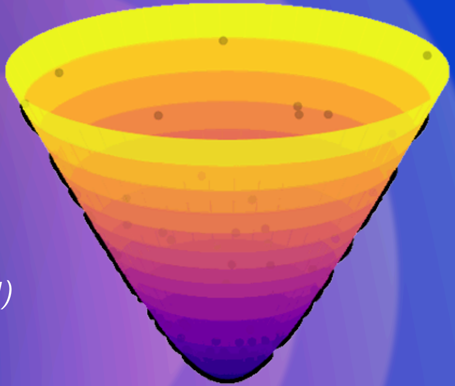
Hyperbolic Learning for Medical Imaging

Deep Learning in Hyperbolic Space

Alvaro Gonzalez-Jimenez
& Simone Lionetti

International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)

23 September 2025



Universität Basel
Universitätsspital Basel
HSLU Hochschule Luzern
University of Plymouth
MICCAI2025 Daejeon Republic of Korea

HyperMI

Website and material:

`https://hyperbolic-miccai.github.io`

Hyperbolic @ MICCAI

- MNM: Multi-level Neuroimaging Meta-analysis with Hyperbolic Brain-Text Representations — B41 (Wed-PM)
- Is Hyperbolic Space All You Need for Medical Anomaly Detection? — A292 (Thu-AM)
- HyperPath: Knowledge-Guided Hyperbolic Semantic Hierarchy Modeling for WSI Analysis — A326 (Wed-PM)
- Hyperbolic Kernel GCN with Structure-Function Connectivity Coupling for Neurocognitive Impairment Analysis — B205 (Fri-PM)
- Fine-Grained Rib Fracture Diagnosis with Hyperbolic Embeddings: A Detailed Annotation Framework and Multi-Label Classification Model — C128 (Thu-PM)

Embedding Space Choices

The embedding space is crucial for a model to faithfully represent relationships between data points.

Should Euclidean geometry remain the de facto choice for deep learning models

Issues with Euclidean Embeddings: Distorsion

Euclidean space leads to **significant distorsion** regardless of the embedding dimension.

Theorem

(Informal; Lee, Naor, and Peres, "Trees and Markov Convexity") There is a lower bound in the minimal distorsion of embedding hierarchical structures (e.g. token relationships) into Euclidean space ($\mathbb{R}^n$)

Issues with Euclidean Embeddings: Distortion

Euclidean space leads to **significant distortion** regardless of the embedding dimension.

Theorem

(Informal; Lee, Naor, and Peres, "Trees and Markov Convexity") There is a lower bound in the minimal distortion of embedding hierarchical structures (e.g. token relationships) into Euclidean space ($\mathbb{R}^n$)

"There is a **performance bottleneck** *on how well Euclidean models can represent complex token relationships"*

Issues with Euclidean Embeddings: Distorsion Dilemma

Euclidean space face the dilemma of **dimension-distorsion tradeoffs**.
High dimensionality is often required to embed complex token relations in Euclidean space with (relatively) low distorision.

Theorem

(Informal; *Lectures on Discrete Geometry*) The dimension required when embedding unweighted graphs (in the form of token relationships/self-attention) grows **near-quadratically** w.r.t. to distorsion.

Issues with Euclidean Embeddings: Distorsion Dilemma

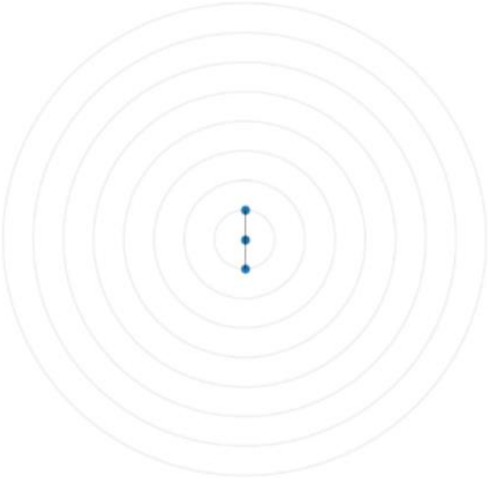
Euclidean space face the dilemma of **dimension-distorsion tradeoffs**.
High dimensionality is often required to embed complex token relations in Euclidean space with (relatively) low distorision.

Theorem

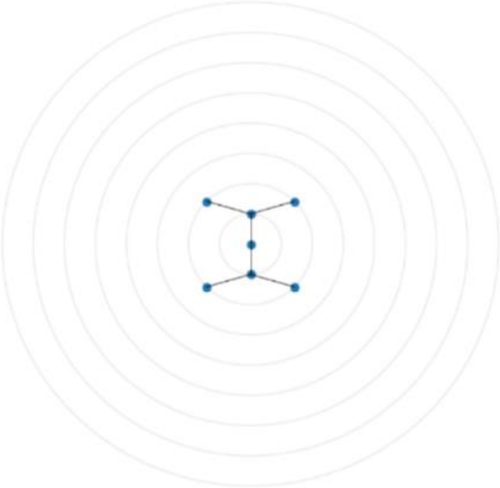
(Informal; *Lectures on Discrete Geometry*) The dimension required when embedding unweighted graphs (in the form of token relationships/self-attention) grows **near-quadratically** w.r.t. to distorsion.

