Hyperbolic Networks:

Theory, Architectures and Applications

Presenters: Nurendra Choudhary[a], Nikhil Rao[b], Karthik Subbian[b], Srinivasan H Sengamedu[b], and Chandan K. Reddy[a,b]

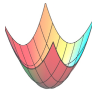
[a] Department of Computer Science, Virginia Tech, Arlington, VA
[b] Amazon, Palo Alto, CA

THE WEB CONFERENCE ACM

Outline

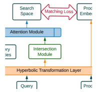
Motivation

1. Why Hyperbolic Space?
2. Where to apply?
3. How to use?



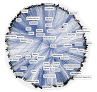
Theory

1. Introduction to Hyperbolic Space
2. Hyperbolic Operations
3. Hyperbolic MLR and activation
4. Riemannian Optimization



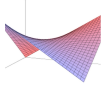
Architectures

1. Hyperbolic Linear Layer
2. Hyperbolic Recurrent Layer
3. Hyperbolic Convolution Layer
4. Hyperbolic Transformers
5. Alternative Formulations



Applications

1. Graph Applications
2. Knowledge Graph Reasoning
3. Product Search
4. Natural Language Processing



Conclusion

1. Summary
2. Challenges
3. Future Directions

Slides and other tutorial materials are available at : https://nurendra.me/hyperbolic-networks-tutorial/

Part 1: Motivation

Motivation

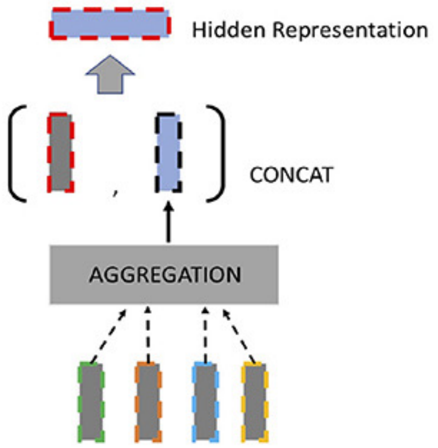
Neural Networks and Applications

Neural networks have been applied to a wide range of problems with different types of datasets such as graphs, text and images.

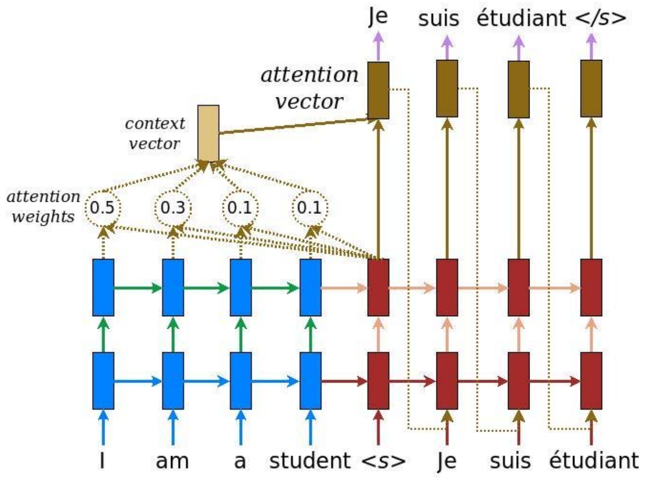
Motivation

Euclidean Spaces

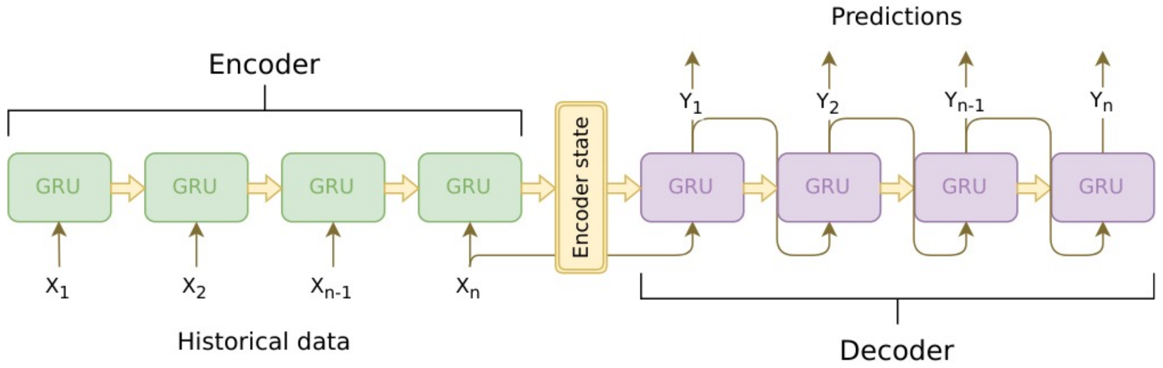
The difference in the inherent properties required to solve these tasks is handled with different types of models, generally based on Euclidean spaces.



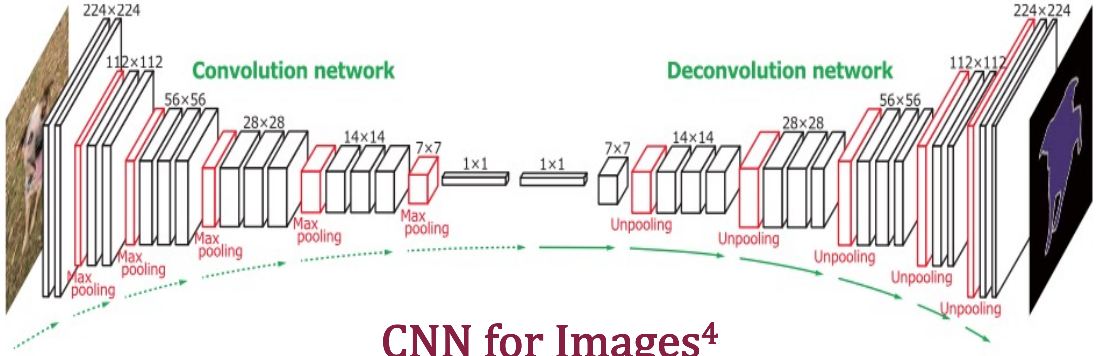
GNN for Graphs[1]



Transformers for Text[2]



RNN for Timeseries[3]



CNN for Images[4]

1. Tan, et al., "Deep representation learning for social network analysis." *Frontiers in big Data* 2 (2019):
2. Bahdanau et al., "Neural machine translation by jointly learning to align and translate." ICLR (2015).
3. Chatbot Tutorial — PyTorch Tutorials 1 (Lasylife.top).
4. https://towardsdatascience.com/review-deconvnet-unpooling-layer-semantic-segmentation-55cf8a6e380e.

Motivation
Where to apply?

- **Graph Analysis:**

Using hierarchies in graph relations.

- **Knowledge Graphs:**

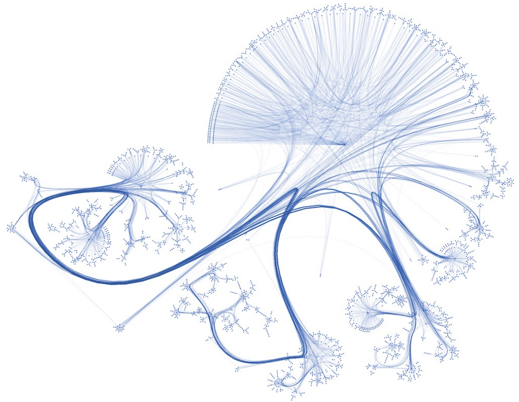
Using hierarchies in knowledge graph representation.

- **Search:**

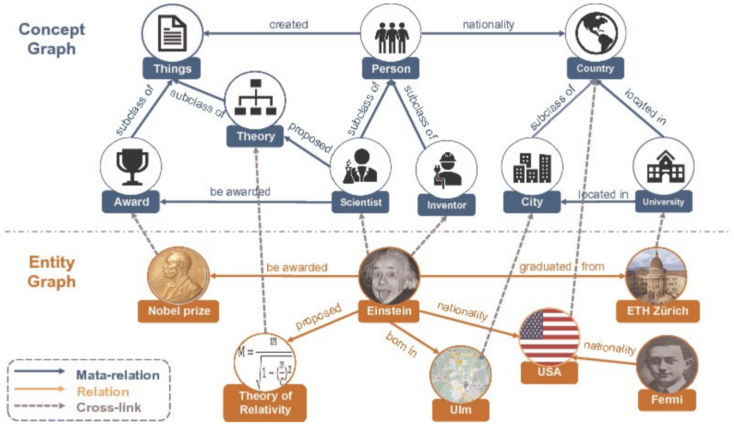
Using hierarchies in retrieval entities.

- **Natural Language Processing**

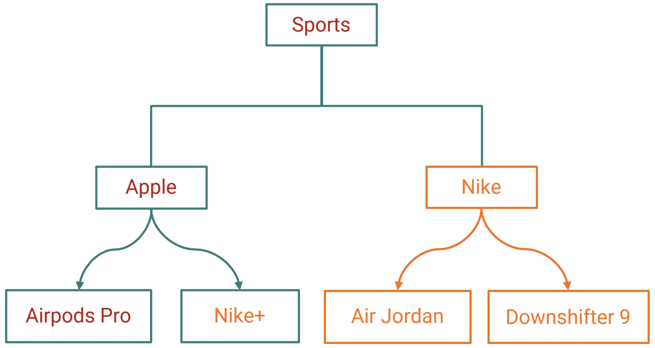
Using hierarchal relations between words of a sentence.



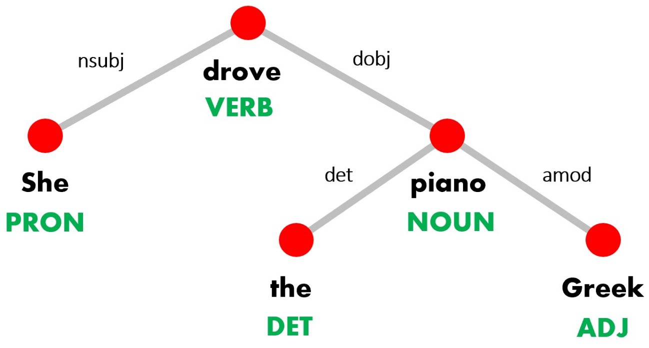
Graph Analysis[1]



Knowledge Graph Representation[2]



Product Search Catalogues



Syntax tree of a Sentence[3]

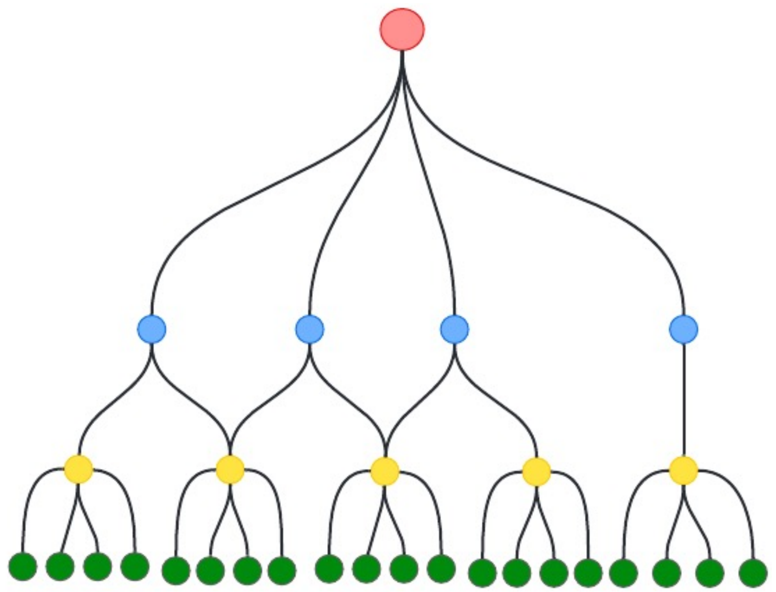
1. https://graphsandnetworks.com/category/graph-machine-learning/ [Cora Dataset]
2. Dong, J., Gu, B., Qu, J., Liu, A., Zhao, L., Chen, Z., & Li, Z. (2021, October). HyperJOIE: Two-View Hyperbolic Knowledge Graph Embedding with Entities and Concepts Jointly. In International Conference on Web Information Systems Engineering (pp. 305-320). Springer, Cham.
3. Getting to grips with parse trees, Vered Zimmerman (Towards Data Science, 2019).

Motivation

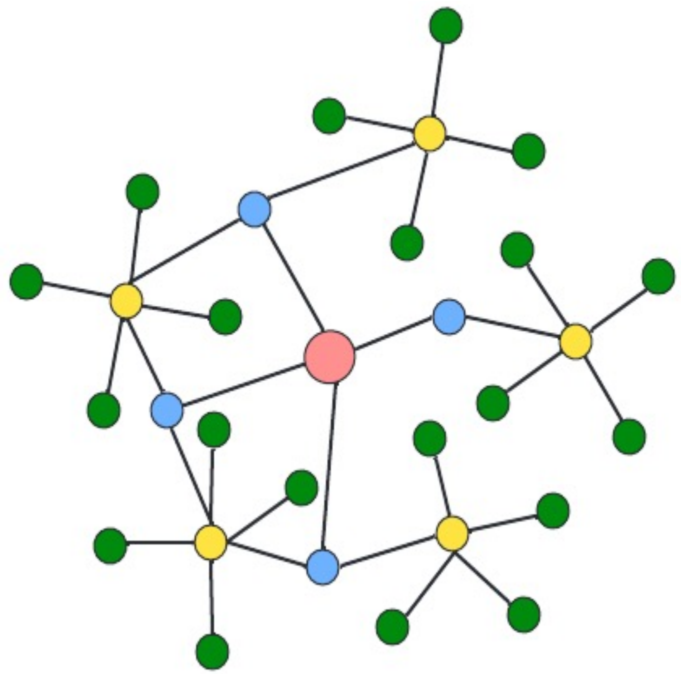
Peculiar Case of Hierarchical Problems

With increasing depth, the number of nodes grow exponentially, however Euclidean space grows linearly (L1-norm, Euclidean distance).

Thus, Euclidean representational volume is not sufficient. Points get clustered together.



Hierarchical Dataset

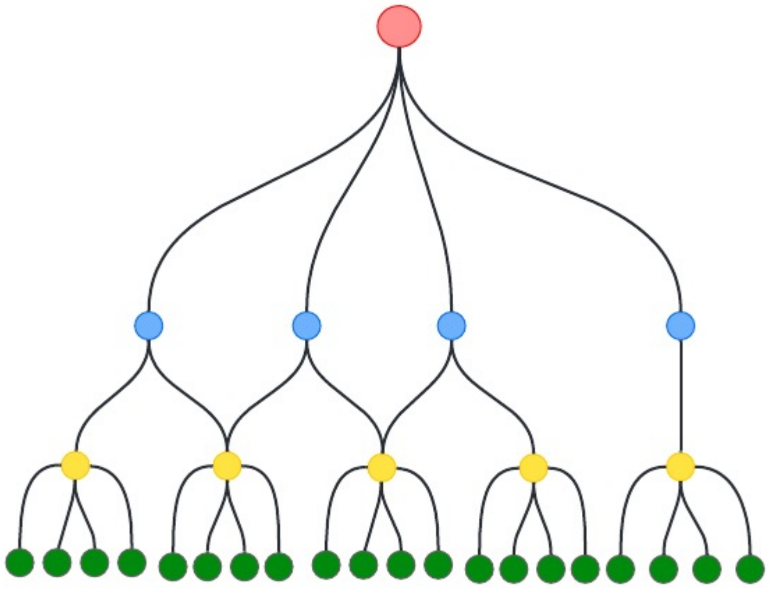


Euclidean Embeddings

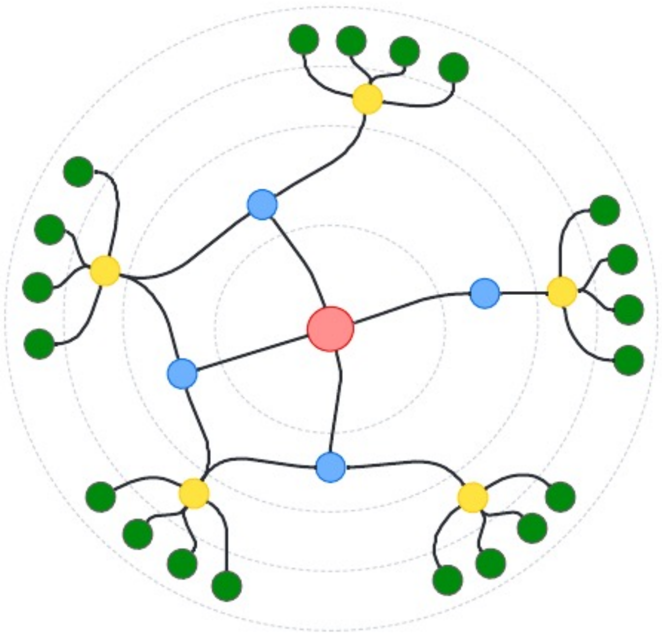
Motivation

Why Hyperbolic Space?

Thus, recent research has shifted to non-Euclidean hyperbolic spaces for capturing hierarchical dependencies in the datasets.



Hierarchical Dataset



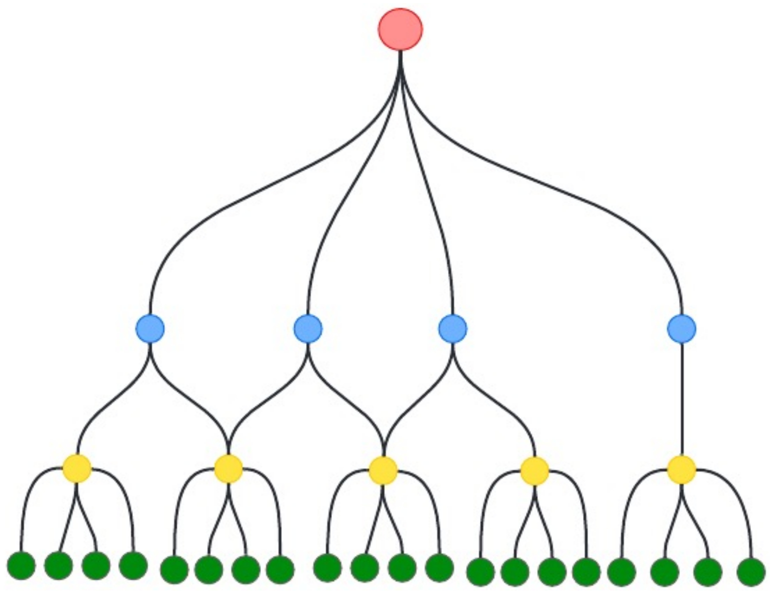
Hyperbolic Embeddings

Motivation

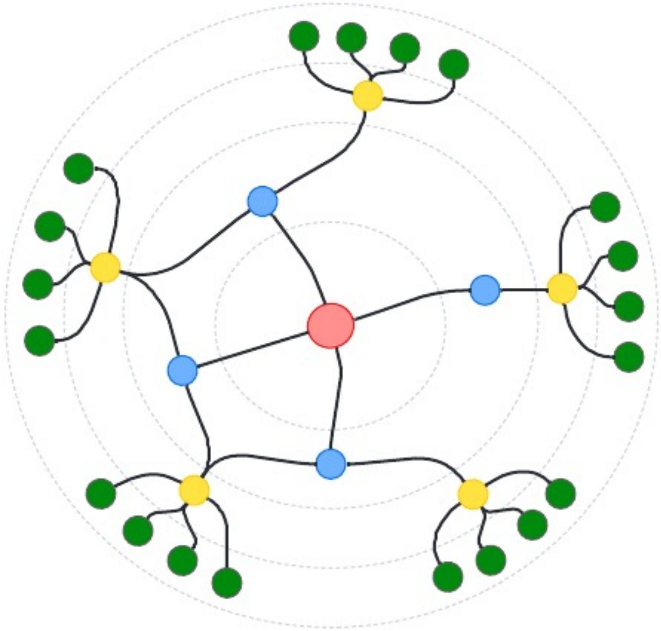
Why Hyperbolic Space?

