{R}^{n+1}$ is known as the *minkowski product*. For any points $x, y \in \mathbb{L}^n$, the (geodesic) distance is given by

$$d_{\mathbb{L}^n}(x, y) = \text{arccosh}(-[x, y]) \tag{2}$$

Like other hyperbolic spaces, Lorentz model is locally Euclidean: the tangent space at x, denoted by $\mathcal{T}_x \mathbb{L}^n$, is a n dimensional Euclidean hyperplane. It is defined by

$$\mathcal{T}_x \mathbb{L}^n := \{v \in \mathbb{R}^{n+1} | \; [v, x] = 0\} \tag{3}$$

An important operation on the tangent space is the exponeital map that maps a vector from the tangent space back to the hyperbolic space with its length preserved. This is known in closed-form by

[2] Though theoretically isometric, different hyperbolic spaces are numerically disparate in representation and optimization due to rounding errors. In this paper we drop the discussion of numerical issues. See [Mishne et al., 2022] for more information

$$\exp_x(v) = \cosh(\|v\|_{\mathbb{L}^n})x + \sinh(\|v\|_{\mathbb{L}^n})\frac{v}{\|v\|_{\mathbb{L}^n}} \tag{4}$$

where $\|v\|_{\mathbb{L}^n} := \sqrt{[v, v]}$. These operations play a crucial role in the definition of a hyperbolic neural network defined later.

3 Hyperbolicity: How Tree-like a Dataset Is?

We introduce the *Gromov's δ-hyperbolicity* as the metric for a unified measurement of the similarity between a dataset and a tree. For any four points u, v, w, x in a metric space (X, d), define

$$\delta(u, v, w, x) = \frac{d(u, v) + d(w, x) - d(u, x) - d(w, v)}{2}, \ \delta(X, d) = \sup_{u,v,w,x \in X} \delta(u, v, w, x) \tag{5}$$

The metric space is thus called δ-hyperbolic with $\delta = \delta(X, d)$. For a complete justification of this derivation, see [Chen et al., 2013].

It can be shown that $\delta \geqslant 0$ for arbitrary metrix space (X, d). Furthermore, a tree has 0 hyperbolicity and a dataset with small hyperbolicity indicates a close resemblance to a tree.

Note that this metric is scale-variant. [Khrulkov et al., 2020] proposes a scale-invariant alternative termed the *relative hyperbolicity* $\delta_{\text{rel}} := \frac{2\delta(X)}{\text{diam}(X)}$, where $\text{diam}(X)$ is the diameter of the space or input features. This metric is used to report the hyperbolic admissibility of a dataset, detailed in Section 5.1.

4 Biologically Plausible Hyperbolic Neural Network

We use a modified *autoencoder* to obtain a lower-dimensional embedding as part of the learning process. *autoencoder* (AE) aims to train a latent representation while maintaining the ability to reconstruct the original input. This is done through a two-part process: *encoder* sends input to a low-dimensional latent space and *decoder* maps latent representations back to input dimension space to align with the input features. Training is thus done by minimizing the $l2$ norm of the difference between the input and reconstruction.

This crude form of *autoencoder* has two problems: latent space is set to Euclidean which may not well represent the input geometry, and the relations (distances) among samples are not defined since it may place samples arbitrarily in the latent space as long as it can be reconstructed. In this sense, *autoencoder* does not sufficiently describe the leanring mechinism.

To address these two issues, we replace the latent space by a hyperbolic one to provide a better fit for hierarchical inputs, and modify the loss term to equip a distance-preserving property of the latent space with respect to the input.

4.1 Loss

The modified loss is as follows: suppose $\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i \in \mathbb{R}^p, \forall i \in \{1, ..., n\}$ are the input dataset and reconstruction. Further assume that $\boldsymbol{z}_i \in \mathbb{R}^d$ are the latent representations in d dimension, we have

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|_2 + \gamma D(\{\boldsymbol{x}_i\}_{i=1}^n, \{\boldsymbol{z}_i\}_{i=1}^n) \tag{6}$$

where $\boldsymbol{\theta}$ are the parameters in the network and $D(., .)$ is a distortion measure between the input space and the latent space. The hyperparameter γ balances the reconstruction and distortion.