"Euclidean models have **limited scalability***"*

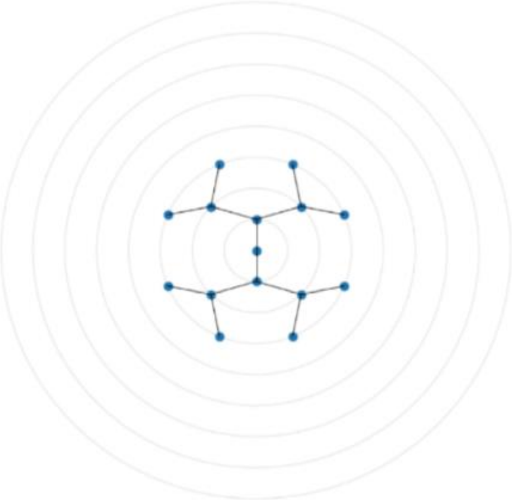
Example: Embedding Tree-structured Data



Example: Embedding Tree-structured Data

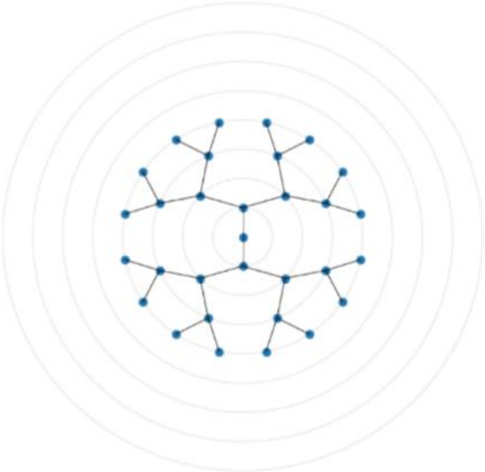


Example: Embedding Tree-structured Data

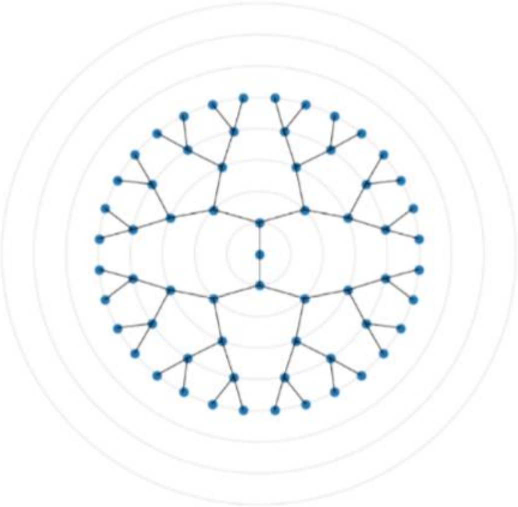


So far, so good. Nodes are close **i.f.f.they are connected by an edge**.

Example: Embedding Tree-structured Data



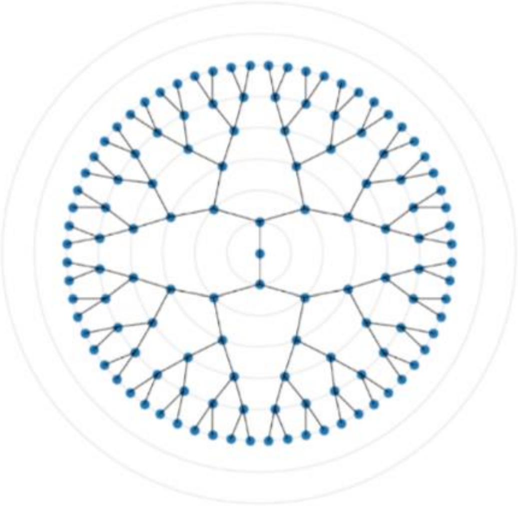
Example: Embedding Tree-structured Data



But the outermost nodes are becoming increasingly close to one another.

Even though they are not connected by an edge in the graph

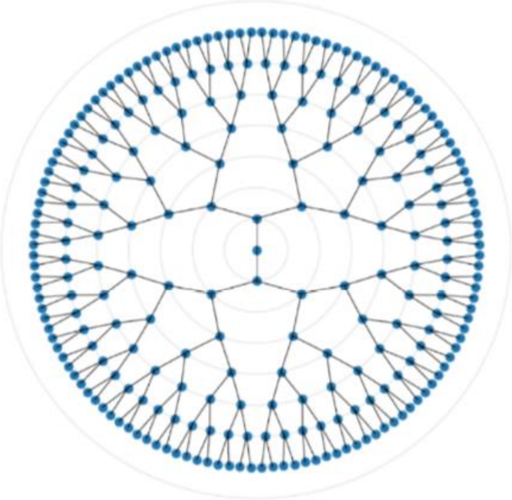
Example: Embedding Tree-structured Data



But the outermost nodes are becoming increasingly close to one another.