Recently, research has shifted to non-Euclidean hyperbolic spaces for capturing hierarchical dependencies in the datasets.

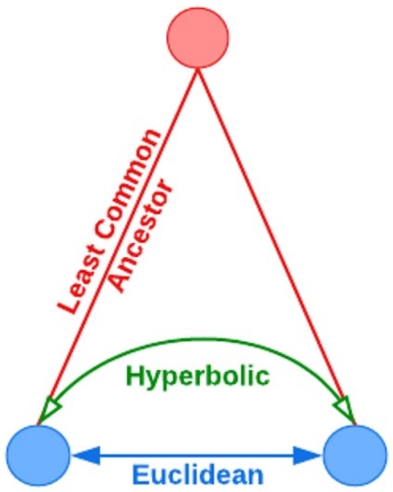
Hyperbolic spaces more closely resemble hierarchical structures than Euclidean space.



Hierarchical Dataset

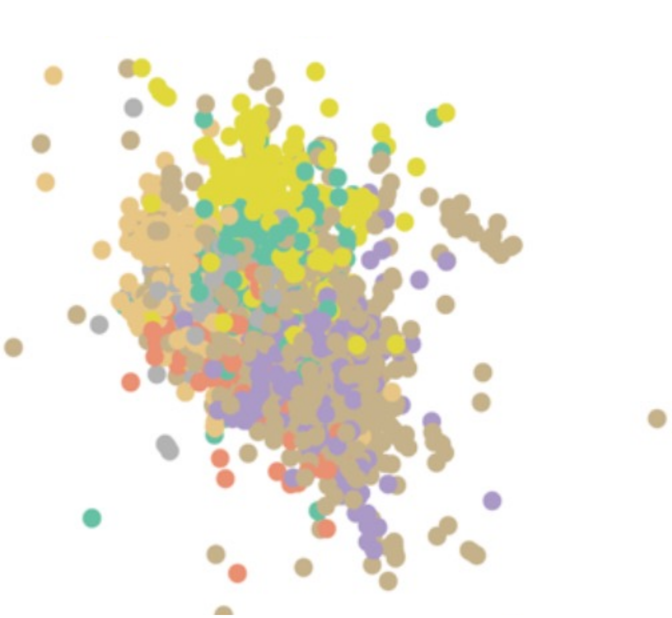


Hyperbolic Embeddings

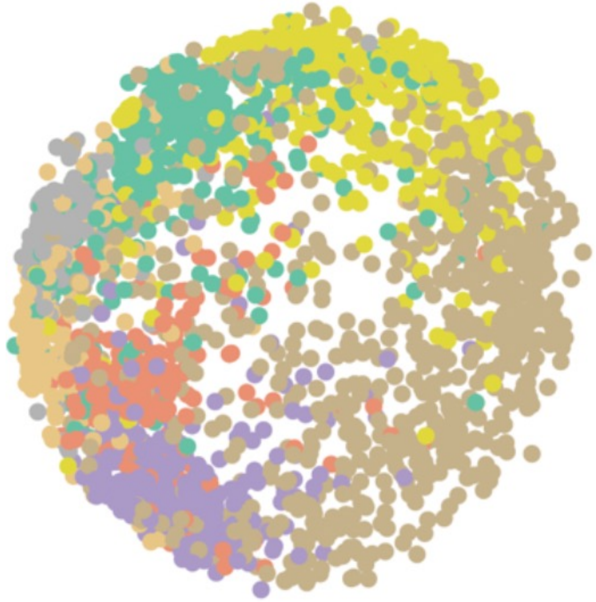


Motivation
How to use?

In such problem settings, hyperbolic models generate better representations.



Euclidean Embeddings*
(GCN, Cora dataset)



Poincaré Embeddings*
(HGCN, Cora dataset)

In the above problem of node classification, hyperbolic model HGCN generates more separable representations, compared to its Euclidean counterpart GCN.

*Chami, Ines, et al. "Hyperbolic graph convolutional neural networks." Advances in neural information processing systems 32 (2019).

Tutorial Objective

Make it easy to use

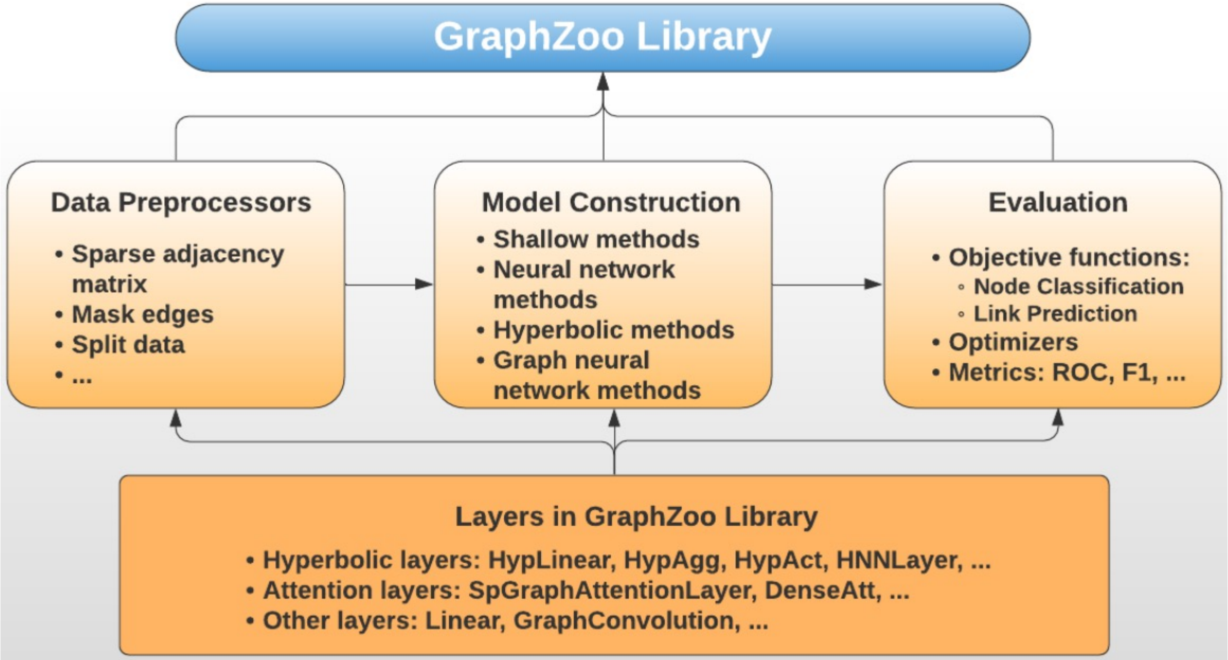
One-Stop-Shop for Hyperbolic Networks

Theoretical Underpinnings of Hyperbolic geometry

Architectural Design Choices and Implementation

Successful Application Scenarios

GraphZoo Toolkit



Tutorial Objective

Make it easy to use

In majority of the cases, to change a Euclidean architecture, we just need to:

1. Change operators

Euclidean:

$w_1 x_1 + w_2 x_2 + w_3 x_3 + \mathrm{b}$

Hyperbolic: Replace with gyrovector operations

$w_1 \otimes_c x_1 \oplus_c w_2 \otimes_c x_2 \oplus_c w_3 \otimes_c x_3 \oplus_c b$

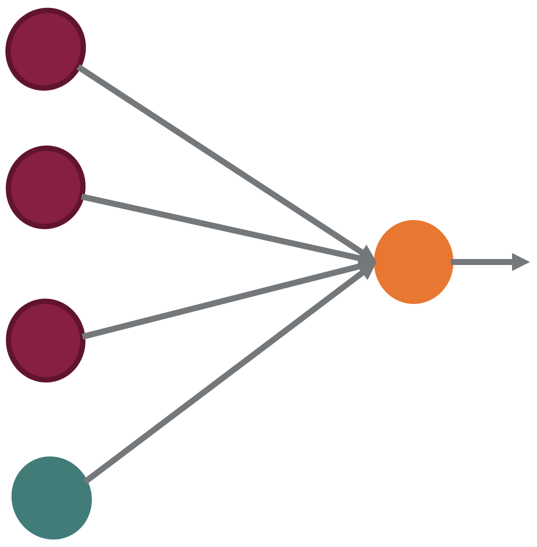
2. Call manifold versions of the Euclidean layers:

Euclidean:

nn.Linear

Hyperbolic: Replace with manifold library

manifolds.hyperbolic.Linear

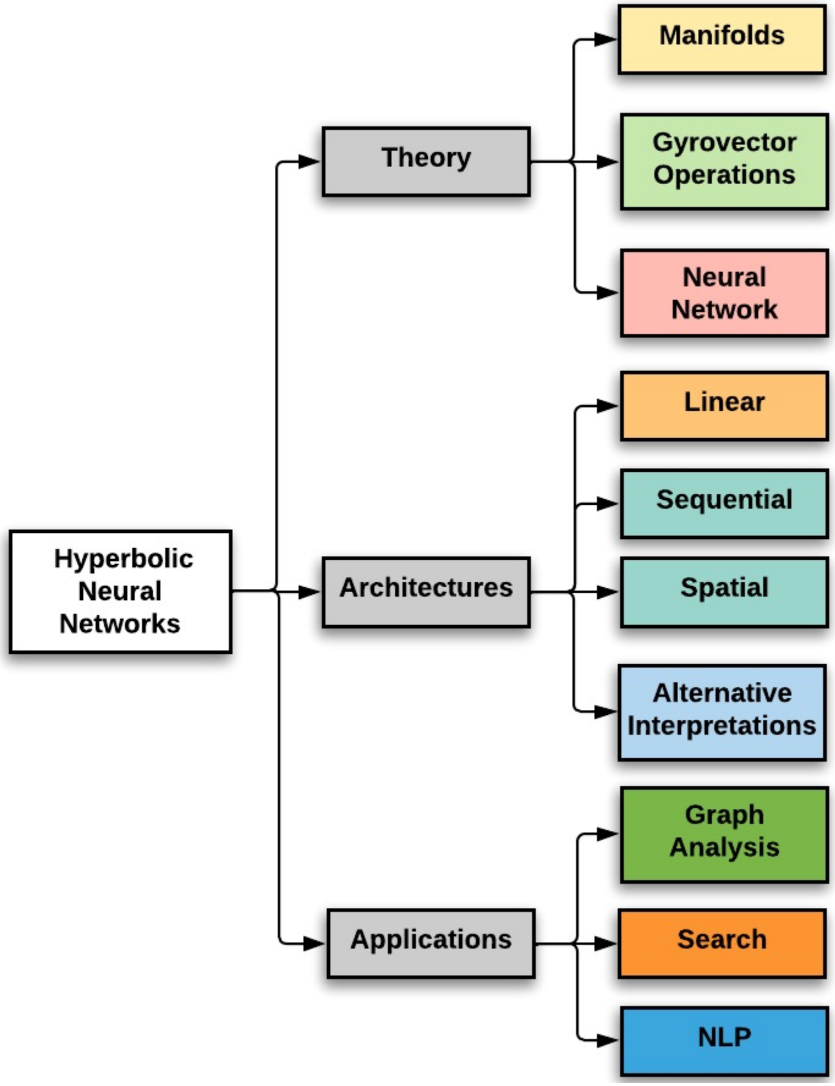


Note that the overarching network structure remains the same, only some components change.

Tutorial
Overview

Our goal is to introduce:

1) Hyperbolic geometry
2) Implementation techniques
3) Existing applications
4) Advantages over Euclidean architectures.

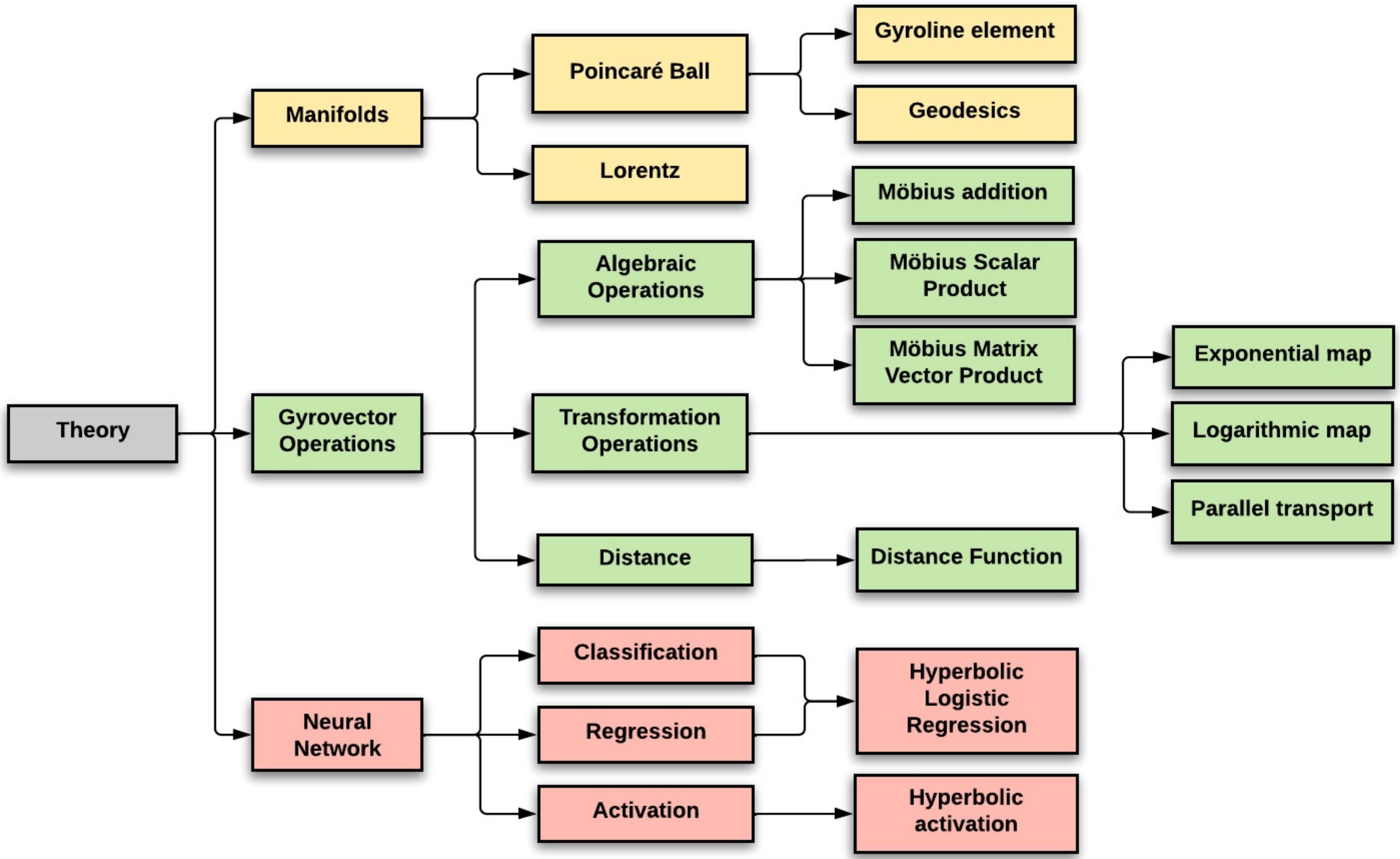


Tutorial

Details: Hyperbolic Geometry

Hyperbolic geometry

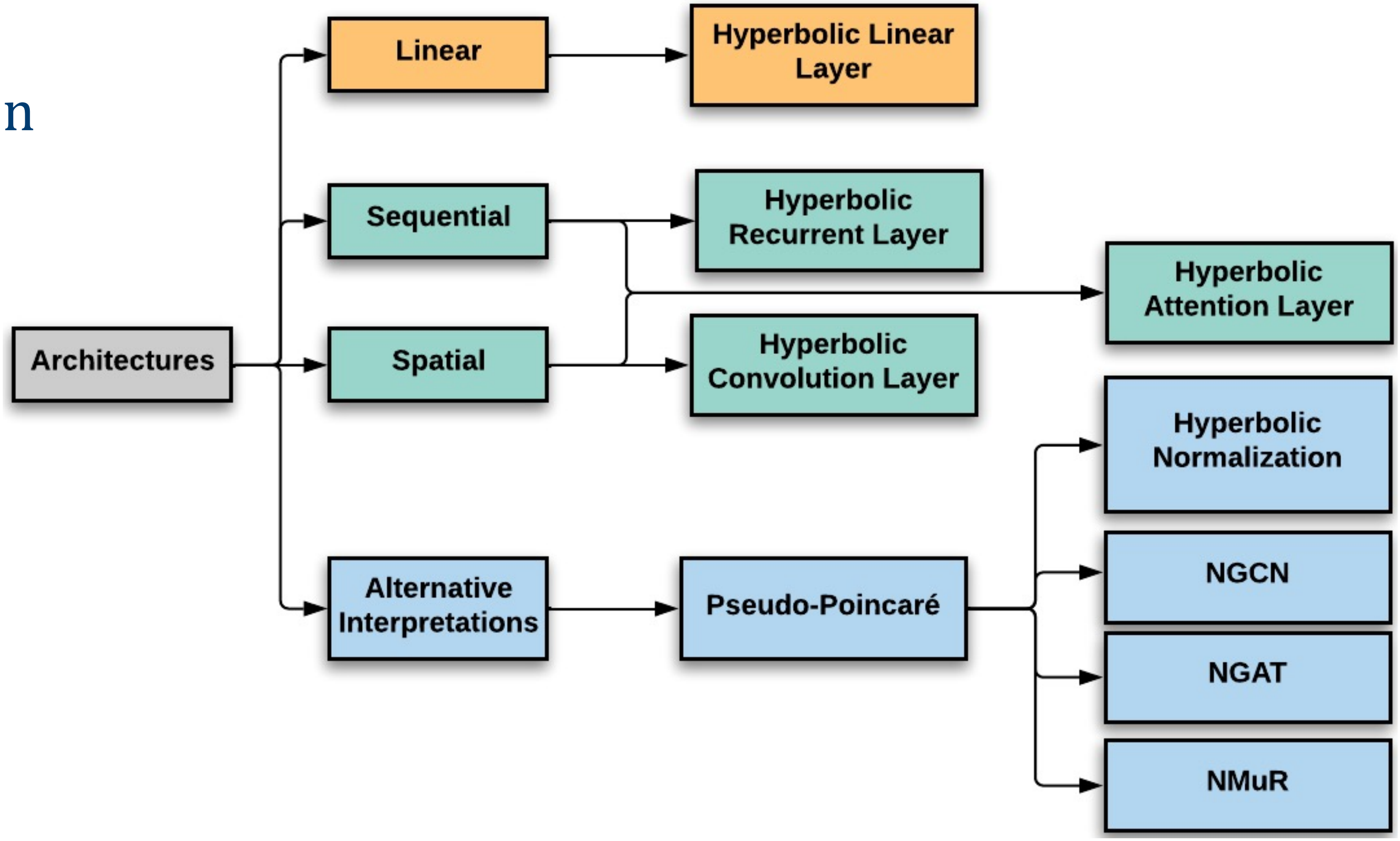
- Manifolds
- Gyrovector Operations
- Neural Networks