The concept of distortion borrows from manifold learning. The most general definition of distortion is as follows: suppose $f : (X, d_X) \mapsto (Z, d_Z)$, where X is the input space and Z the embedding space, the distrotion of the mapping is

$$
\begin{aligned}
D_{\text{contraction}} &= \underset{a \neq b \in X}{\text{mean}} \frac{d_X(a,b)}{d_Z(f(a), f(b))} \\
D_{\text{expansion}} &= \underset{a \neq b \in X}{\text{mean}} \frac{d_Z(f(a), f(b))}{d_X(a,b)} \\
D(X,Z) &= D_{\text{contraction}} \cdot D_{\text{expansion}}
\end{aligned}
\tag{7}
$$

Note that this formulation does not work well directly as the loss empirically. See Appendix A for more detailed discussion in the choice of distortion losses. We thus uses a different distortion proxy in training and we report the distortion according to this definition Equation (7).

4.2 Architecture

We introduce the architecture of Hyperbolic Autoencoder (HAE) here. The replacement of Euclidean space by a Hyperbolic space for the latent representation is done by two essential components: an exponential map, introduced in Equation (4), sending Euclidean features to Hyperbolic, and a Geodesic layer mapping hyperbolic features back to Euclidean as the input for subsequent decoder.

For exponential map, one popular choice is the exponential map at the original of the Lorentz model (i.e. $[1, 0, ..., 0] \in \mathbb{R}^{p+1}$). This treats the original latent space as the tangent space of Lorentz model at the origin.

Geodesic layer is the hyperbolic version of linear layer by measuring the normed distance between latent samples to different hyperbolic hyperplanes. If the hidden layer is m, then the geodesic layer learns m Lorentz hyperplanes in $\mathbb{L}^d$, the d-dimensional latent space. Geodesic layer is first introduced in the context of hyperbolic multilogistic regression by [Ganea et al., 2018] and later simplified by [Peng et al., 2021, Mishne et al., 2022]. The upshot is that though hyperbolic, reparametrization can be done such that we can use Euclidean features to denote the Lorentz hyperplane as to obviate riemannian optimization [Kochurov et al., 2020]. See Appendix B for more detail in the layer formulation.

The full model is pictorially described by Figure 2. Note that the exponential map is deterministic and geodesic layer is learned. HAE has more parameters to train than AE due to the GeodesicLayer.