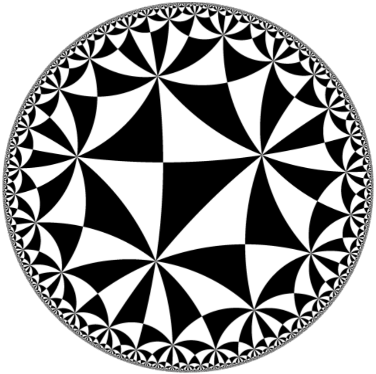
Even though they are not connected by an edge in the graph

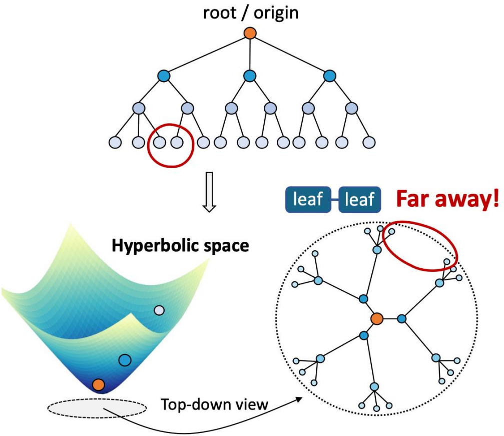
Example: Embedding Tree-structured Data



Things only get worse!
We have lost our property
"close i.f.f share edge"

Potential Solution: Hyperbolic Embedding Space





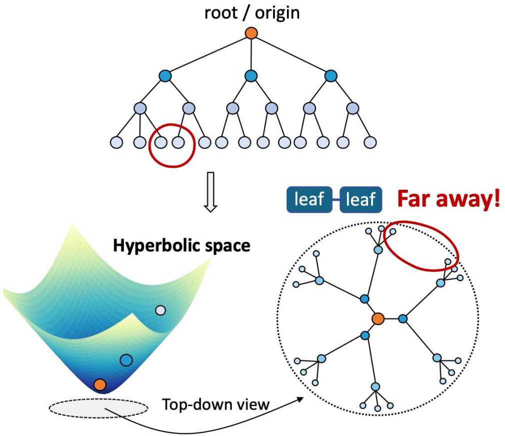
The volume of a ball in the hyperbolic space grows **exponentially** with its radius

Hyperbolic Geometry for Deep Learning

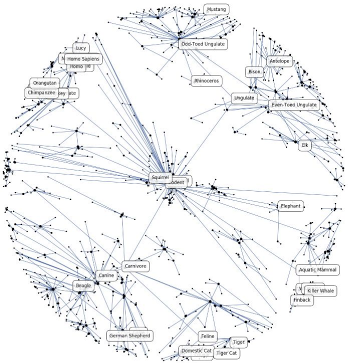
We need an embedding space that can **better represent token relationship**

- The distance between low-level tokens on different branches should be maximized and far away
- The distance between a high-level token and a low-level token should be minimized and close.l

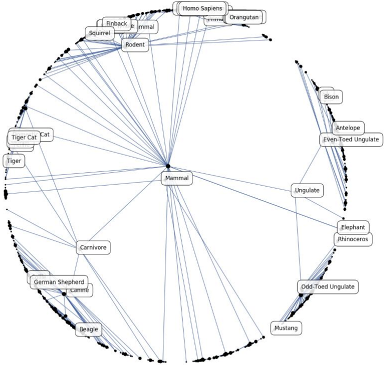
Solution: any tree (i.e. hierarchical distribution) can be embedded into **hyperbolic space** with **arbitrarily low distorsion**.



Poincaré Embeddings



(a) Intermediate embedding after 20 epochs



(b) Embedding after convergence

Nickel and Kiela, *Poincaré Embeddings for Learning Hierarchical Representations*

Optimizing Poincaré Embeddings

Nodes:
$\mathcal{S} = \{x_i\}_{i=1}^{n}$

Parent-child relations:
$\mathcal{D} = \{(u, v)\}$

Non-Parent-child relations:
$\mathcal{N}(u) = \{v' | (u, v') \notin \mathcal{D}\} \cup \{v\}$

Optimizing Poincaré Embeddings

Nodes:
$\mathcal{S} = \{x_i\}_{i=1}^{n}$

Parent-child relations:
$\mathcal{D} = \{(u, v)\}$

Non-Parent-child relations:
$\mathcal{N}(u) = \{v' | (u, v') \notin \mathcal{D}\} \cup \{v\}$