Tutorial
Details: Architectures

Architectures/ Implementation

- ➢ Linear layer
- ➢ Sequential data
- ➢ Spatial data
- ➢ Other formulations

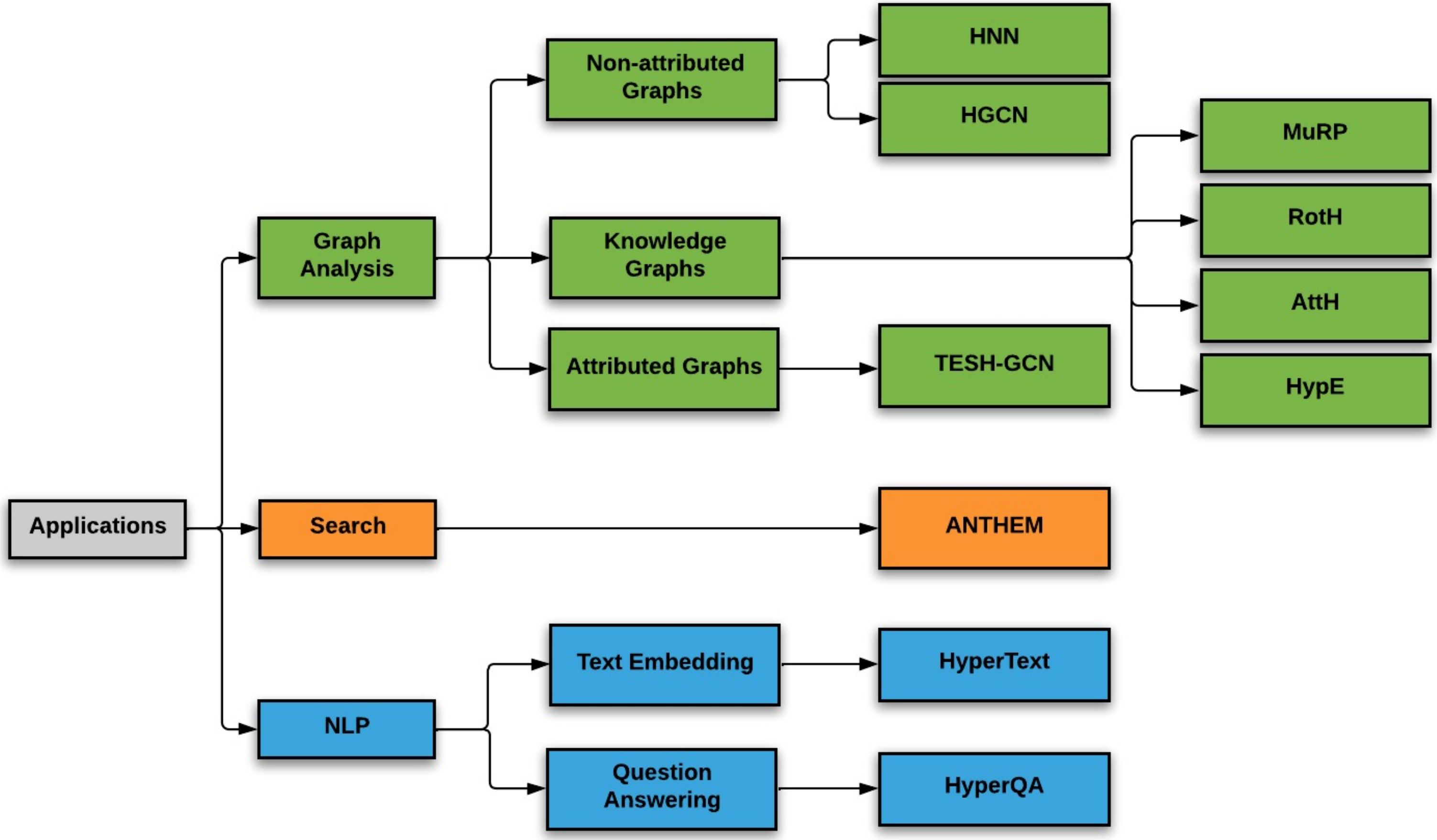


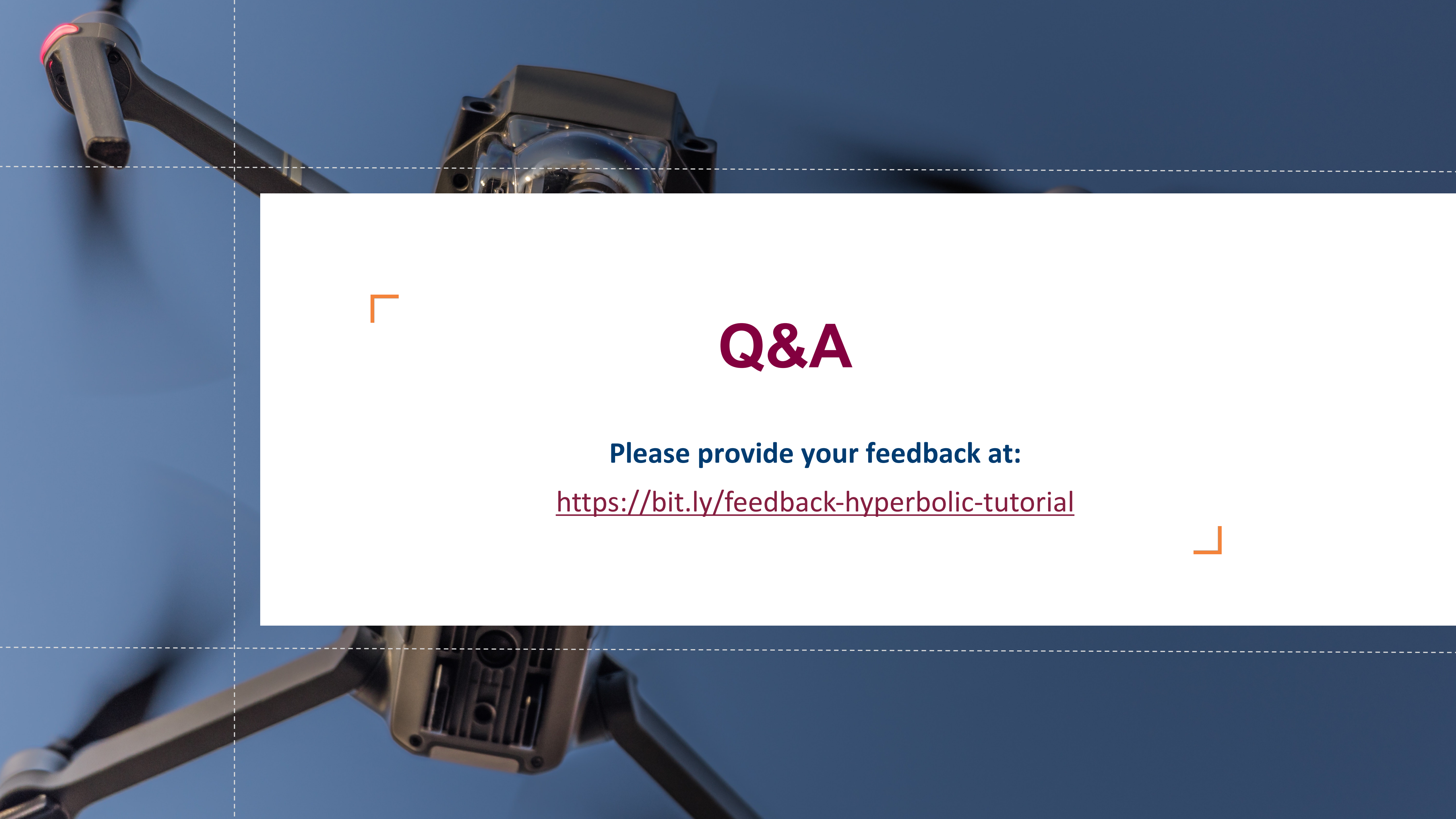
Tutorial

Details: Applications

Real-World Applications

- Graph Analysis
- Knowledge Graph Mining
- Search
- Natural Language Processing



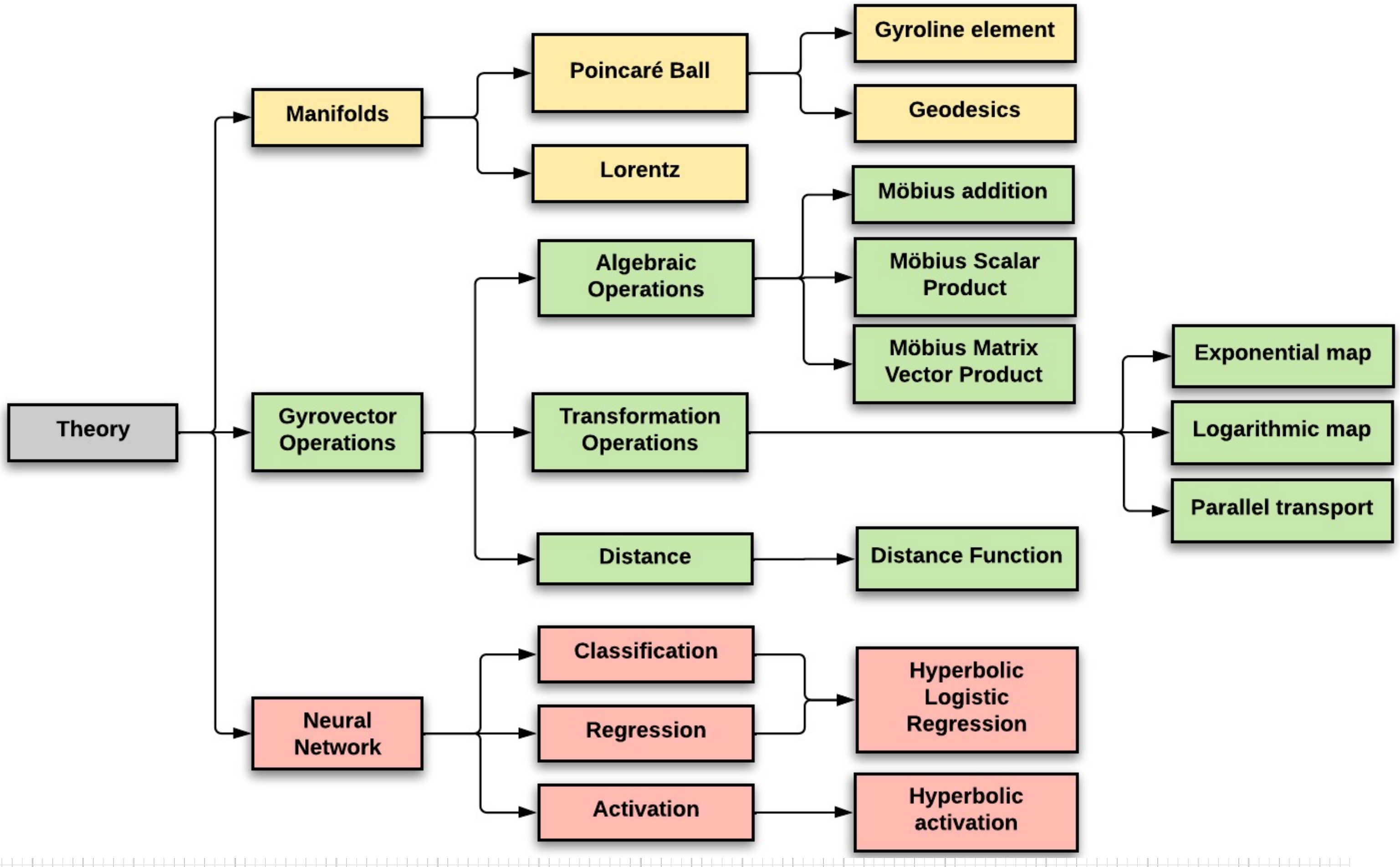


Q&A

Please provide your feedback at:

https://bit.ly/feedback-hyperbolic-tutorial

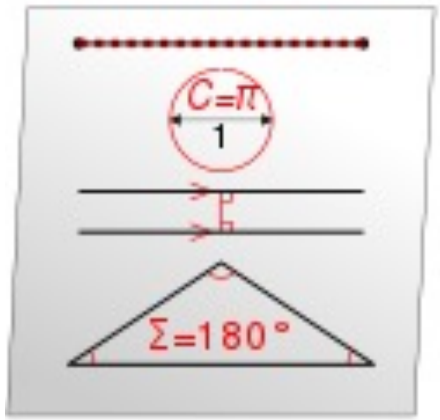
Part 2: Theory



What is Hyperbolic Space?

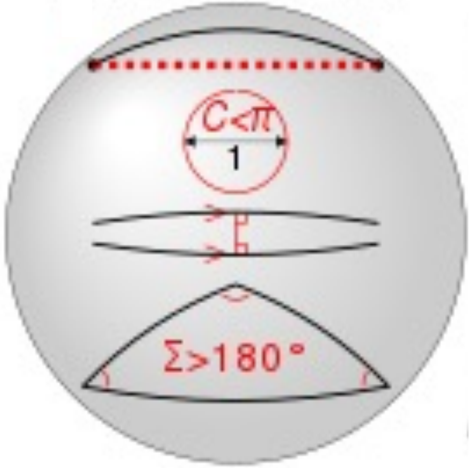
Euclidean vs Elliptic vs Hyperbolic

Euclidean*
(Curvature = 0)



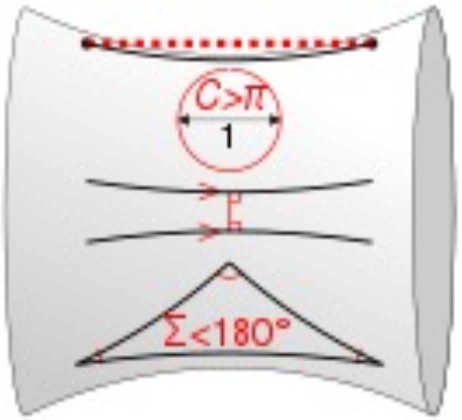
- Circumference/Diameter $= \pi$
- One parallel line
- Sum of angles in a $\Delta = 180°$

Elliptic*
(Curvature > 0)



- Circumference/Diameter $< \pi$
- No Parallel lines.
- Sum of angles in a $\Delta > 180°$

Hyperbolic*
(Curvature < 0)

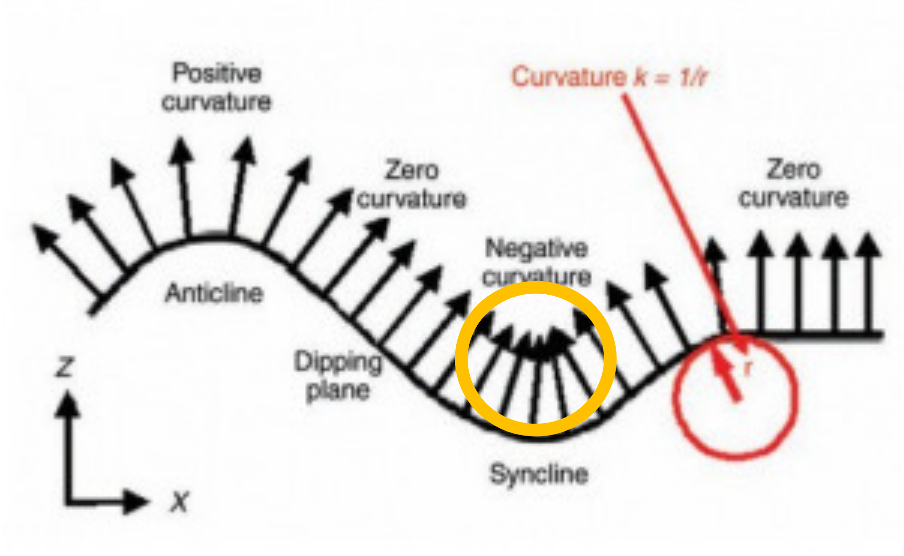


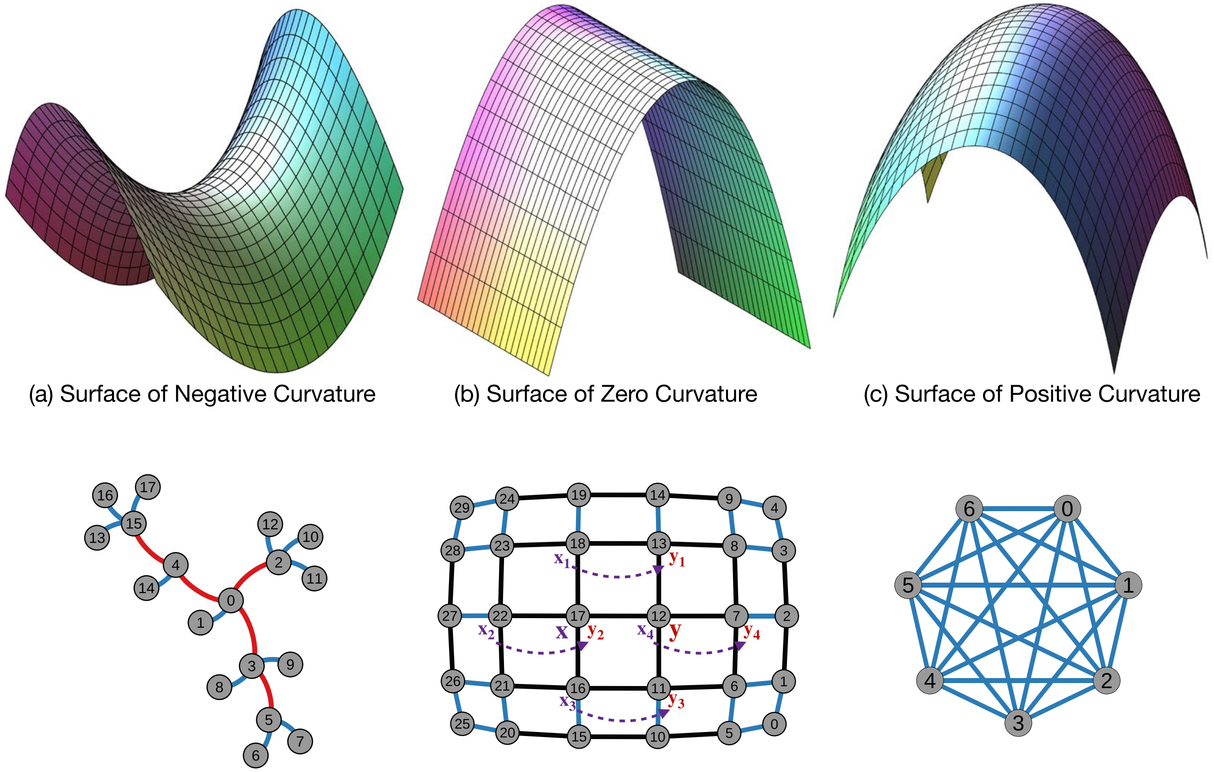
- Circumference/Diameter $> \pi$
- Infinitely many parallel lines
- Sum of angles in a $\Delta < 180°$

An n-dimensional hyperbolic space is an n-dimensional complete Riemannian manifold with a constant negative sectional curvature.