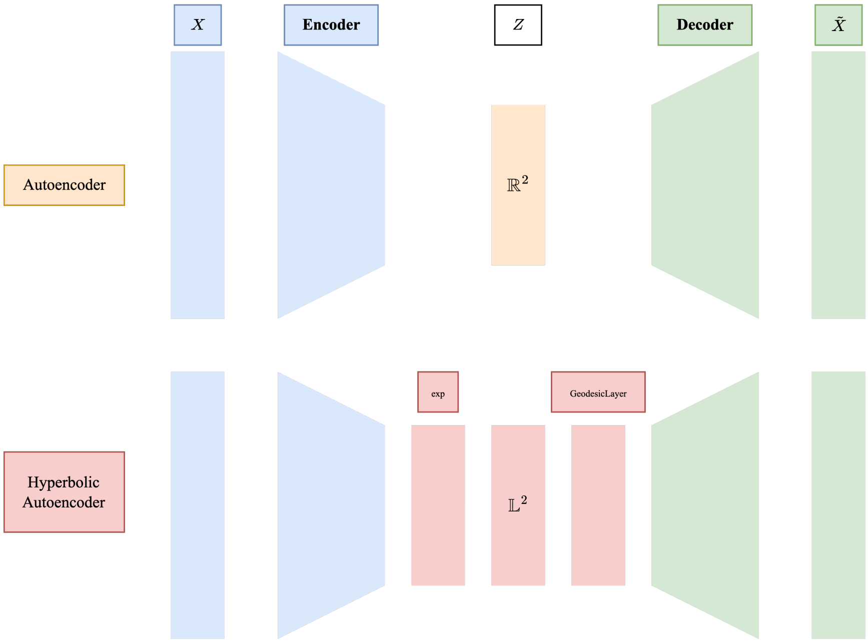


Figure 2: Hyperbolic Autoencoder: different from Autoencoder by its latent space representations

5 Empirical Study

This section covers empirical validations for the previous claims that olfactory data are hyperbolic [Zhou et al., 2018] and image dataset likewise [Khrulkov et al., 2020] through the lens of hyperbolicity and performance in the modified autoencoder (HAE).

Dataset The odor dataset [Zhou et al., 2018] provides chemical concentration measurement of tomato, blueberry, and strawberry picked from different locations across the globe, and the image dataset in this report uses the MNIST hand-written digits [LeCun and Cortes, 2010] and CIFAR10 [Krizhevsky et al., 2009], both of which come with 10 different labels for classifications. A brief summary of the number of observations n and number of features p is included in Table 1. Note that features with small variations are trimed before data processing.

Table 1: Data Summary

dataset	tomato	blueberry	strawberry	MNIST	CIFAR10
n	278	54	164	70000	50000
p	25	98	50	784	1024

5.1 Hyperbolic Admissibility

We use *relative hyperbolicity* δ_{rel} introduced in Section 3 to measure if a dataset is hyperbolic. Moreover, we compare the hyperbolicity with counterpart for the standard gaussian dataset with corresponding n and p for each dataset to see if the dataset is closer to a tree than its randomized version.

In Table 2 we present 10 measurement of the each dataset with 1000 samples drawn per measurement. Note that the closer to 0 the *relative hyperbolicity* is the more tree-like a dataset is. The odor dataset indeed demosntrate more tree-likeness than its random counterpart; the image dataset is rather divided. Although MNIST appears much more hierarchical than its gaussian partner, CIFAR10 behaves otherwise. Note that the metric reported here aligns with that in [Khrulkov et al., 2020] but challenges the claim that CIFAR10 is hyperbolic. Though the true *relative hyperbolicity* is significantly closer to 0 than 1, the comparison with gaussian indicates that the dataset purposefully deviates from being tree-like.

In general, the low *relative hyperbolicity* indicates that dataset considered above admits hyperbolic representations. This motivates obtaining a faithful distance-preserving hyperbolic representation using the modified *autoencoder*.

Table 2: hyperbolicity by dataset. Gaussian: δ_{rel} from standard gaussian from the same shape; True: the δ_{rel} from the original dataset.

dataset	tomato	blueberry	strawberry	MNIST	CIFAR10
Gaussian	0.2820 ± 0.02	0.2257 ± 0.02	0.2445 ± 0.02	0.0947 ± 0.01	**0.0769 ± 0.01**
True	**0.2209 ± 0.02**	**0.2004 ± 0.02**	**0.2301 ± 0.03**	**0.0440 ± 0.01**	0.2313 ± 0.03

5.2 Hyperbolic Embedding Performance

We use the architecture outlined in Section 4 to train and obtain a hyperbolic representation. We benchmark the performance by feeding into the normal autoencoder as well.

Odor dataset are of a smaller scale. We use 1 hidden layer for both encoder and decoder with a hidden dimenson of 20. Image dataset are fed into network with 2 hidden layers with 20 dimensions each. All data are embedded into 2D. We use Adam with a learning rate of 0.001, 1000 epochs, maximum batchsize of 500 to train all models.