Hyperbolic representation of nodes: $\Theta = \{\theta_i\}_{i=1}^{n}$

$$\Theta' \leftarrow \operatorname{argmin} \mathcal{L}(\Theta) \quad \text{s.t.} \forall \theta_i \in \Theta : \|\theta_i\| < 1 \tag{1}$$

Pull parent-child nodes, push others.

$$\mathcal{L}(\Theta) = \sum_{(u,v) \in \mathcal{D}} \log \frac{e^{-d(\mathbf{u},\mathbf{v})}}{\sum_{\mathbf{v}' \in \mathcal{N}(u)} e^{-d(\mathbf{u},\mathbf{v}')}} \tag{2}$$

$$d(\mathbf{u}, \mathbf{v}) = \operatorname{arcosh}\left(1 + 2\frac{\|\mathbf{u} - \mathbf{v}\|^2}{\left(1 - \|\mathbf{u}\|^2\right)\left(1 - \|\mathbf{v}\|^2\right)}\right) \tag{3}$$

Optimizing Poincaré Embeddings

$$\theta_{t+1} = \Re_{\theta_t}\left(-\eta_t \nabla_R \mathcal{L}\left(\theta_t\right)\right) \tag{4}$$

Optimize node embeddings with Riemmanian gradient descent.

$$\theta_{t+1} \leftarrow \mathrm{proj}\left(\theta_t - \eta_t \frac{\left(1 - \|\theta_t\|^2\right)^2}{4} \nabla_E\right) \tag{5}$$

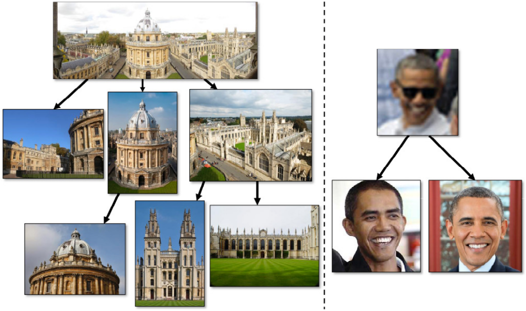
Riemmanian gradient descent = Standard gradient + scaling + projection.

Poincaré Embeddings

			Dimensionality					
			5	10	20	50	100	200
WordNet Reconstruction	**Euclidean**	Rank	3542.3	2286.9	1685.9	1281.7	1187.3	1157.3
		MAP	0.024	0.059	0.087	0.140	0.162	0.168
	Translational	Rank	205.9	179.4	95.3	92.8	92.7	91.0
		MAP	0.517	0.503	0.563	0.566	0.562	0.565
	Poincaré	Rank	4.9	4.02	3.84	3.98	3.9	**3.83**
		MAP	0.823	0.851	0.855	0.86	0.857	**0.87**
WordNet Link Pred.	**Euclidean**	Rank	3311.1	2199.5	952.3	351.4	190.7	81.5
		MAP	0.024	0.059	0.176	0.286	0.428	0.490
	Translational	Rank	65.7	56.6	52.1	47.2	43.2	40.4
		MAP	0.545	0.554	0.554	0.56	0.562	0.559
	Poincaré	Rank	5.7	**4.3**	4.9	4.6	4.6	4.6
		MAP	0.825	0.852	0.861	**0.863**	0.856	0.855

		Dimensionality							
		Reconstruction				Link Prediction			
		10	20	50	100	10	20	50	100
AstroPh N=18,772; E=198,110	**Euclidean**	0.376	0.788	0.969	0.989	0.508	0.815	0.946	0.960
	Poincaré	0.703	0.897	0.982	0.990	0.671	0.860	0.977	0.988
CondMat N=23,133; E=93,497	**Euclidean**	0.356	0.860	0.991	0.998	0.308	0.617	0.725	0.736
	Poincaré	0.799	0.963	0.996	0.998	0.539	0.718	0.756	0.758
GrQc N=5,242; E=14,496	**Euclidean**	0.522	0.931	0.994	0.998	0.438	0.584	0.673	0.683
	Poincaré	0.990	0.999	0.999	0.999	0.660	0.691	0.695	0.697
HepPh N=12,008; E=118,521	**Euclidean**	0.434	0.742	0.937	0.966	0.642	0.749	0.779	0.783
	Poincaré	0.811	0.960	0.994	0.997	0.683	0.743	0.770	0.774