* https://thereaderwiki.com/en/Hyperbolic_geometry

Curvature





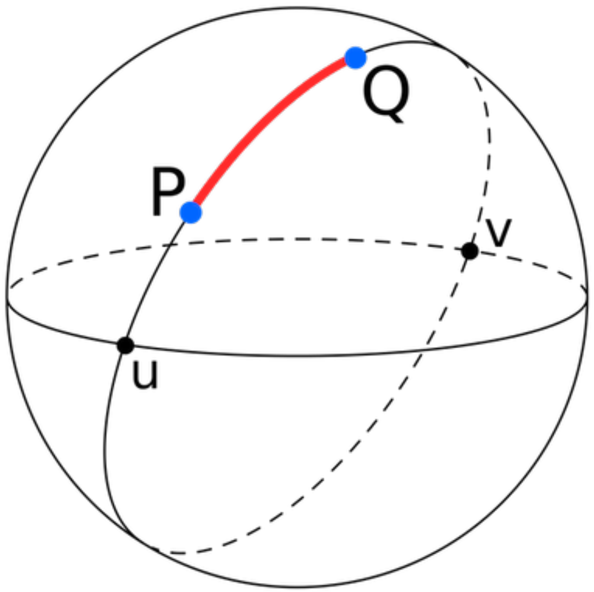
(a) Surface of Negative Curvature (b) Surface of Zero Curvature (c) Surface of Positive Curvature

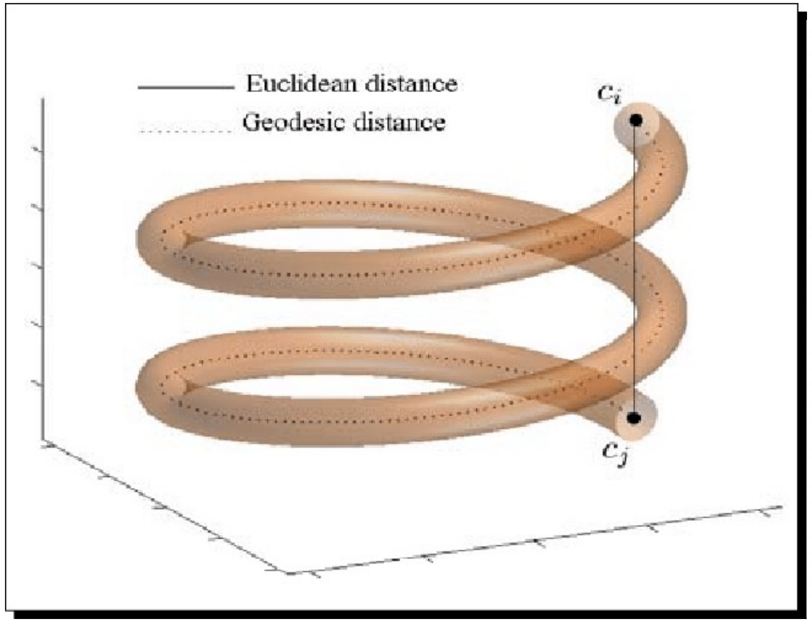
(d) Negative Curvature (e) Zero Curvature (f) Positive Curvature

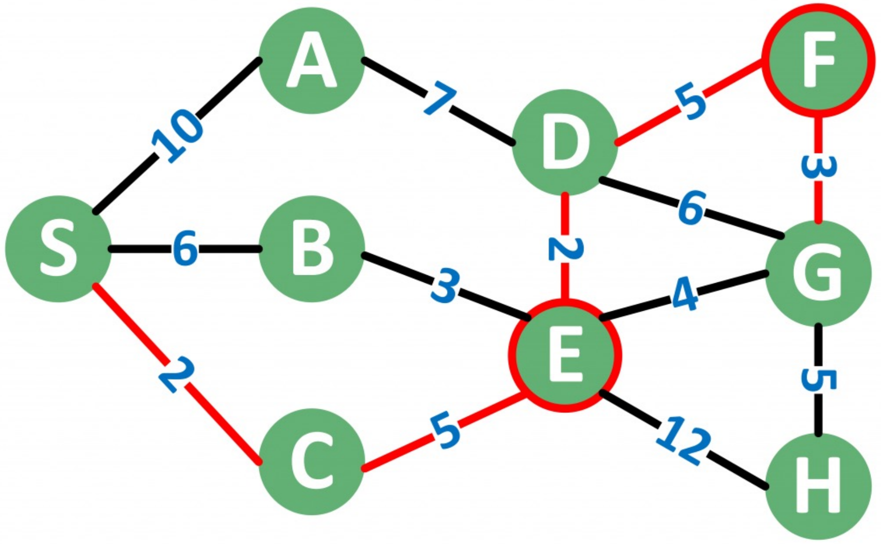
From *Community Detection on Networks with Ricci Flow*, 2019

Given a dataset, estimate curvature.
Learn suitable curvature for a given task and dataset.
Measures: Gromov Hyperbolicity, Ricci Curvature

Geodesics







Wikipedia
https://www.researchgate.net/figure/Euclidean-vs-geodesic-distance-on-a-nonlinear-manifold_fig3_2894469
https://www.baeldung.com/cs/shortest-path-to-nodes-graph

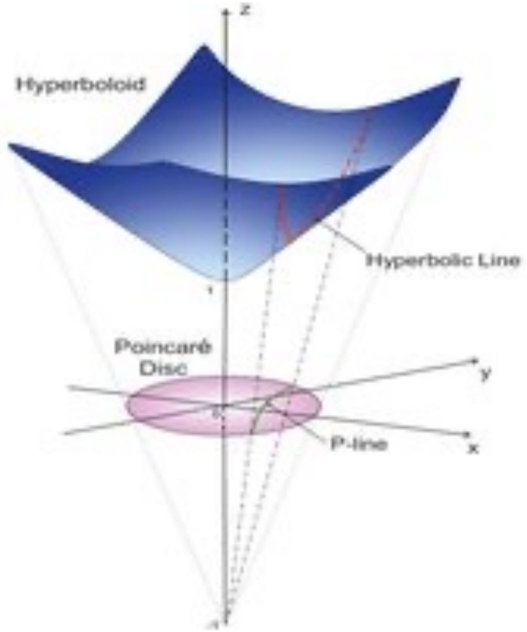
Geodesic: Some kind of shortest distance between two points on the manifold. Applies to graphs too.

Geodesics are generalization of straight lines in Euclidean geometry.
eodesics are constant velocity curves which are locally distance minimizin

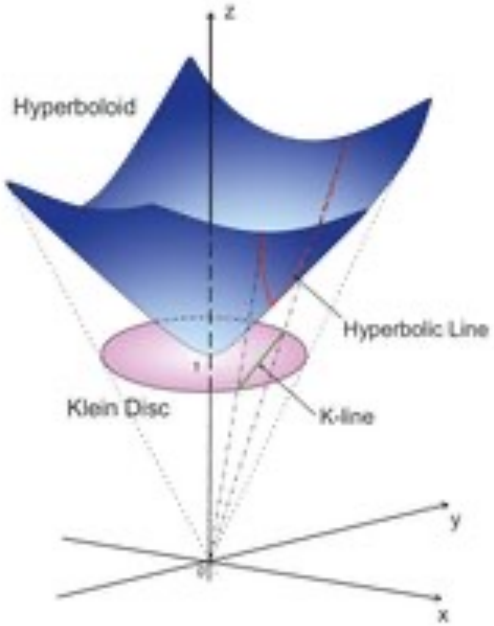
What is Hyperbolic Space?

Isometric models

Defining operations on the large hyperbolic space is difficult. Different projections of the Hyperbolic space are used for different purposes. The projections are isometric to each other, implying, both satisfy the negative curvature property of hyperbolic space.



Poincaré Disk*
Projection on circular disk at origin
Point of projection: (0,0,-1)



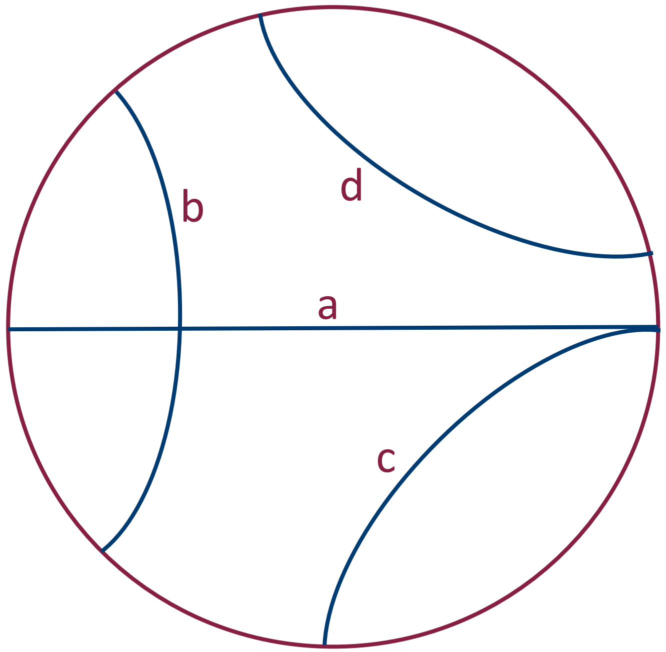
Beltrami-Klein Model*
Projection on circular disk at (0,0,1)
Point of projection: Origin

Poincaré Disk and Klein Disk are isometric representations of hyperboloid sheet.

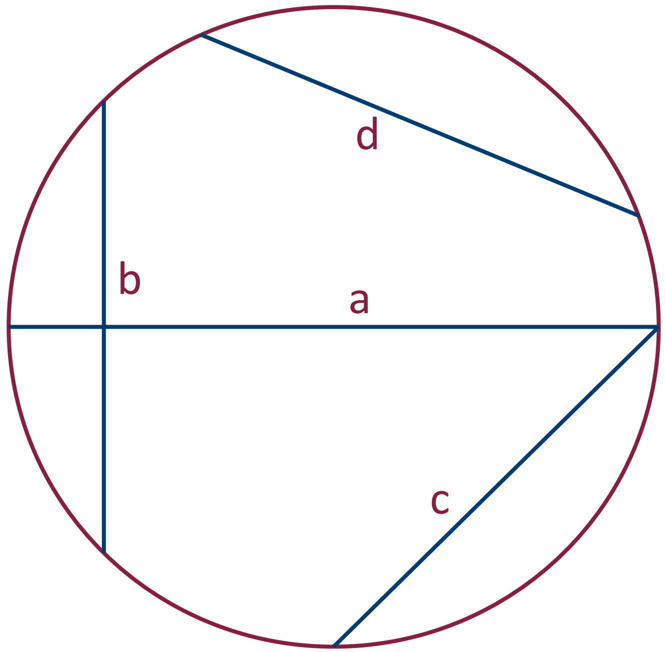
* W. Peng, T. Varanka, A. Mostafa, H. Shi and G. Zhao, "Hyperbolic Deep Neural Networks: A Survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

What is Hyperbolic Space?

Isometric Models



Poincaré Disk
Preserves Angles
Purpose: Good for capturing hierarchy



Beltrami-Klein Model
Preserves Convexity
Purpose: Good for aggregation of values (Attention models)

Use a representation appropriate for the task: visualization, aggregation, etc.

Development of Neural Networks

Requirements

The basic numerical representation in DNNs is vectors/tensors. The underlying operations on these objects are:

- ❑ Distances between points and angles between vectors
- ❑ Addition (by extension subtraction) of vectors
- ❑ Scalar Multiplication of vectors
- ❑ Matrix Multiplication and Concatenation
- ❑ Applying Function (for activations and transformation)
- ❑ SoftMax - Multinomial Logistic Regression (MLR)

To adopt DNNs to a new space, we need to define well-known operations in the new space.

Poincaré Ball Manifold

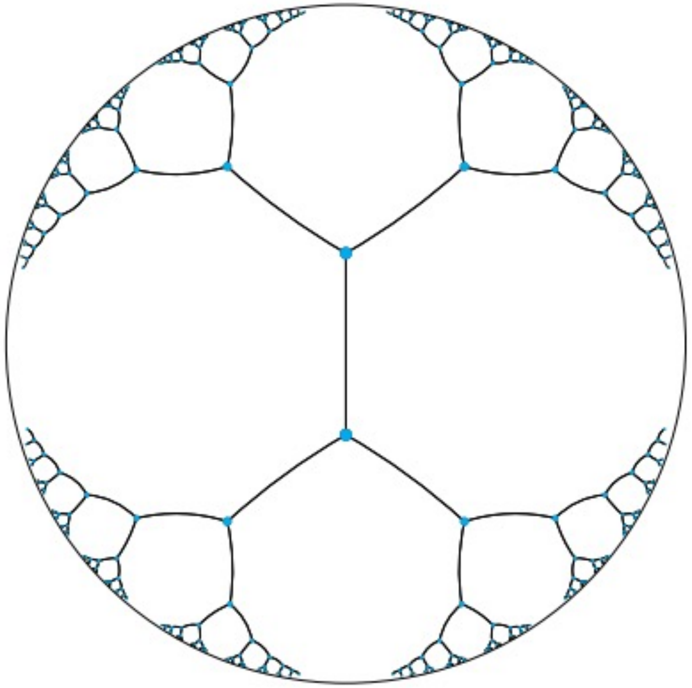
Riemannian Metric

1. Volume of the ball increases exponentially with respect to the radius: Circumference: $2\pi \sinh(r) = O(e^x)$
2. In a hierarchy, the number of nodes increases exponentially with depth.

Riemannian metric helps with the definition of hyperbolic operations; $g^H = \lambda_x^2 g^E, where\ \lambda_x = \frac{2}{1-\|x\|^2},\ and\ g^E = I^n$

Suitable for capturing hierarchies and visualization.

Hyperbolic spaces have exponential capacity.
With *depth ∝ radius*, we can model hierarchy.



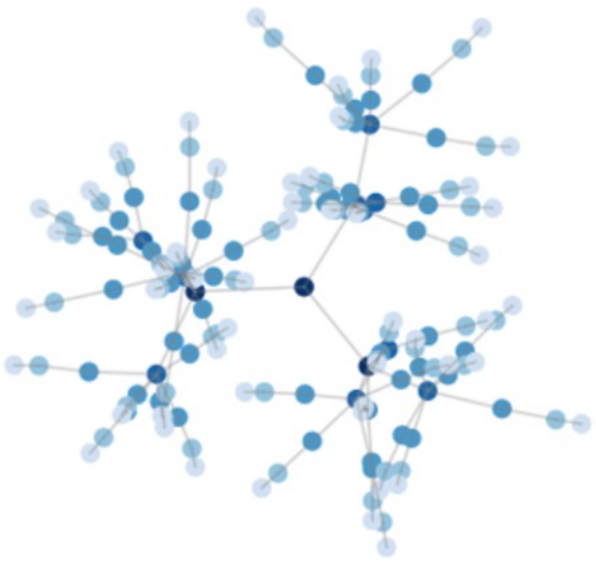
Tree embedded in a Poincaré Disk*
All line segments have the same length

*https://bjlkeng.github.io/posts/hyperbolic-geometry-and-poincare-embeddings/

Poincaré Ball Manifold

Euclidean vs Poincaré Ball Embeddings

DISEASE dataset: Synthetic dataset on disease propagation. The network is trained for Link Prediction. Node Colors represent node depth.



Euclidean Embeddings*
(GCN, DISEASE dataset)

Poincaré Embeddings*
(HGCN, DISEASE dataset)

*Chami, Ines, et al. "Hyperbolic graph convolutional neural networks." Advances in neural information processing systems 32 (2019).

Euclidean Space

Lines, angles, and shortest distance

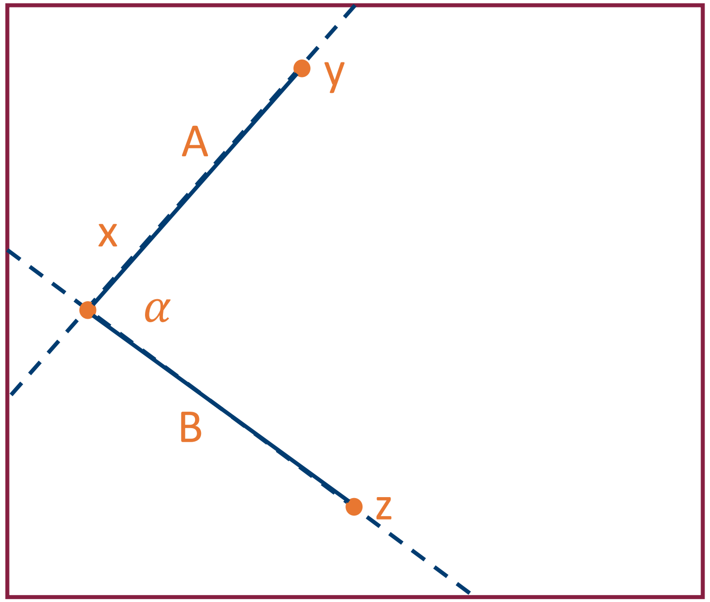
The Euclidean space is defined by the vector space model.