We perform a 60/15/25 split into training, validation, and testing set. We test $\gamma \in \{0, 0.001, 0.01, 0.1, 1\}$ and select the one with the lowest validation reconstruction loss. The test reconstruction and distortion loss are reported in Table 3

Table 3: Performance by Model and Dataset

model	AE		HAE	
metric	reconstruction	distortion	reconstruction	distortion
tomato	19.8030	2.1563	**17.3361**	**1.0452**
blueberry	51.0061	1.6640	**41.1122**	**1.3420**
strawberry	**247.2654**	2.3036	282.2981	**1.0581**
MNIST	**33.6140**	1.7376	52.4670	**1.6122**
CIFAR10	**2044469.4045**	2.1770	3027167.7543	**1.2244**

Note that in all models we have that HAE achieves a lower distortion, suggesting that Hyperbolic spaces indeed are better to preserve hyperbolic structure of the inputs. The reconstruction part is rather ambivalent. This is possibily due to insufficient training of HAE for larger dataset. Note that HAE contains more parameters than AE in the decoder portion. HAE may be better off if longer training epochs are given.

In addition, we would like to comment that HAE performs well with shallow networks comparable to deep AE. In the odor dataset, we need 2 to 3 hidden layers for AE to achieve comparable reconstruction and distortion with 1-hidden-layer HAE. This implies that HAE has an additional advantage of saving the number of parameters to train and biologically more economic to obtain useful memories.

Problems: Spurious States of the Current Architecture A visualization of some subsets of samples suggest that the modified network converges to a spurious state where samples are all clustered together barring a useful generalization. In Figure 3, we have visualized subset of samples (0 vs 1, 4 vs 9) by their latent space representations in AE and HAE. We use a stereographic projection for $\mathbb{L}^2$ features to be visualizable in $\mathbb{B}^2$, an open unit ball in 2D. Notice that HAE provides denser representations and is not able to distinguish between different digits well.

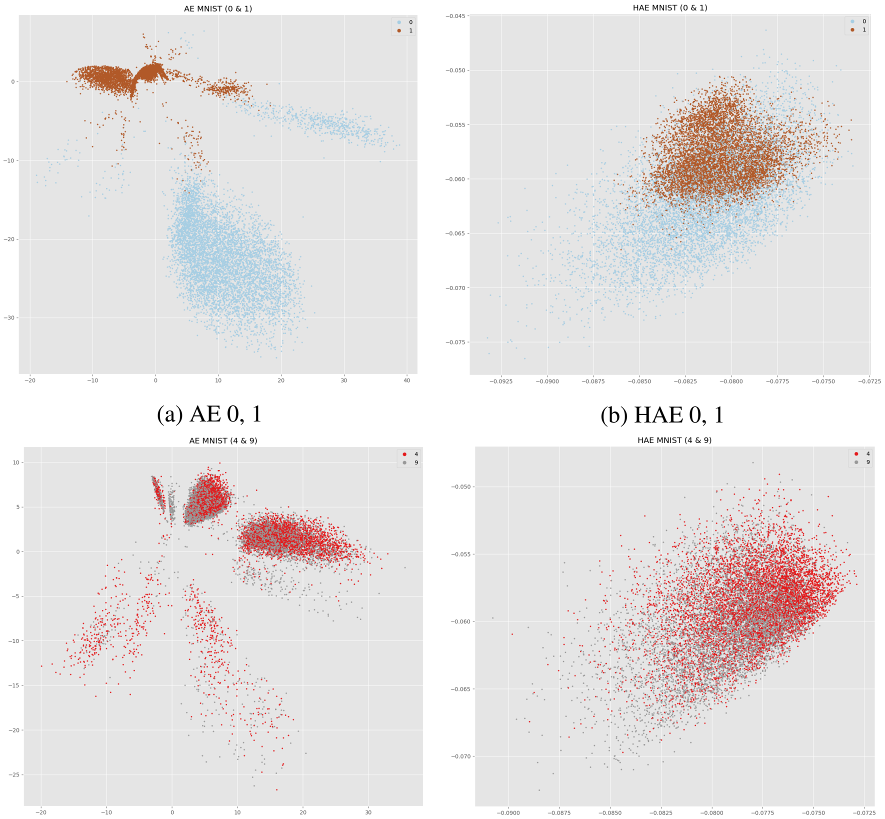
The reason is as follows: the definition of distortion Equation (7) (and the robust version defined in Equation (8)) has a tendancy to bring points closer in the embedding space for a lower distortion. Distortion is lower-bounded by 1, and if every embedding distance is a constant multiple of the real distance, then distortion achieves 1. This constant can be arbitrarily small, meaning that it tends to contract. Similar phenomenon has been observed in [Gala et al., 2019] but in a coupled autoencoder case. This suggests adding some penalization of the diameter of the embedding would be helpful in obtaining a more useful latent space memeorization.