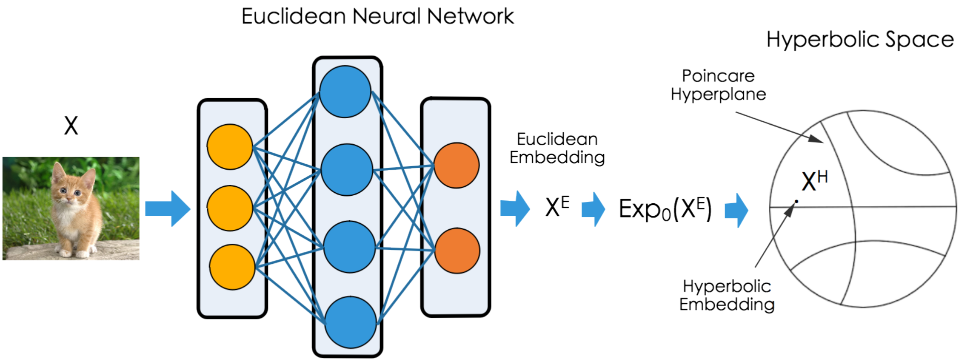
Hyperbolic Image Embeddings



Encoder	Dataset			
	CIFAR10	CIFAR100	CUB	*Mini*ImageNet
Inception v3 [49]	0.25	0.23	0.23	0.21
ResNet34 [14]	0.26	0.25	0.25	0.21
VGG19 [42]	0.23	0.22	0.23	0.17

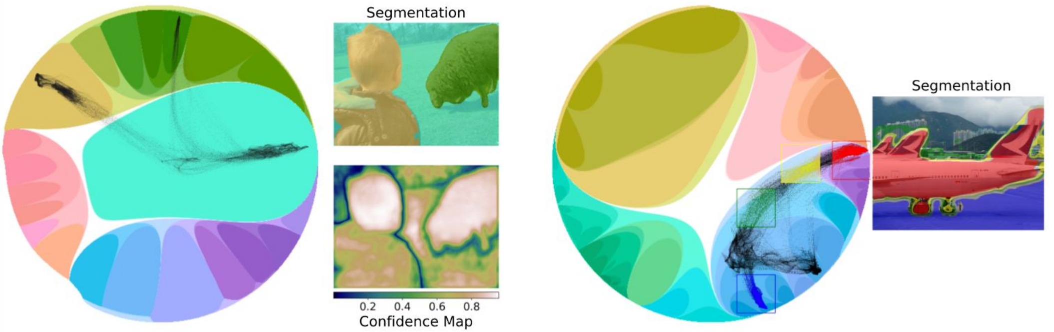
Khrulkov et al., *Hyperbolic Image Embeddings*

Convolutional Networks with Hyperbolic Embeddings



Guo et al., "Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers"

Hyperbolic Segmentation

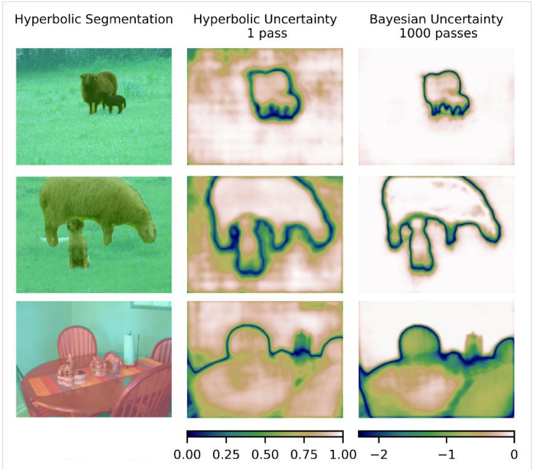


(a) Prediction uncertainty for free

(b) Boundary information for free

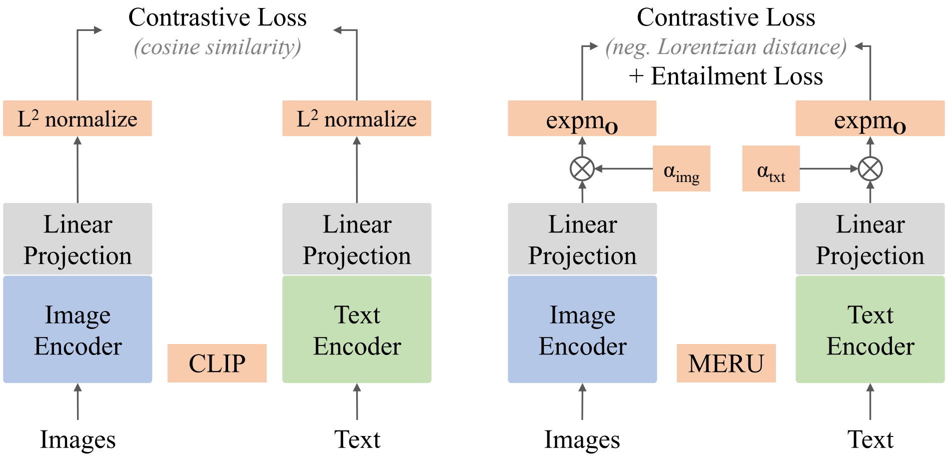
GhadimiAtigh et al., *Hyperbolic Image Segmentation*