Lines:

$$\gamma_A = x + (y - x)t \ (-\infty < t < \infty)$$
$$\gamma_B = x + (z - x)t \ (-\infty < t < \infty)$$

Angles: Angle between vectors.

$$\cos\alpha = \left\langle \frac{y - x}{\|y - x\|}, \frac{z - x}{\|z - x\|} \right\rangle$$



Vectors:

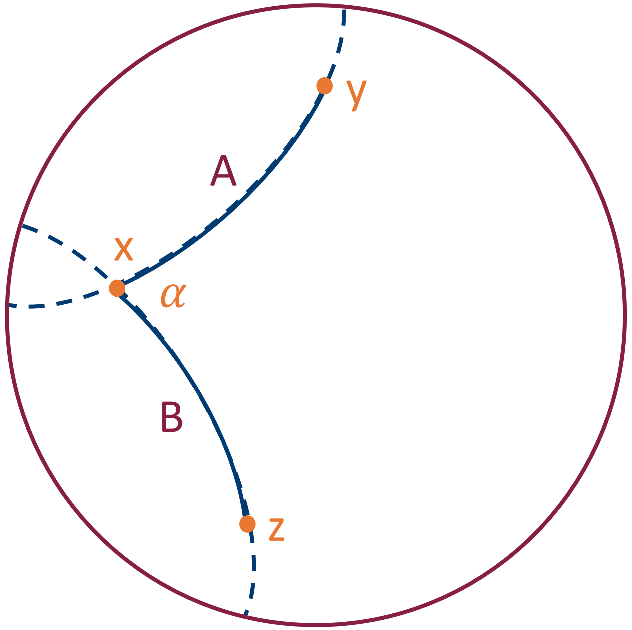
$$\gamma_{x\to y} = y - x$$
$$\gamma_{x\to z} = z - x$$

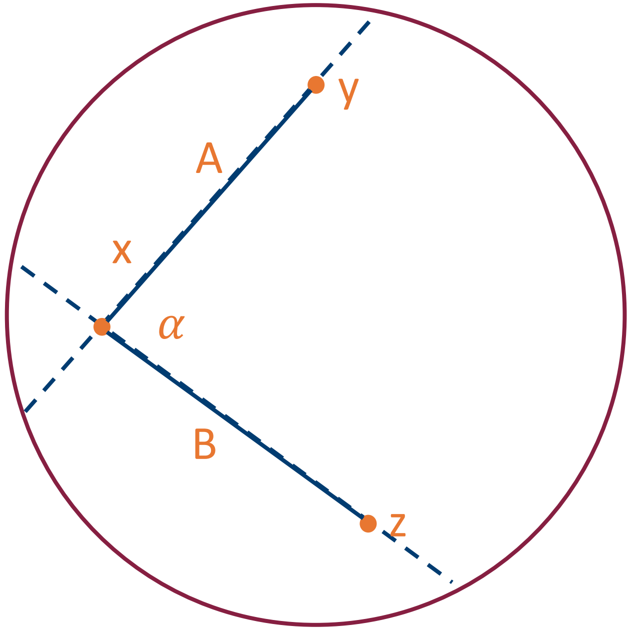
Geodesic length:

$$A(x,y) = \|y - x\|$$
$$B(x,z) = \|z - x\|$$

Hyperbolic Space

Gyrolines in Poincaré Ball and Klein Model





Gyrolines: Line in the gyrovector space.

$$\gamma_A = x \oplus (\ominus x \oplus y) \otimes t \ (-\infty < t < \infty)$$

$$\gamma_B = x \oplus (\ominus x \oplus z) \otimes t \ (-\infty < t < \infty)$$

Gyrolines are algebraic counterparts of Geodesics.
What are the closed form expressions?

Hyperbolic Space

Gyroline and Geodesic Distance

The Euclidean geometry are defined in the vector space model,
For hyperbolic geometry on Poincaré ball, we rely on the gyrovector space model.

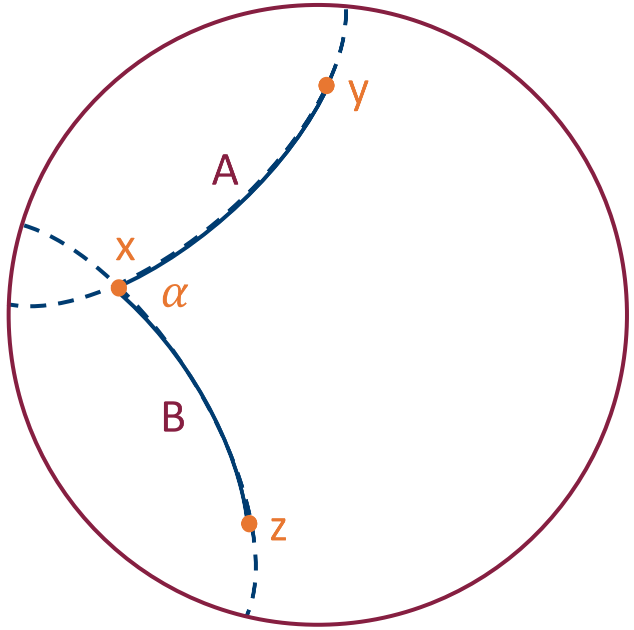
Gyrolines: Line in the gyrovector space.

$$\gamma_A = x \oplus (\ominus x \oplus y) \otimes t \quad (-\infty < t < \infty)$$

$$\gamma_B = x \oplus (\ominus x \oplus z) \otimes t \quad (-\infty < t < \infty)$$

Angles: Angle between segments.

$$cos\alpha = \left\langle \frac{\ominus x \oplus y}{\| \ominus x \oplus y \|}, \frac{\ominus x \oplus z}{\| \ominus x \oplus z \|} \right\rangle$$



Segment (solid): Vector in gyrovector space.

$$\gamma_{x \to y} = \ominus x \oplus y$$

$$\gamma_{x \to z} = \ominus x \oplus z$$

Geodesic length: Length of vector/Shortest distance between two points in gyrovector space.

$$\mathrm{A(x,y)} = \| \ominus x \oplus y \|$$

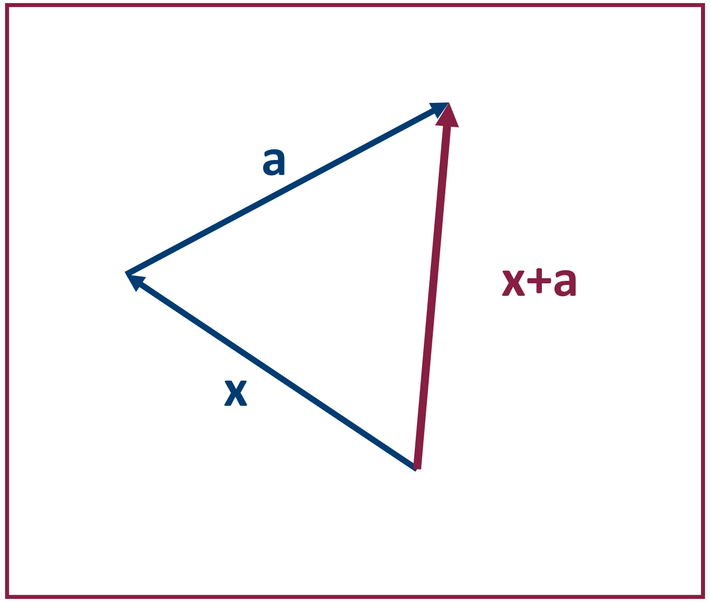
$$\mathrm{B(x,z)} = \| \ominus x \oplus z \|$$

Gyrovector space is similar in notation to vector space.
Not Commutative or associative in the usual sense.

Poincaré Ball Manifold
Euclidean Addition

Euclidean Addition *x+a*, can be defined as the translation of **vector *x*** by **vector *a***.

$$x + a = (x_1+a_1, x_2+a_2)$$



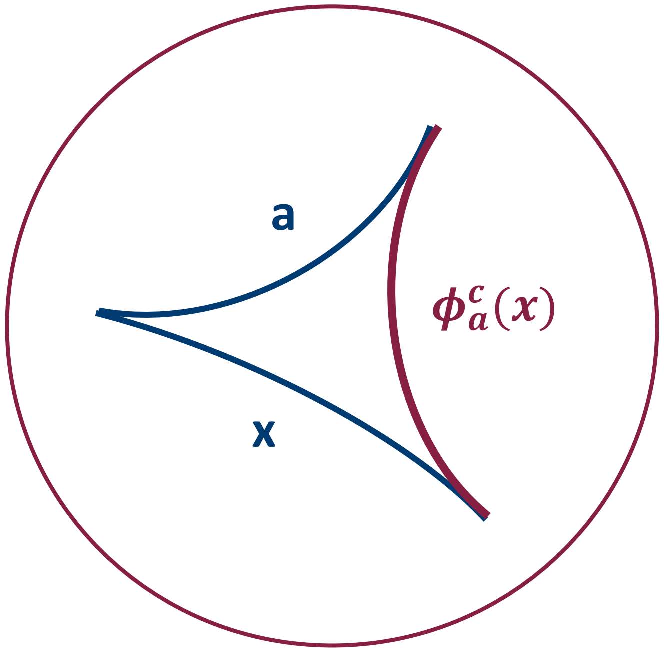
Note: A translation vector is a type of transformation that moves a figure in the coordinate plane from one location to another.

Poincaré Ball Manifold

Gyro addition through Möbius Transformations

Such translation in Poincaré Ball is possible through Möbius Transformation;

$$f(a,x) = \phi_a^c(x)$$



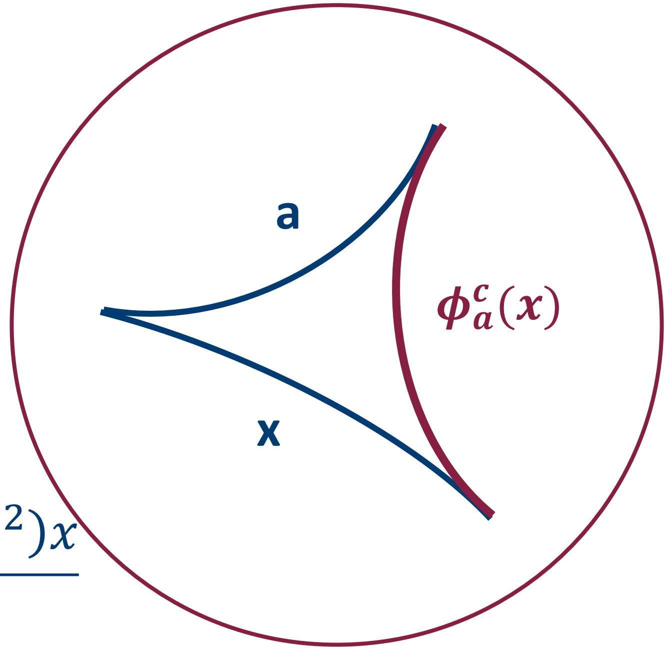
Poincaré Ball Manifold

Gyro addition through Möbius Transformations

Such translation in hyperbolic space is possible through Möbius Transformation which for the Poincaré ball is;

$$f(a,x) = \phi_a^c(x)$$

$$\phi_a^c(x) = \frac{x + ca}{c\tilde{a}x + 1} = \frac{(1 + 2ca \cdot x + c \parallel x \parallel^2)a + (1 - c \parallel a \parallel^2)x}{1 + 2ca \cdot x + c^2 \parallel a \parallel^2 \parallel x \parallel^2}$$



Poincaré Ball Manifold

Gyro addition through Möbius Transformations

Möbius Addition:

$$x \oplus_c a = \frac{(1 + 2ca \cdot x + c \parallel x \parallel^2)a + (1 - c \parallel a \parallel^2)x}{1 + 2ca \cdot x + c^2 \parallel a \parallel^2 \parallel x \parallel^2}$$

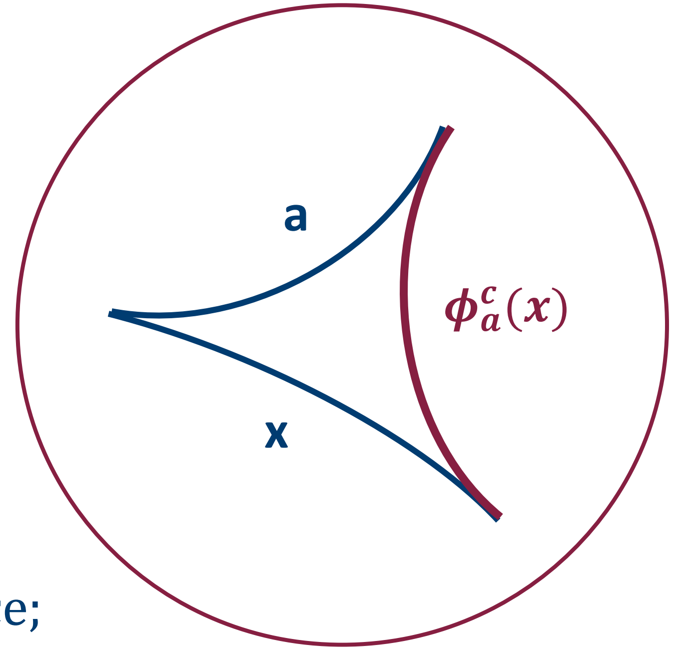
Similarly, Möbius subtraction is handled as;

$$x \ominus_c a = x \oplus_c (-a)$$

Note that zero-curvature gives back the Euclidean space;

$$a \oplus_0 x = x + a$$

$$x \ominus_0 a = x - a$$



Poincaré Ball Manifold

Möbius scalar multiplication

Möbius scalar product:

$$r \otimes_c x = x \oplus_c x \oplus_c \dots \oplus_c x \text{ (r times)}$$
$$= \frac{1}{\sqrt{c}} \tanh(r \tanh^{-1}(\sqrt{c} \parallel x \parallel)) \frac{x}{\|x\|}$$

Note that zero-curvature gives back the Euclidean space;

$$\lim_{c \to 0} r \otimes_c x = rx$$

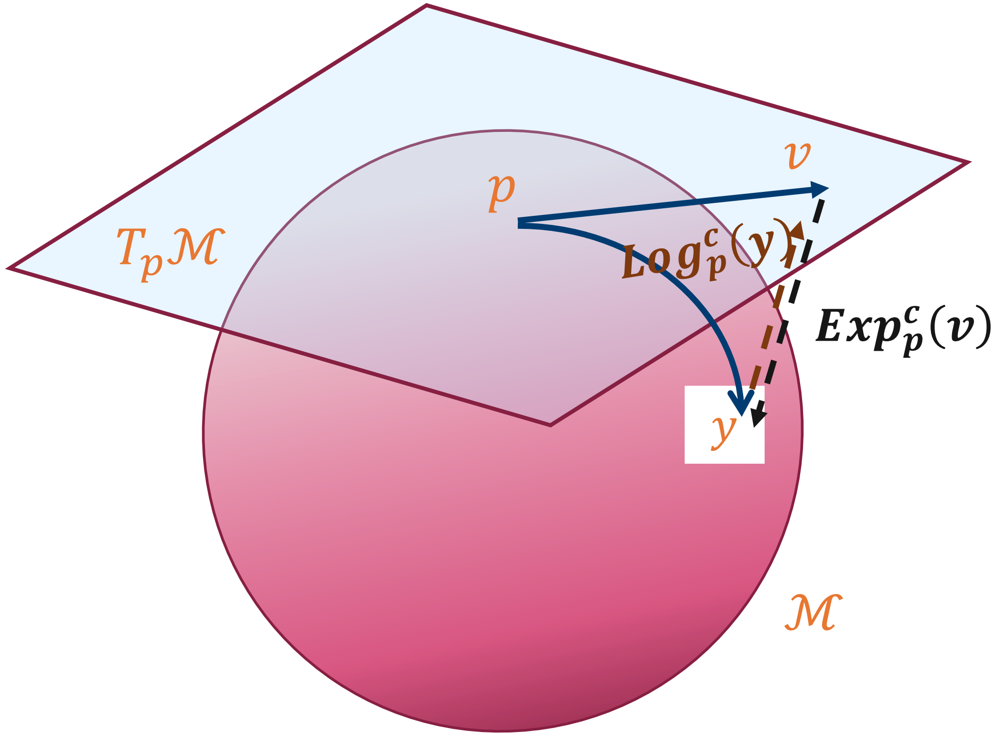
Möbius operations provide the basis for vector addition and scalar multiplication on Poincaré Ball.
How do we perform such operations on a general Riemannian manifold?
A general Riemannian manifold does not have a coordinate representation.

Poincaré Ball Manifold

Transformations: Local Tangent Space

For defining operations in hyperbolic space, we utilize the tangent plane which consists of velocity vectors and is Euclidean.

To move between a Poincaré ball manifold $\mathcal{M}$ and local tangent plane $T_p\mathcal{M}$ at point p, we use the logarithmic map $\mathrm{Log}_p(y) : \mathcal{M} \to T_p\mathcal{M}$ and exponential map $\mathrm{Exp}_p(v) : T_p\mathcal{M} \to \mathcal{M}$



$$Log_p^c(y) = \frac{2}{\sqrt{c}\lambda_p^c} \tanh^{-1}(\sqrt{c}\| - p \oplus_c y\|) \frac{-p \oplus_c y}{\| - p \oplus_c y\|}$$

$$Exp_p^c(v) = p \oplus_c \left(\tanh\left(\sqrt{c} \frac{\lambda_x^c \parallel v \parallel}{2} \right) \frac{v}{\sqrt{c} \parallel v \parallel} \right)$$

In $\mathbb{R}^n$, $Exp_p(v) = p + v$

Tangent space is Euclidean and admits coordinate representation. Tangent space is different from the manifold.