However, there is another implication that HAE demonstrates strong potentials: [Sarkar, 2011] theoretically justifices that to obtain a low-distortion embedding of a finite tree in the hyperbolic space, the diameter of the embedding needs to be large. Currently the network is able to achieve lower distortion with a small scale; if penalization of the diameter is added, the hope is that points can be further pushed to the boundary and even lower distortion can be achieved and better hierarchical structure can be observed.

6 Future Work

The immediate extension is to incorporate a penalization of small diameter in the network training and identify theories supporting the biological plausibility of this term to prevent trivial memorization and provide some separability of instances from different classes. Also, it would be interesting to see how the performance change by embedding into other dimensions.

In addition, preliminary works have been done on hyperbolic multilogistic regression but are not able to show that the hyperbolic version performs better than the Euclidean one. It would be interesting to identify the right modification to better leverage hyperbolic spaces in classification tasks. Code for MLR is also included in Section 7.



(a) AE 0, 1

(b) HAE 0, 1

(c) AE 4, 9

(d) HAE 4, 9

Figure 3: Unhelpful Memorization by HAE: clustered in a small region. left panel: AE; right panel: HAE

7 Code Availability

All codes for measuring hyperbolicity and Hyperbolic Autoencoder are available at https://github.com/yangshengaa/hyperbolic-neural-circuits.

8 Conclusion

In this work we have provided a unified metric to measure the similarity of a dataset to a tree using the *relative hyperbolicity* and demonstrated a neural network model leveraging the hyperbolic structure by incorporating a distortion term in the loss in a modified *autoencoder* framework. The HAE shows potential in retaining low-distortion latent representations but requires further research into expanding the scale of the embeddings for lower distortion as well as results with better generalizations.

Broader Impact

In this paper, we propose a modified architecture of *autoencoder* to generate trustworthy latent space representation of hierarchical natural information such as odor and images. The applications also extend to other fields with hierarchical structures such as gene expressions, social networks, and supply chain. Each of these applications may provide substantial social impact in providing insights of the original dataset by analyzing its simpler but faithful representation in the latent space.

However, with careless model tunning, historical bias may persist in the latent representation. A good reflection of reality does not suggest an impartial or beneficial generalization. Before making predictions using the "faithful" latent space, one should identify if the trained represenetation inherit such social bias. If so, the elemination or rebalancing of these issues are non-trivial tasks to go through. We encourage all researchers to recognize the risk and develop necessary tools to help with the elemination.

References

Wei Chen, Wenjie Fang, Guangda Hu, and Michael W Mahoney. On the hyperbolicity of small-world and treelike random graphs. *Internet Mathematics*, 9(4):434–491, 2013.

Rohan Gala, Nathan Gouwens, Zizhen Yao, Agata Budzillo, Osnat Penn, Bosiljka Tasic, Gabe Murphy, Hongkui Zeng, and Uygar Sümbül. A coupled autoencoder approach for multi-modal analysis of cell types. *Advances in Neural Information Processing Systems*, 32, 2019.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

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

Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature communications*, 11(1):1–9, 2020.

Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Gal Mishne, Zhengchao Wan, Yusu Wang, and Sheng Yang. The numerical stability of hyperbolic representation learning. *arXiv preprint arXiv:2211.00181*, 2022.

Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *arXiv preprint arXiv:2101.04562*, 2021.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer, 2011.

Yuansheng Zhou and Tatyana O Sharpee. Hyperbolic geometry of gene expression. *Iscience*, 24(3): 102225, 2021.

Yuansheng Zhou, Brian H Smith, and Tatyana O Sharpee. Hyperbolic geometry of the olfactory space. *Science advances*, 4(8):eaaq1458, 2018.

A On Distortion Loss

In Section 4, we propose to use a distrotion preserving metric to validify distance measuring in the latent space. Empirically, since Equation (7) is defined through averaging, the metric is not robust to extreme pairwise distances.

A proxy measure we call *scaled* distortion is helpful in robustifying against these outliers. Likewise, consider a mapping $f : (X, d_X) \mapsto (Z, d_Z)$:

$$\begin{aligned} &D_{\text{scaled}}(X, Z) \\ &= \frac{1}{n(n-1)} \sum_{a \neq b \in X} \left(\frac{d_Z(f(a), f(b))}{\frac{1}{n(n-1)} \sum_{a' \neq b' \in X} d_Z(f(a'), f(b'))} - \frac{d_X(a, b)}{\frac{1}{n(n-1)} \sum_{a' \neq b' \in X} d_X(a', b')} \right)^2 \end{aligned} \tag{8}$$

This is controlling the l_2 difference between the input distance pairs and the embedding distances pairs but with each item normalized by sum in respective domain. Optimizing with this metric typically brings lower distortion in the original sense. We therefore train the neural network using Equation (8) but report the distortion defined by Equation (7).

B GeodesicLayer

In Section 4, we introduce HAE with a geodesic layer to send Lorentz features back to Euclidean. This works as a Hyperbolic Linear layer and the layer describe Lorentz hyperplanes to measure the distances from samples to these hyperplanes. More details can be found in [Mishne et al., 2022].

We first introduce the notion of Lorentz hyperplane. Consider $\boldsymbol{p} \in \mathbb{L}^n, \boldsymbol{w} \in \mathcal{T}_{\boldsymbol{p}}\mathbb{L}^n$, the Lorentz hyperplane passing through $\boldsymbol{p}$ perpandicular to $\boldsymbol{w}$ is reparametrized by $a \in \mathbb{R}, \boldsymbol{z} \in \mathbb{R}^n$ and given by

$$H_{\boldsymbol{z},a} = \{x \in \mathbb{L}^n | \cosh(a)\langle z, \boldsymbol{x_r}\rangle = \sinh(a)\|z\|x_0\} \tag{9}$$

where $\boldsymbol{x}_r = (x_1, \ldots, x_n)$ denotes the latter n components of $\boldsymbol{x} = (x_0, x_1, \ldots, x_n)$. The geometric meaning of this parametrization is as follows: the hyperplane is passing through a point $\boldsymbol{p}$ which is $|a|$ far away from the origin and along the direction $\boldsymbol{z}$, and is perpendicular to a vector $\boldsymbol{w}$ which has length $\|\boldsymbol{z}\|$.

The normed distance (mimicking $Ax + b$ in the linear layer) between a point x and the hyperplane can be derived by

$$d(\boldsymbol{x}, H_{\boldsymbol{z},a}) = \text{arcsinh}\left(-\frac{[\boldsymbol{x}, \boldsymbol{w}]}{\|\boldsymbol{w}\|_{\mathbb{L}^n}}\right)\|\boldsymbol{w}\|_{\mathbb{L}^n} \tag{10}$$

If the geodesic layer is mapping from $\mathbb{L}^d$ to $\mathbb{R}^m$, then there are m such hyperplanes to train, and each hyperplane yields a normed distance defined in Equation (10). These sets of $\boldsymbol{z}, a$ are thus trained during optimizing the reconstruction and distortion loss.