Uncertainty and boundary information for free



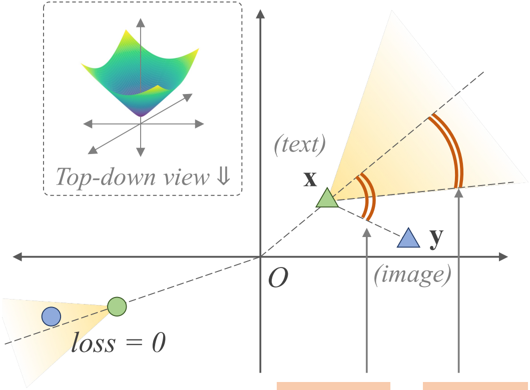
GhadimiAtigh et al., *Hyperbolic Image Segmentation*

Hyperbolic Image-Text Representations



Desai et al., "Hyperbolic Image-text Representations"

Entailment Cones



$$loss = \boxed{\mathrm{ext}(\mathbf{x}, \mathbf{y})} - \boxed{\mathrm{aper}(\mathbf{x})}$$

Desai et al., "Hyperbolic Image-text Representations"

MERU

MERU	CLIP
avocado toast	*avocado toast*
healthy breakfast	*delicious*
delicious	$\downarrow$
homemade	$\downarrow$
fresh	$\downarrow$
[ROOT]	[ROOT]

MERU	CLIP
brooklyn bridge	*photo of brooklyn bridge, new york*
new york city	*new york city*
city	*new york*
outdoors	$\downarrow$
day	$\downarrow$
[ROOT]	[ROOT]

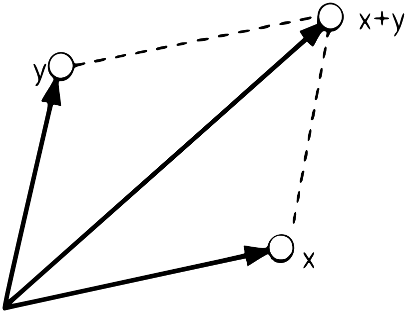
MERU	CLIP
taj mahal	*taj mahal through an arch*
monument	*travel*
architecture	*inspiration*
travel	$\downarrow$
day	$\downarrow$
[ROOT]	[ROOT]

MERU	CLIP
sydney opera house	*sydney opera house*
opera house	*opera house*
holiday	*gift*
day	*beauty*
[ROOT]	[ROOT]

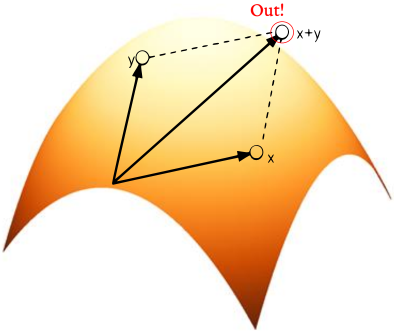
Desai et al., "Hyperbolic Image-text Representations"

Hyperbolic Operations: Difficulties

Addition in Euclidean Space



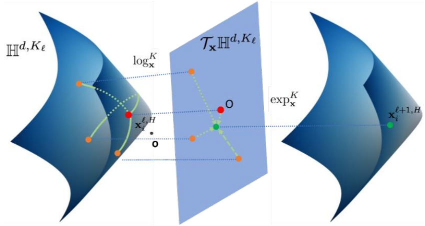
Addition in Hyperbolic Space?



Considerations:

1. Satisfy manifold constraints.
2. Satisfy neural operation properties.

Strategy 1: Tangent-Space Based Operations



Idea:

Map input to the tangent space at the origin, so f is a valid operation. Then perform Euclidean operation and finally lift the output back to $\mathbb{H}^{d,K}$:

$$f^{T,K}(x) = \exp_{\mathbf{o}}^K\left(f\left(\log_{\mathbf{o}}^K(x)\right)\right) \quad (6)$$

Chami et al., "Hyperbolic Graph Convolutional Neural Networks"

Strategy 1: Cons

Computational Inefficiency

The repeated mappings to and from the tangent space cause significant computational overhead.

Numerical Instability

The mappings could cause numerical stability issues. e.g. in the logarithmic map:

$$\log_x^K(y) = \mathcal{D}_{\mathbb{L}}^K(x, y) \frac{y + \frac{1}{K} \langle x, y \rangle_{\mathbb{L}} \, x}{\left\| y + \frac{1}{K} \langle x, y \rangle_{\mathbb{L}} \, x \right\|_{\mathbb{L}}} \tag{7}$$

if the points are close together, we risk dividing by or calling `arccosin` on 0.