Poincaré Ball Manifold

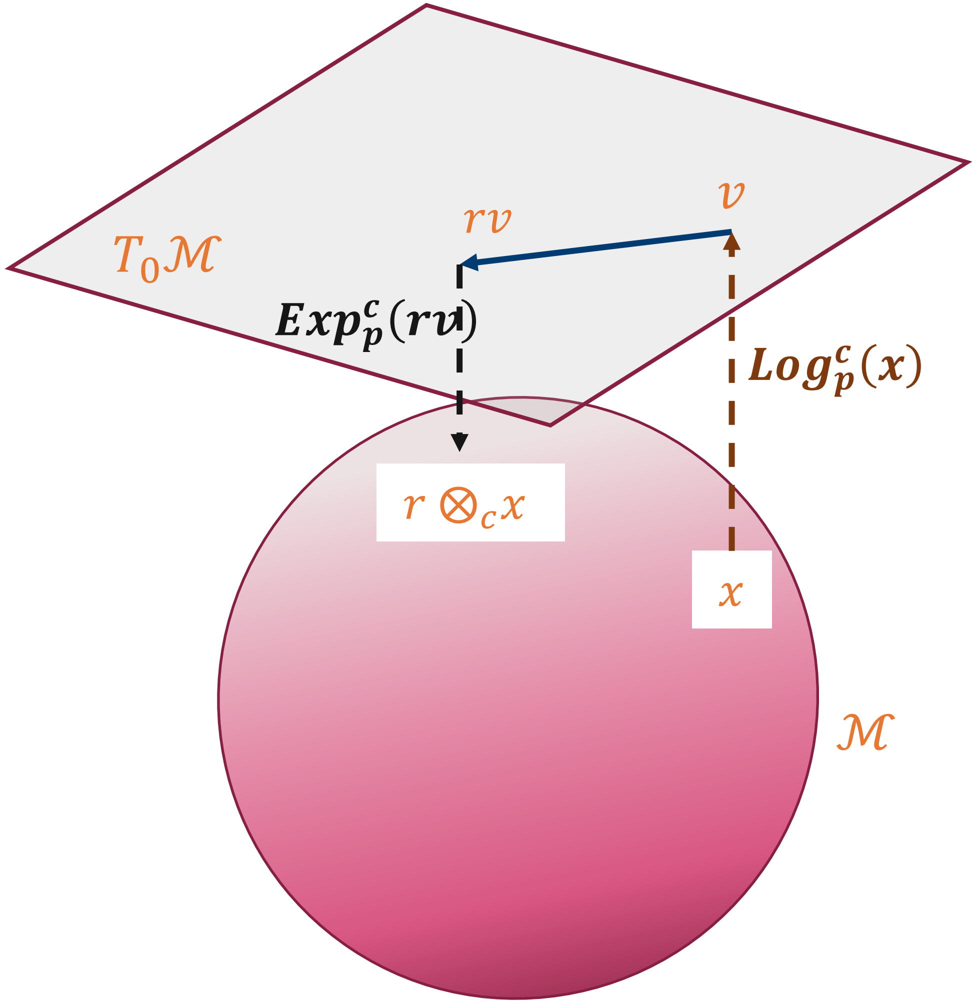
Transformations: Local Tangent Space

Note that, tangent plane behaves like a Euclidean, so a Reformulation of Möbius scalar multiplication is valid:

$$r \otimes_c x = Exp_0^c\left(rLog_0^c(x)\right)$$

Möbius matrix-vector product;

$$\begin{bmatrix} M_1 & M_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = M_1 \otimes_c x_1 \oplus_c M_2 \otimes_c x_2$$



Poincaré Ball Manifold

Neural Networks: Activation Function

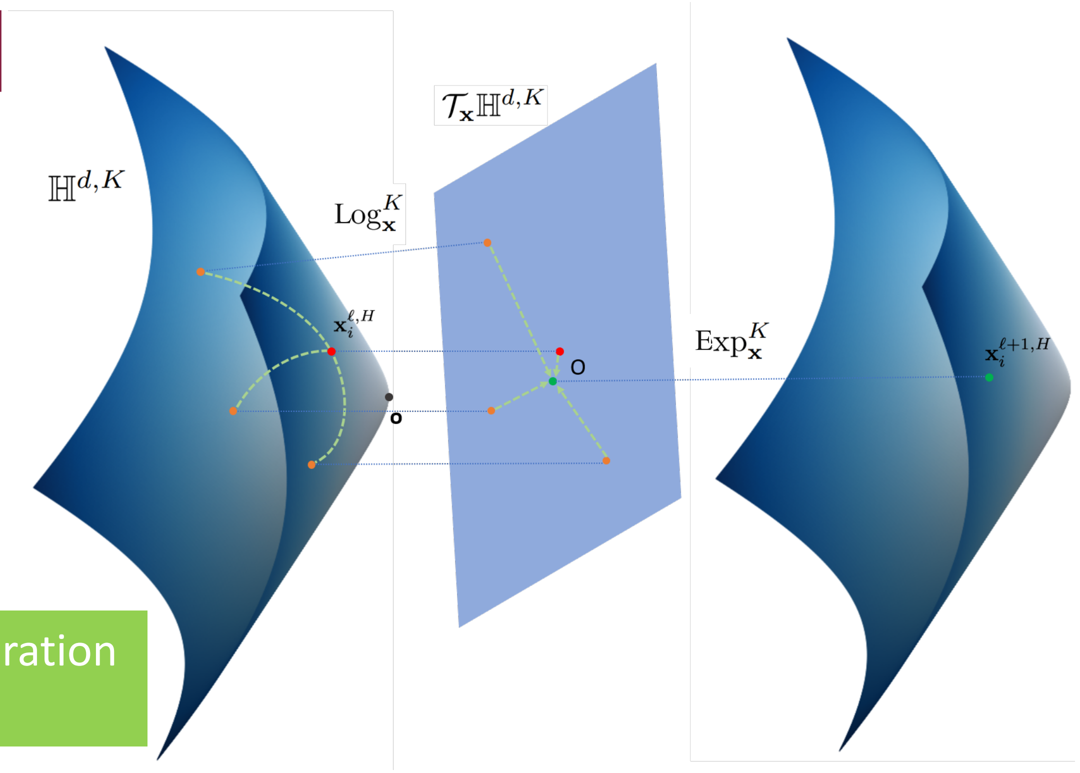
Using a Euclidean non-linear activation $f(x): \mathbb{H}^d \rightarrow \mathbb{H}^d$ in hyperbolic space;

Map to tangent space, so Euclidean functions are valid

Apply Euclidean function to the mapped input

Map the output back to the Poincaré Ball

$$f^{\otimes_c}(x) = Exp_0^c(f(Log_0^c(x)))$$



General Recipe: Map to tangent space, perform the operation like aggregation, etc. and map back to manifold.

Image Source: Chami, Ines, et al. "Hyperbolic graph convolutional neural networks." Advances in neural information processing systems 32 (2019).

Poincaré Ball Manifold

Neural Networks: Distance Function

We can use the mapping function to compute hyperbolic distances too;

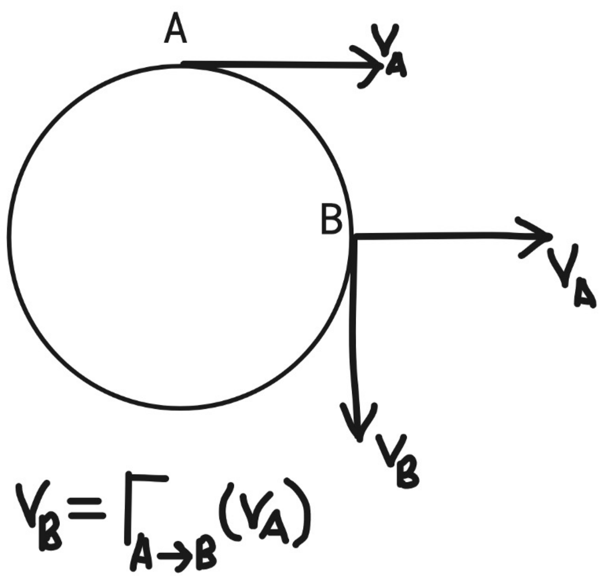
$$d_{\mathbb{R}}(x, y) = \| y - x \|$$

$$d_{\mathbb{H}}(x, y) = \; Exp_0^c(\| -x \oplus_c y \|)$$

$$d_{\mathbb{H}}(x, y) = \frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \, \| -x \oplus_c y \|)$$

Poincaré Ball Manifold

Neural Networks: Parallel Transport



https://miro.medium.com/max/1400/1*iB5lwZrALNHssRYtRdO2wA.png

How do we move around hyperbolic space?

Poincaré Ball Manifold

Neural Networks: Parallel Transport

To move a point from one tangent plane to another, we use Parallel Transport.

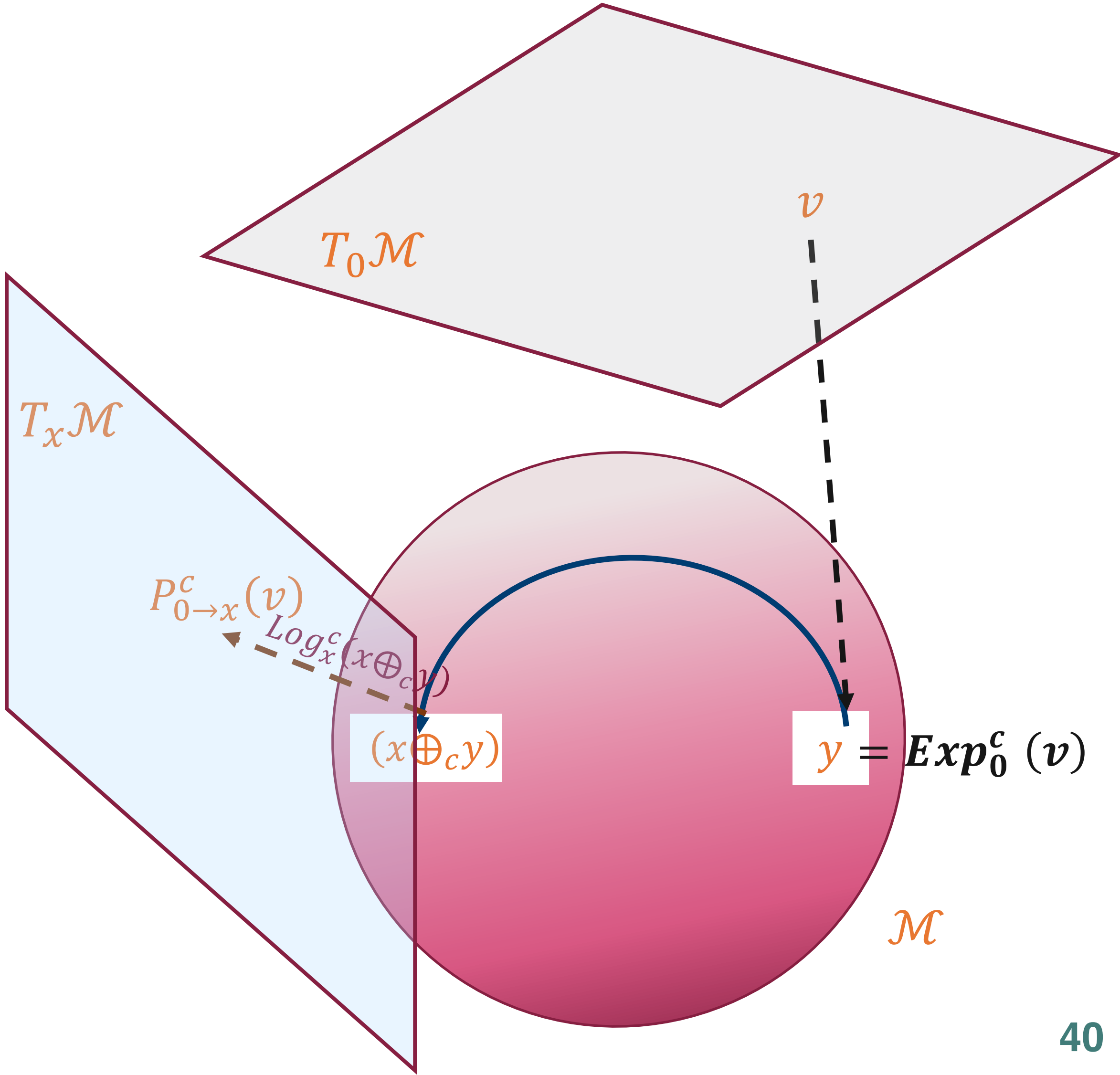
Parallel Transport is the movement of a tangent vector at y to a tangent vector at x along the geodesic between y and x.

$$P^c_{0\to x}(v) = Log^c_x\left(x \oplus_c Exp^c_0(v)\right) = \frac{\lambda^c_0}{\lambda^c_x} v$$

In Euclidean space, parallel transport is identity transformation because the space is flat.

The common application of PT is for *hyperbolic* bias addition.

$$x \oplus b = Exp_x\left(P_{0\to x}\left(Log_0\ (b)\right)\right)$$



Poincaré Ball Manifold

Multiclass Logistic Regression

Euclidean MLR: $P(y=k|x) \propto exp(\langle a_k, x\rangle - b_k)$

Let us say for a class k, the Euclidean separator hyperplane is defined as $\mathbb{E}_k\colon \langle a_k, x\rangle - b_k = 0$

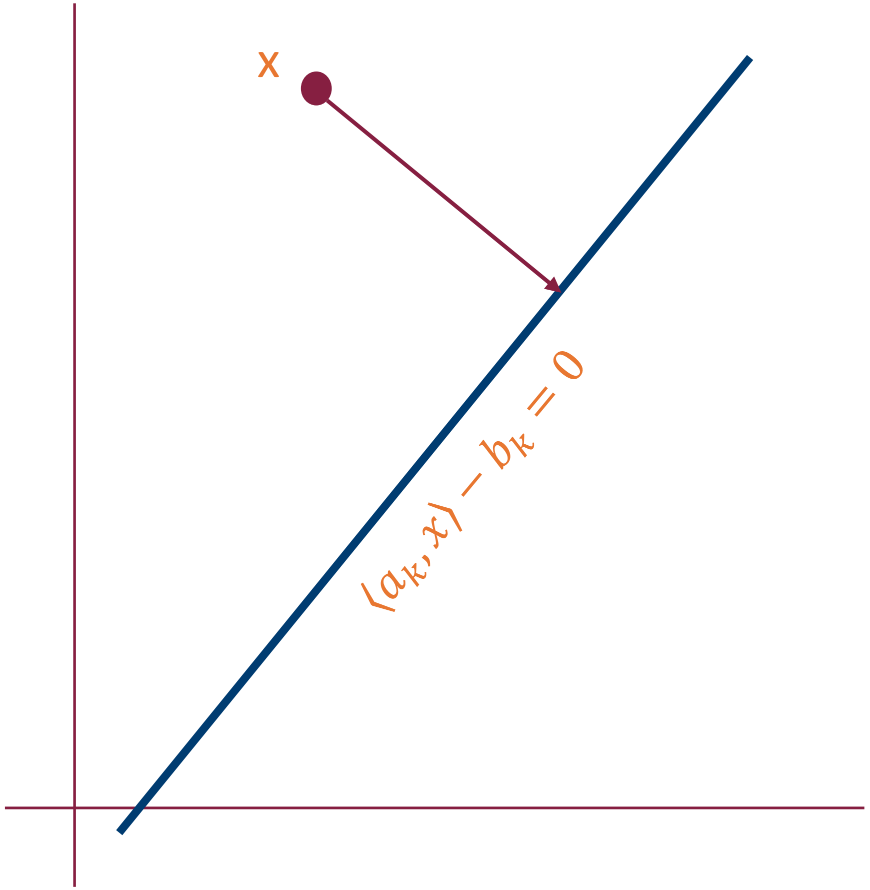
Consequently,

$$P(y=k|x) \propto \exp\Big(sign(\langle a_k, x\rangle - b_k)\|a_k\| d_{\mathbb{R}}(x, \mathbb{E}_k)\Big)$$

Half space(+/-)

Perpendicular Magnitude

Distance from Hyperplane



Poincaré Ball Manifold
Multiclass Logistic Regression

Euclidean MLR:

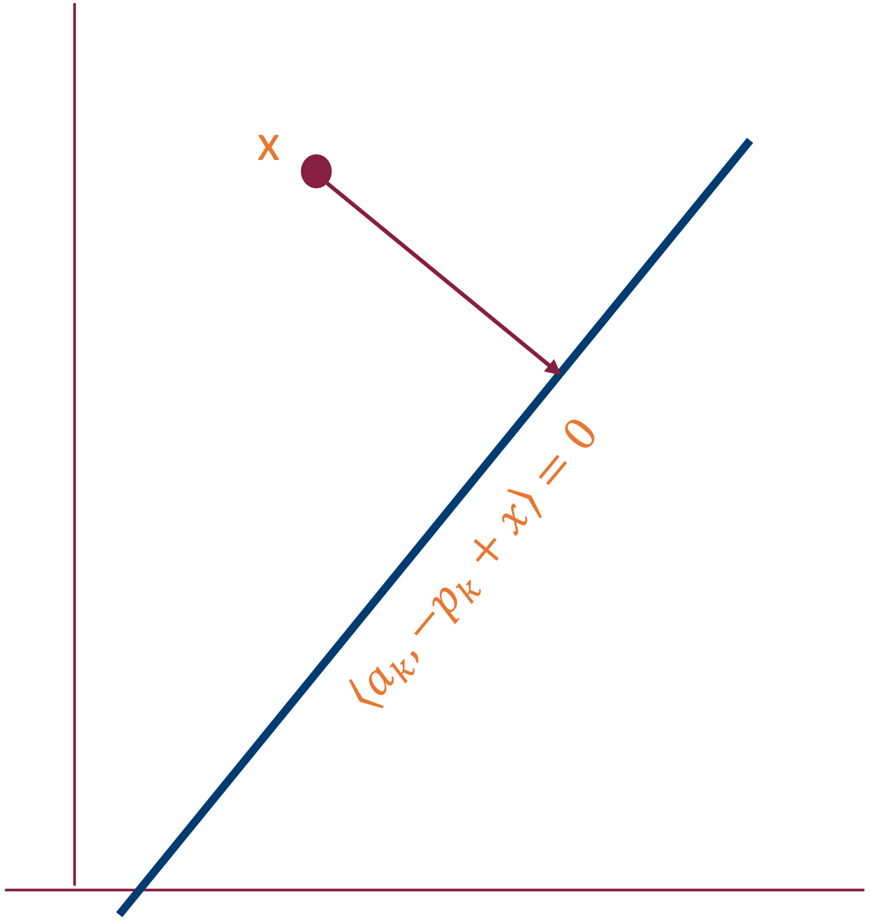
Replacing $b_k = a_k p_k$; $\mathbb{E}_k: \langle a_k, -p_k + x \rangle = 0$
Consequently,

$$P(y = k|x) \propto exp\big(sign(\langle a_k, -p_k + x\rangle) \|a_k\| d_{\mathbb{R}}(x, \mathbb{E}_k)\big)$$

Half space (+/-)

Perpendicular Magnitude

Distance from Hyperplane



MLR in Euclidean space is defined by hyperplanes.
Probability depends on distance from the hyperplane and side.

Poincaré Ball Manifold

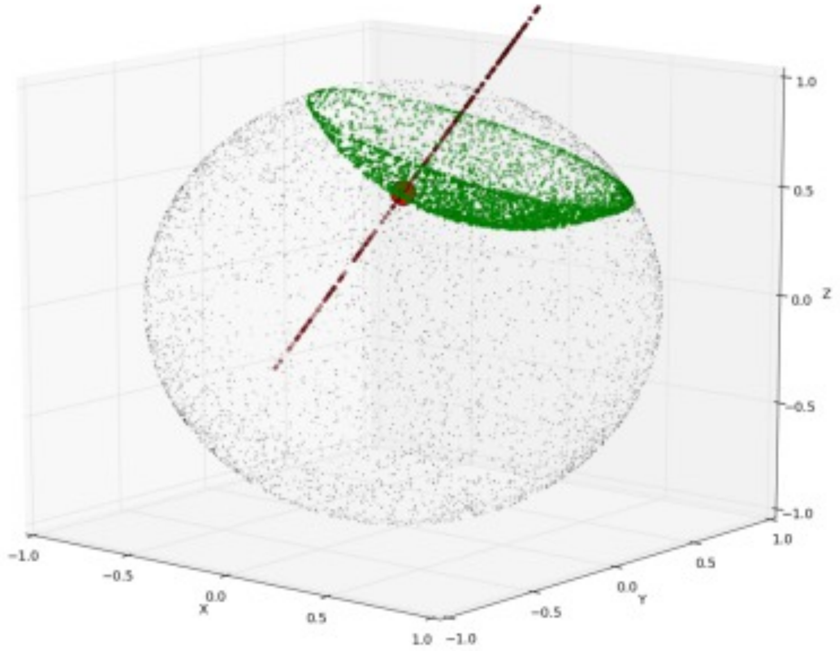
Multiclass Logistic Regression

Hyperbolic MLR:

The hyperbolic equivalent of the hyperplane; $\mathbb{H}_k^c : \langle -p_k \oplus_c x, a_k \rangle = 0$
Consequently,

$$P(y = k|x) \propto exp\left(sign(\langle -p_k \oplus_c x, a_k \rangle)\sqrt{g_{p_k}^c(a_k, a_k)}d_{\mathbb{H}}(x, \mathbb{H}_k^c)\right)$$

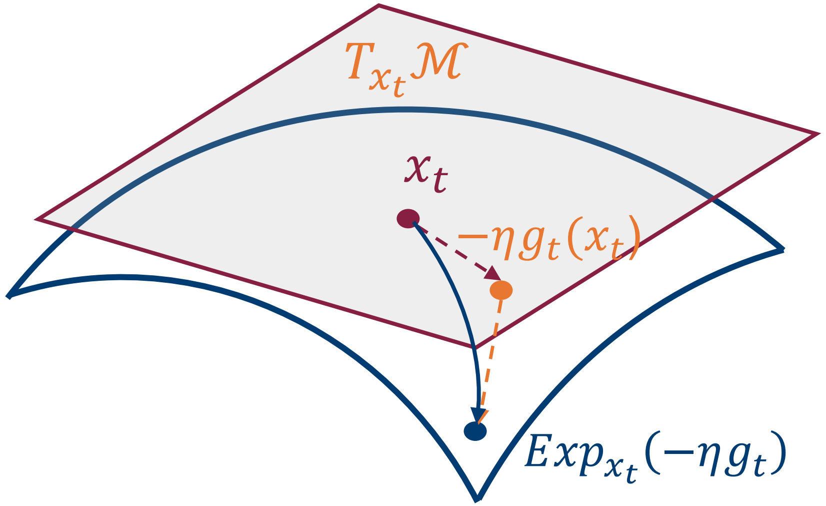
$$P(y = k|x) \propto exp\left(\frac{\lambda_{p_k}^c \|a_k\|}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_k \oplus_c x, a_k \rangle}{(1 - c\|-p_k \oplus_c x\|^2)\|a_k\|}\right)\right)$$



Green point cloud denotes hyperplane in the Hyperbolic Plane.
The red point is p.
The red normal axis to the hyperplane through p is parallel to a.

Image Source: Ganea, Octavian, Gary Bécigneul, and Thomas Hofmann. "Hyperbolic neural networks." Advances in neural information processing systems 31 (2018).

Riemannian Optimization



$$\text{SGD: } x_{t+1} \leftarrow x_t - \alpha g_t$$

RSGD consists of the following three steps at x_t.

1. Evaluate the gradient g_t
2. Project g_t to the tangent space
3. Perform gradient step in the negative direction of the tangent vector

$$\text{RSGD: } x_{t+1} \leftarrow Exp_{x_t}(-\alpha g_t)$$

Approximation: For computational efficiency, we use *retraction map* instead of the *exponential map*.

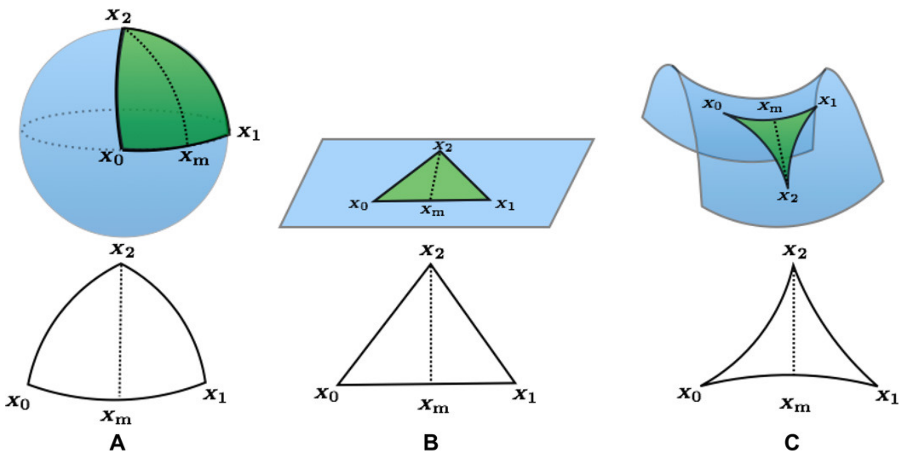
$$x_{t+1} \leftarrow Proj_{\mathcal{M}}(x_t - \alpha g_t)$$

Riemannian AdaGrad: Decompose manifold, $\mathcal{M}$ into cartesian product of n sub-manifolds $\mathcal{M}_k$ and updates are manifold-wise and not coordinate-wise.

$$x_{t+1}^i \leftarrow \mathrm{Exp}_{x_t^i}^i\left(-\alpha g_t^i / \sqrt{\sum_{k=1}^{t} \|g_k^i\|_{x_k^i}^2}\right)$$

Hyperbolicity of Network Data and Clustering

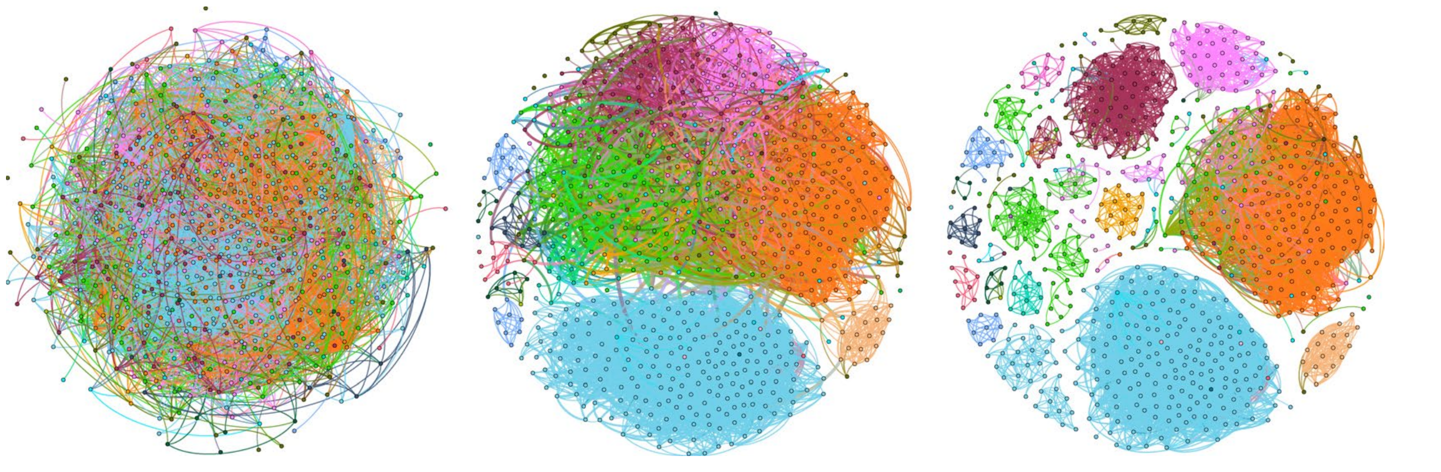
Can we measure hyperbolicity of network data?



From https://web.math.princeton.edu/~ruobingz/research.html

A geodesic metric space (X,d) is called δ-hyperbolic if every geodesic triangle δ-slim, i.e. any point on any of the sides of the triangle is within distance δ of the other two sides. Finally, a metric space is Gromov hyperbolic if it is δ-hyperbolic for some δ ≥ 0.

Ricci tensor is that it describes how much a volume element would differ in curved space compared to Euclidean or flat space.



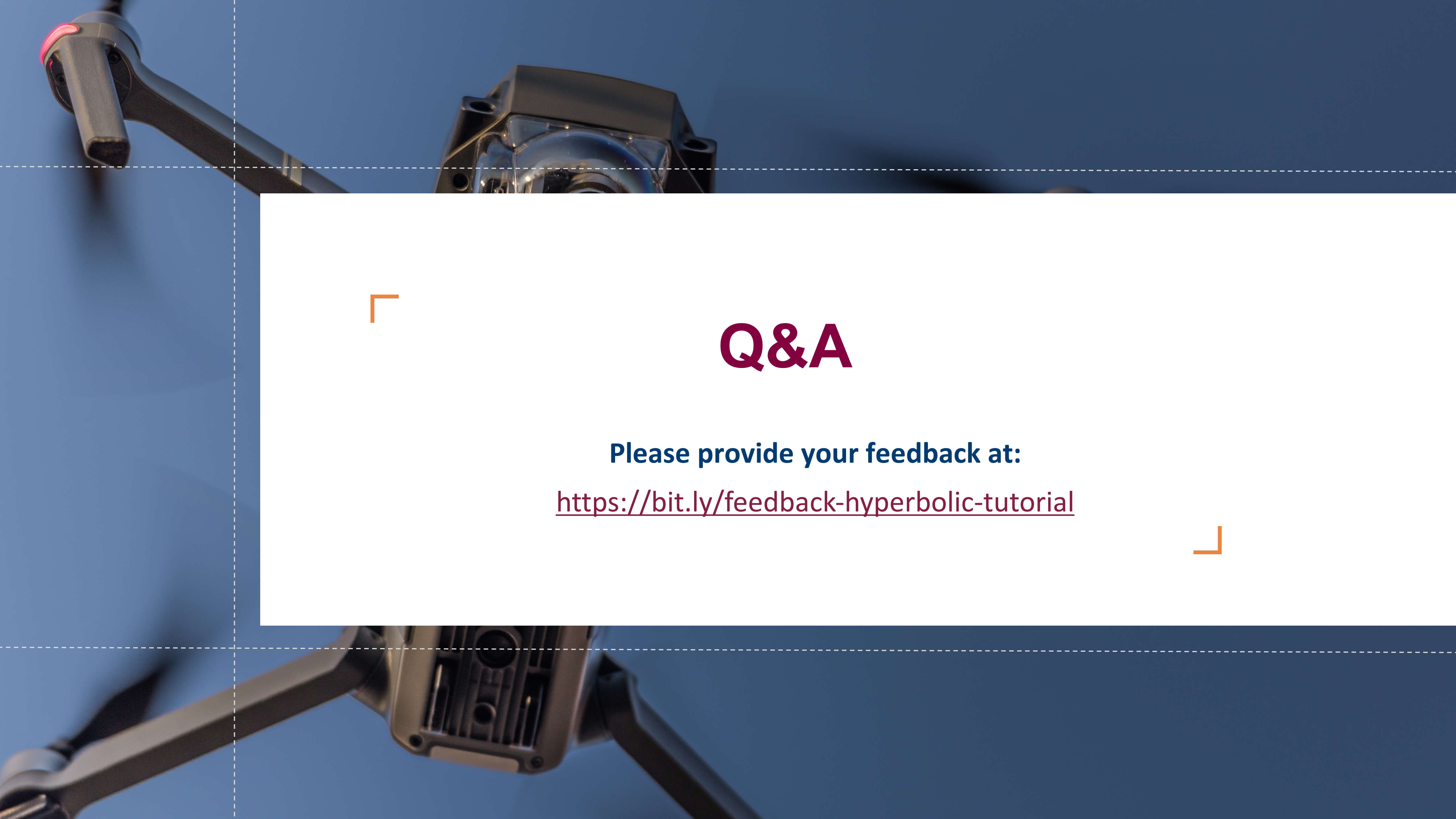
(a) (b) (c)

(a) A Facebook ego network of one user with 792 friends and 14025 edges generated by Gephi's Fruchterman Reingold layout59. The colors represent 24 different friend circles (communities) hand labeled by the user.
(b) By the Ricci flow process of 20 iterations, the weights of inter-community edges are increased (thick edges in the figure) while the weights of intra-community edges gradually shrink to 0 (thin edges in the figure).
(c) By removing the inter-community edges with high weights, the communities are clearly detected.

Poincaré Ball Manifold

Summary

Attributes	Euclidean	Poincaré Ball
Metric	$g^E = I$(flat)	$\lambda_x^2 g^E$(curved)
Geodesic	Straight Line	Line (or) Arc/Circle
Algebraic structure	Vector Space	Gyrovector Space
Operations	Addition, Subtraction, scalar product, matrix-vector product	Möbius versions
Activation	$f(x)$	$Exp_0^c(f(Log_0^c(x)))$
Distance	$L2 - norm$	$\frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \parallel -x \oplus_c y \parallel)$
Multiclass Logistic Regression	$exp\left(sign(\langle a_k, -p_k + x\rangle)\|a_k\| d_{\mathbb{R}}(x, \mathbb{E}_k)\right)$	$exp\left(sign(\langle -p_k \oplus_c x, a_k\rangle)\sqrt{g_{p_k}^c(a_k, a_k)} d_{\mathbb{H}}(x, \mathbb{H}_k^c)\right)$
Optimization	$x_{t+1} \leftarrow x_t - \alpha g_t$ (SGD)	$x_{t+1} \leftarrow Exp_{x_t}(-\alpha g_t)$ (RSGD)

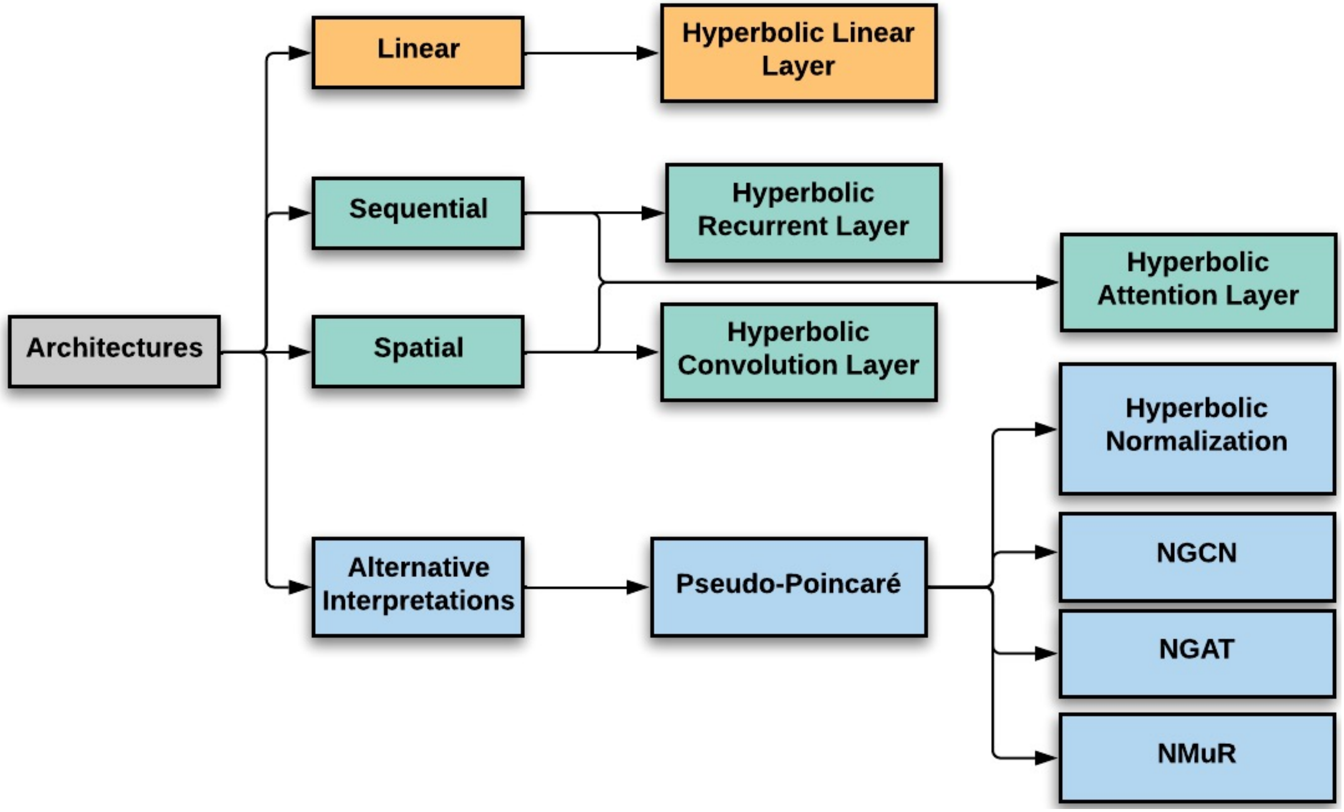


Q&A

Please provide your feedback at:

https://bit.ly/feedback-hyperbolic-tutorial

Part 3: Architectures



Requirements

Development of Neural Networks

Functional requirements for adopting a space into neural networks:

- ❑ Addition (by extension subtraction)
- ❑ Scalar Multiplication
- ❑ Matrix Multiplication
- ❑ Distance Metric
- ❑ Applying Function
- ❑ SoftMax - MLR