Strategy 2: Fully Hyperbolic Operations

Solution: operate directly on the manifold "Fully Hyperbolic"
Two strategies: *Pseudo Lorentz Rotation* vs *Pseudo Lorentz Boost*

Strategy 2: Fully Hyperbolic Operations

Solution: operate directly on the manifold "Fully Hyperbolic"
Two strategies: *Pseudo Lorentz Rotation* vs *Pseudo Lorentz Boost*

Pseudo Lorentz Boost

1. Use euclidean function $f : \mathbb{R}^{d+1} \rightarrow \mathbb{R}^{d+1}$ e.g. Linear transformation $f(x) = Wx + b$.
2. Perform f on $x \in \mathbb{H}^{d,K}$ $\rightarrow$ Transformation on **both** time and space components.
3. Compute the associating time-like dimensions $\rightarrow$ Impose Lorentzian constraints.

$$f^{F,K}(x) = \left(\underbrace{\sqrt{\|Wx_{\text{time, space}}\|^2 - \frac{1}{K}}}_{\text{time-like dim}}, \quad \underbrace{Wx_{\text{time, space}}}_{\text{space-like dim}} \right) \tag{8}$$

Strategy 2: Fully Hyperbolic Operations

Solution: operate directly on the manifold "Fully Hyperbolic"
Two strategies: *Pseudo Lorentz Rotation* vs *Pseudo Lorentz Boost*

Pseudo Lorentz Rotation

1. Use euclidean function $f : \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$ e.g. $f(x) = ReLU(x)$.
2. Perform f on the space-like dimension of $x \in \mathbb{H}^{d,K}$ $\to$ Transformation on **only** space dimension.
3. Compute the associating time-like dimensions $\to$ Impose Lorentzian constraints.

$$f^{F,K}(x) = \left(\underbrace{\sqrt{\|Wf(x_{\text{space}})\|^2 - \frac{1}{K}}}_{\text{time-like dim}}, \quad \underbrace{f(x_{\text{space}})}_{\text{space-like dim}} \right) \tag{9}$$

Strategy Comparison

Pseudo Lorentz Rotation: Transformation on without time and space interaction.

$$\begin{pmatrix} \frac{\sqrt{\|f(x_{\text{space}})\|^2 - 1/K}}{x_{\text{time}}} & 0 \\ 0 & f(\cdot) \end{pmatrix} \binom{x_{\text{time}}}{x_{\text{space}}}$$

Off-diagonal values are zero

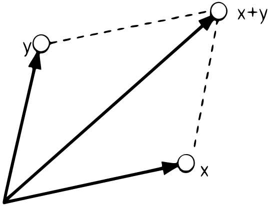
Pseudo Lorentz Boost: Transformation on both time and space-like dimension.

$$\begin{pmatrix} \sqrt{\|Wx\|^2 - 1/K}\mathbf{e}_0, & W_{0,:} \\ \sqrt{\|Wx\|^2 - 1/K}\mathbf{e}_{1:d'} & W_{1:,:} \end{pmatrix} \binom{x_{\text{time}}}{x_{\text{space}}}$$

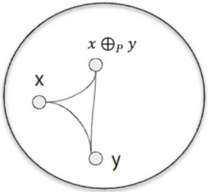
Non-zero off-diagonal values

Hyperbolic Residual Connection & Addition

Vector space formulation



Gyrovector space formulation



Mobius Addition

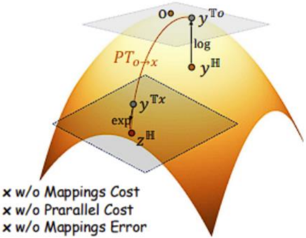
Tangent-Space based method: Mobius addition based on parallel transport:

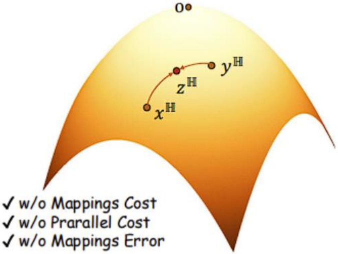
$$x \oplus_P y = \exp_x^K(P_{\mathbf{o}\to x}(\log_{\mathbf{o}}^K(y))) \tag{10}$$

Generalized Lorentz Weighted Sum

More efficient, stable and expressive.

$$
\begin{aligned}
x \oplus_L y &= \alpha x + \beta y \\
\text{where} \quad \alpha &= \frac{W_x}{\sqrt{-K}\,\|W_x x + W_y y\|_{\mathbb{L}}}, \quad \beta = \frac{W_y}{\sqrt{-K}\,\|W_x x + W_y y\|_{\mathbb{L}}} \\
\text{with} \quad W_x&, W_y > 0
\end{aligned} \tag{11}
$$





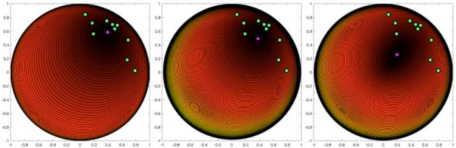
Hyperbolic Midpoint Operations

Hyperbolic midpoints have closed-form expressions in the Lorentz, Poincaré, and Einstein models, and are **equivalent under isometric mappings**.

$$\mathrm{LMid}_K(x_1, \ldots, x_N; v_i) = \frac{\sum_j v_j x_j}{\sqrt{-K} \left\| \sum_j v_j x_j \right\|_{\mathbb{L}}}$$

$$\mathrm{PMid}_K(x_1, \ldots, x_N; v_i) = \frac{1}{2} \otimes_K \frac{\sum_j v_j \lambda_{x_j}^K x_j}{\sum_j |v_j| (\lambda_{x_j}^K - 1)}$$

$$\lambda_x^K = \frac{2}{1 + K\|x\|^2}$$



[a]

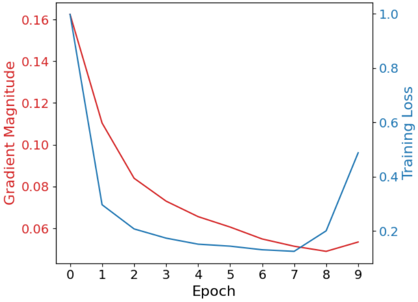
[a] Law et al., "Lorentzian Distance Learning for Hyperbolic Representations"

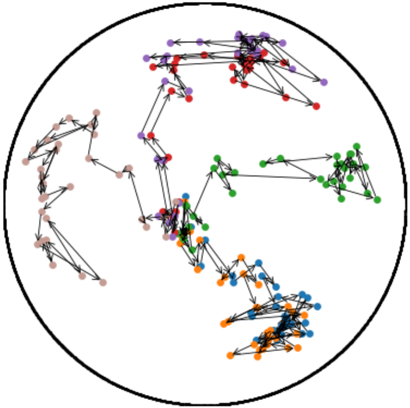
Hyperbolic Self-Attention

Hyperbolic self-attention can be formulated with hyperbolic midpoint operations and similarity score computed using the negative hyperbolic distance.

$$
\begin{aligned}
\mathrm{LAtten}(Q, K, V) &= \mathrm{LMid}(v_1, \cdots, v_N, \{\alpha_{i,j}\}_{j=1}) \\
\mathrm{PAtten}(Q, K, V) &= \mathrm{PMid}(v_1, \cdots, v_N, \{\alpha_{i,j}\}_{j=1}) \\
\alpha_{i,j} &= \frac{\exp(-d_{\mathbb{H}}^2(q_i, v_j))}{\sum_l \exp(-d_{\mathbb{H}}^2(q_l, v_l))}
\end{aligned}
$$

Gradients Vanishing





Guo et al., “Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers”

Summary

- Euclidean embeddings struggle with distortion and scalability in hierarchical settings.
- Hyperbolic embeddings boost performance in tasks like image classification and segmentation.
- Fully hyperbolic models avoid tangent-space mappings but require specialized operations.
- Gradient vanishing and numerical instability remain open challenges in hyperbolic learning.

References I

- Chami, Ines et al. "Hyperbolic Graph Convolutional Neural Networks". In: *Neural Information Processing Systems* (2019).
- Desai, Karan et al. "Hyperbolic Image-text Representations". In: *Proceedings of the 40th International Conference on Machine Learning*. PMLR, July 2023, pp. 7694–7731. (Visited on 02/05/2025).
- GhadimiAtigh, Mina et al. *Hyperbolic Image Segmentation*. Mar. 2022. DOI: 10.48550/arXiv.2203.05898. (Visited on 02/05/2025).
- Guo, Yunhui et al. "Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers". In: *Computer Vision and Pattern Recognition* (2021). DOI: 10.1109/CVPR52688.2022.00010.
- Khrulkov, Valentin et al. *Hyperbolic Image Embeddings*. Mar. 2020. DOI: 10.48550/arXiv.1904.02239. (Visited on 02/05/2025).

References II

- Law, Marc et al. "Lorentzian Distance Learning for Hyperbolic Representations". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 3672–3681. URL: https://proceedings.mlr.press/v97/law19a.html.
- *Lectures on Discrete Geometry*. Springer New York, 2002. ISBN: 9781461300397. DOI: 10.1007/978-1-4613-0039-7. URL: http://dx.doi.org/10.1007/978-1-4613-0039-7.
- Lee, James R., Assaf Naor, and Yuval Peres. "Trees and Markov Convexity". In: *Geometric and Functional Analysis* 18.5 (Dec. 2008), pp. 1609–1659. ISSN: 1420-8970. DOI: 10.1007/s00039-008-0689-0. URL: http://dx.doi.org/10.1007/s00039-008-0689-0.

References III

Nickel, Maximilian and Douwe Kiela. *Poincaré Embeddings for Learning Hierarchical Representations*. May 2017. DOI: 10.48550/arXiv.1705.08039. (Visited on 02/26/2024).