Hyperbolic Operations

Development of Neural Networks

Functional requirements for adopting a space into neural networks:

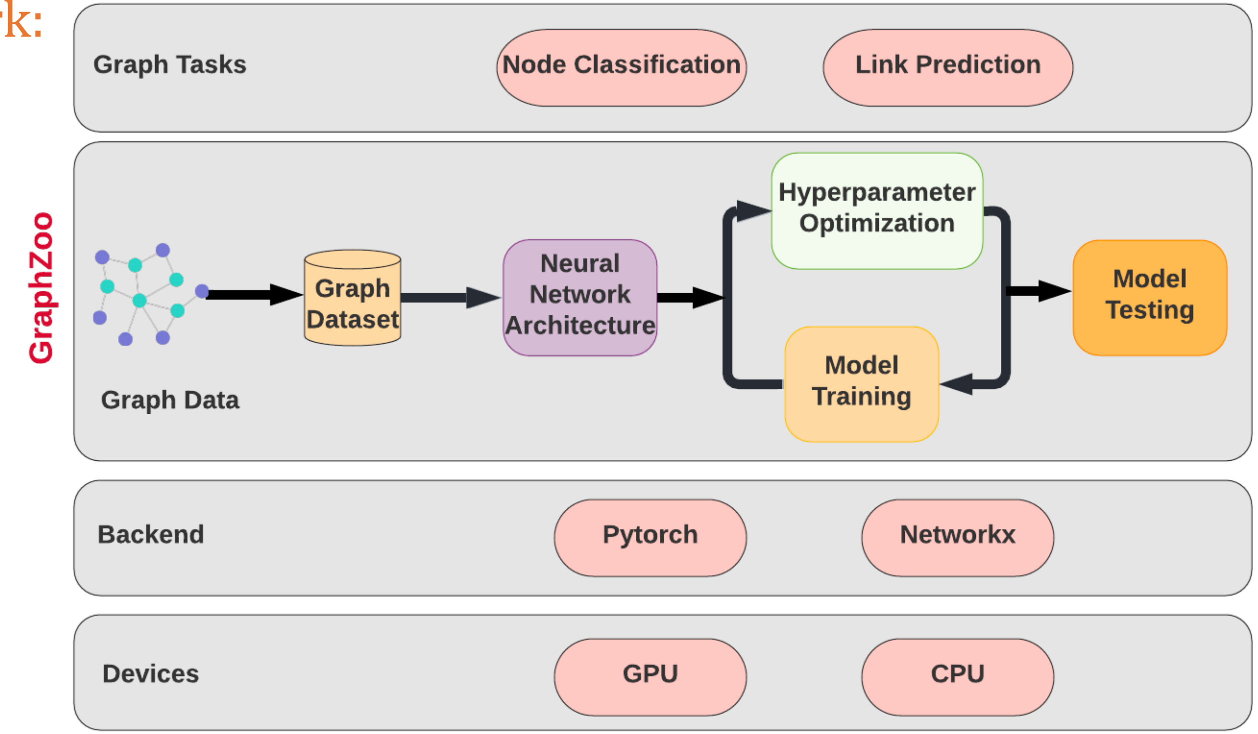
- ✓ Addition (by extension subtraction) $$x \oplus_c a = \frac{(1 + 2ca \cdot x + c \parallel x \parallel^2)a + (1 - c \parallel a \parallel^2)x}{1 + 2ca \cdot x + c^2 \parallel a \parallel^2 \parallel x \parallel^2}$$
- ✓ Scalar Multiplication $$r \otimes_c x = exp_0^c\left(rlog_0^c(x)\right)$$
- ✓ Matrix Multiplication $$[M_1 \quad M_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = M_1 \otimes_c x_1 \oplus_c M_2 \otimes_c x_2$$
- ✓ Distance Metric $$d_{\mathbb{H}}(x, y) = \frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c} \parallel -x \oplus_c y \parallel)$$
- ✓ Applying Function $$f^{\otimes_c}(x) = exp_0^c(f(Log_0^c(x)))$$
- ✓ SoftMax - MLR $$P(y = k|x) \propto exp\left(\frac{\lambda_{p_k}^c \|a_k\|}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}\langle -p_k \oplus_c x, a_k\rangle}{(1 - c\|-p_k \oplus_c x\|^2)\|a_k\|}\right)\right)$$

Hyperbolic Neural Networks

GraphZoo Toolkit

Supports designing a hyperbolic neural network:

- Design a Manifold
- Activation Functions
- Layers
- Optimizers
- Loss function
- Model Architecture (based on task)



Link to Toolkit: https://github.com/reddy-lab/GraphZoo
(Implemented in Pytorch)

Hyperbolic Neural Networks

GraphZoo Toolkit

In this part of the tutorial, we will understand and show the implementation of the following parts:

- Manifolds; Poincaré Ball and Lorentz (same as Klein)
- Activation Functions; General Hyperbolic activation
- Loss function; Hyperbolic distance
- Optimizers; Reimannian SGD and Reimannian AdaM
- Layers; Linear, Recurrent, Convolution and Attention
- Model Architecture; Combine to form a full-fledged model

Hyperbolic Neural Networks

Designing the Manifold

ManifoldParameter

1. Defines the manifold type
2. Defines the curvature

Exponential and Logarithmic Maps

1. Defines the function of exponential map from tangential space to manifold.
2. Defines the function of logarithmic map from manifold to tangential space at a point.

Operations

1. Defines the manifold operations
2. The manifold operations include addition, scalar product, matrix vector product.

Distance

1. Defines the function of hyperbolic distance

Projections

1. Defines the projection operation to maintain the range of hyperbolic space.

Parallel Transport

1. Defines the function of parallel transport from one tangential space to another.

Egrad2Rgrad

1. Defines the function of transforming Euclidean gradient to Riemannian gradient.

Hyperbolic Neural Networks

Euclidean Model: Designing the Manifold

Distance
$d_{\mathbb{R}}(x, y) = \parallel y - x \parallel$

Operations
$x \oplus_c y = x + y$
$\begin{bmatrix} M_1 & M_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = M_1 x_1 + M_2 x_2$

Parallel Transport
$P^c_{0 \to x}(v) = x + v$

Exponential and Logarithmic Maps
$log^c_p(y) = y - p$
$exp^c_p(v) = p + v$

Egrad2Rgrad
$\delta_{\mathbb{R}}(p) = p$

Hyperbolic Neural Networks

Poincaré Ball Model: Designing the Manifold

Distance

$$d_{\mathbb{H}}(x,y) = \frac{2}{\sqrt{c}}\tanh^{-1}(\sqrt{c} \parallel -x \oplus_c y \parallel)$$

Operations

$$x \oplus_c y = \frac{(1+2cy\cdot x + c \parallel x \parallel^2)y + (1 - c \parallel y \parallel^2)x}{1 + 2cy\cdot x + c^2 \parallel y \parallel^2 \parallel x \parallel^2}$$

$$\begin{bmatrix} M_1 & M_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = M_1 \otimes_c x_1 \oplus_c M_2 \otimes_c x_2$$

Parallel Transport

$$P_{0\to x}^c(v) = log_x^c\left(x \oplus_c exp_0^c(v)\right) = \frac{\lambda_0^c}{\lambda_x^c} v$$

Exponential and Logarithmic Maps

$$log_p^c(y) = \frac{2}{\sqrt{c}\lambda_p^c}\tanh^{-1}(\sqrt{c}\| -p \oplus_c y\|)\frac{-p \oplus_c y}{\| -p \oplus_c y\|}$$

$$exp_p^c(v) = p \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_x^c \parallel v \parallel}{2}\right)\frac{v}{\sqrt{c} \parallel v \parallel}\right)$$

Egrad2Rgrad

$$\delta_{\mathbb{H}}(p) = \frac{\delta_{\mathbb{R}}(p)}{\lambda_x^c(p)^2}$$

Hyperbolic Neural Networks

Poincaré Ball Model: Designing the Manifold

Distance
`manifold.poincare.sqdist`

Operations
`manifold.poincare.mobius_add` `manifold.poincare.mobius_matvec`

Parallel Transport
`manifold.poincare.ptransp` `manifold.poincare.ptrans0`

Exponential and Logarithmic Maps

`manifold.poincare.expmap`
`manifold.poincare.expmap0`
`manifold.poincare.logmap`
`manifold.poincare.logmap0`

Egrad2Rgrad
`manifold.poincare.egrad2rgrad`

Hyperbolic Neural Networks

Lorentz Model: Designing the Manifold

Distance

$$d_{\mathbb{L}}(x,y) = \frac{1}{c}\left(\cosh^{-1}\left(-c(\langle x,y\rangle_{\mathbb{L}})\right)\right)^2$$

$$\langle x,y\rangle_{\mathbb{L}} = -x_0y_0 + \sum_{i=1}^{n} x_iy_i$$

Operations

$$x \oplus_c y = Exp_x^c\left(P_{0\to x}^c\left(Log_0^c y\right)\right)$$

$$M \otimes_c x = Exp_0^c(M \times Log_0^c(x))$$

Parallel Transport

$$\alpha = \frac{\langle Log_x^c(y), u\rangle_{\mathbb{L}}}{d_{\mathbb{L}}(x,y)}$$

$$P_{x\to y}^c(u) = \mathrm{u} - \alpha\left(Log_x^c(y) + Log_y^c(x)\right)$$

Exponential and Logarithmic Maps

$$\|v\|_{\mathbb{L}}^c = \sqrt{c\langle v,v\rangle_{\mathbb{L}}}$$

$$Exp_x^c(v) = \cosh(\|v\|_{\mathbb{L}}^c)x + \sinh(\|v\|_{\mathbb{L}}^c)\frac{v}{\|v\|_{\mathbb{L}}^c}$$

$$Log_x^c(y) = d_{\mathbb{L}}(x,y)\frac{y + \langle x,y\rangle_{\mathbb{L}} \times x \times c}{\|y + \langle x,y\rangle_{\mathbb{L}} \times x \times c\|_{\mathbb{L}}^c}$$

Hyperbolic Neural Networks

Lorentz Model: Designing the Manifold

Distance
`manifold.lorentz.sqdist`
`manifold.lorentz.minkowski_dot`

Exponential and Logarithmic Maps
`manifold.lorentz.expmap`
`manifold.lorentz.expmap0`
`manifold.lorentz.logmap`
`manifold.lorentz.logmap0`

Operations
`manifold.lorentz.mobius_add`
`manifold.lorentz.mobius_matvec`

Parallel Transport
`manifold.lorentz.ptransp`
`manifold.lorentz.ptrans0`

Hyperbolic Neural Networks

Activation Function

- Activation Functions; General Hyperbolic activation

Activation Function
$f^{\otimes_c}(x) = exp_0^c(f(log_0^c(x)))$

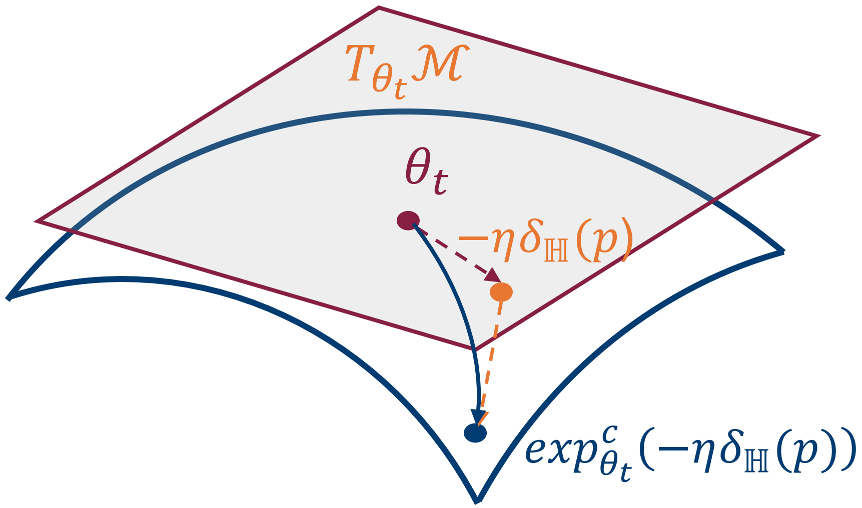
Implementation
`layers.HyperbolicActivation("poincare")` `layers.HyperbolicActivation("lorentz")`

Hyperbolic Neural Networks

Optimizer Functions

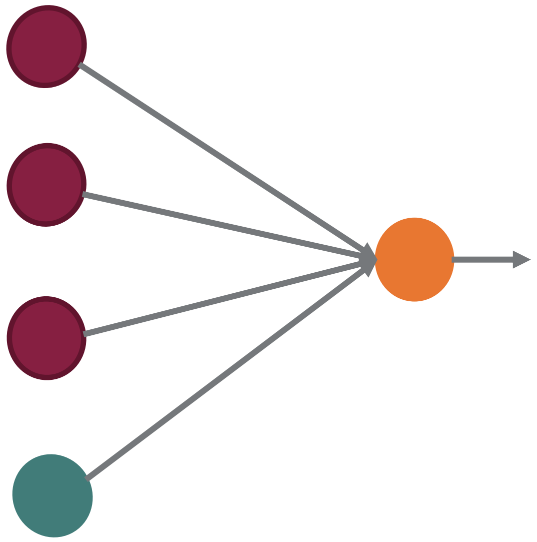
- Optimizers; RSGD and RAdaM

Optimizer Function	Implementation
$\theta_{t+1} = exp^{c}_{\theta_t}(-\eta\delta_{\mathbb{H}}(p))$	grad = manifold.egrad2rgrad(point, grad, c)



Hyperbolic Neural Networks

Layer: Linear Layer



$$[w_1 \quad w_2 \quad w_3 \quad 1]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ b \end{bmatrix}$$

Euclidean:

$w_1 x_1 + w_2 x_2 + w_3 x_3 + \mathrm{b}$

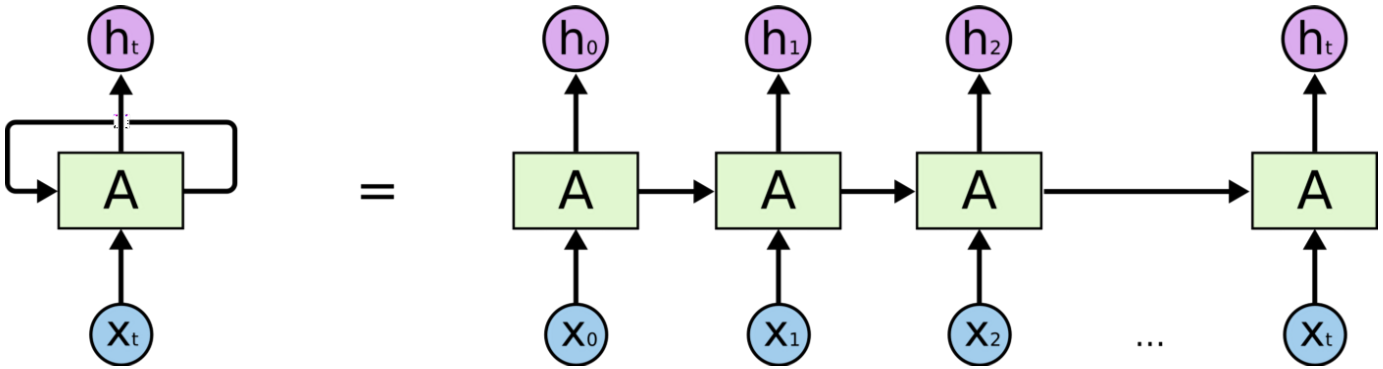
Hyperbolic: Replace with gyrovector operations

$w_1 \otimes_c x_1 \oplus_c w_2 \otimes_c x_2 \oplus_c w_3 \otimes_c x_3 \oplus_c \mathrm{b}$

Implementation
`Change "+" to manifold.mobius_add` `Change "matmul" to manifold.mobius_matvec`

Hyperbolic Neural Networks

Layer: Recurrent Layer



- Formulation is similar to Linear layer, but back-propagation happens over timesteps.
- The same replacement of operations to hyperbolic operations are needed.

Euclidean:

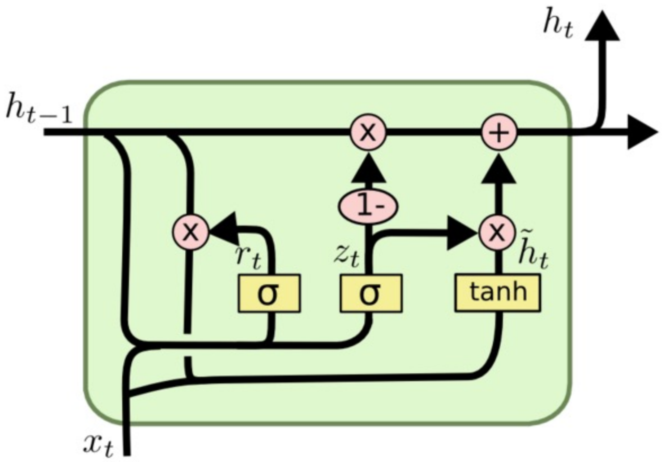
$$h_t = \phi(Wh_{t-1} + Ux_t + b)$$

Hyperbolic:

$$h_t = \phi^{\otimes_c}(W \otimes_c h_{t-1} \oplus_c U \otimes_c x_t \oplus_c b)$$

Hyperbolic Neural Networks

Layer: GRU Layer



Euclidean:

$$r_t = \sigma(W^r r_t + U^r x_t + b^r)$$
$$z_t = \sigma(W^z r_t + U^z x_t + b^z)$$
$$\widetilde{h_t} = \phi(W(r_t \odot h_{t-1}) + U x_t + b)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t}$$

Hyperbolic:

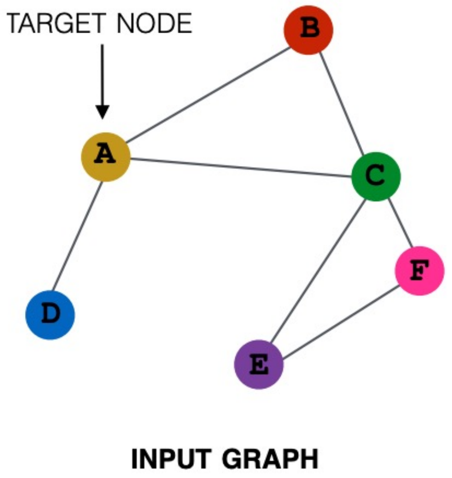
$$r_t = \sigma(log_0^c(W^r \otimes_c r_t \oplus_c U^r \otimes_c x_t \oplus_c b^r))$$
$$z_t = \sigma(log_0^c(W^z \otimes_c r_t \oplus_c U^z \otimes_c x_t \oplus_c b^z))$$

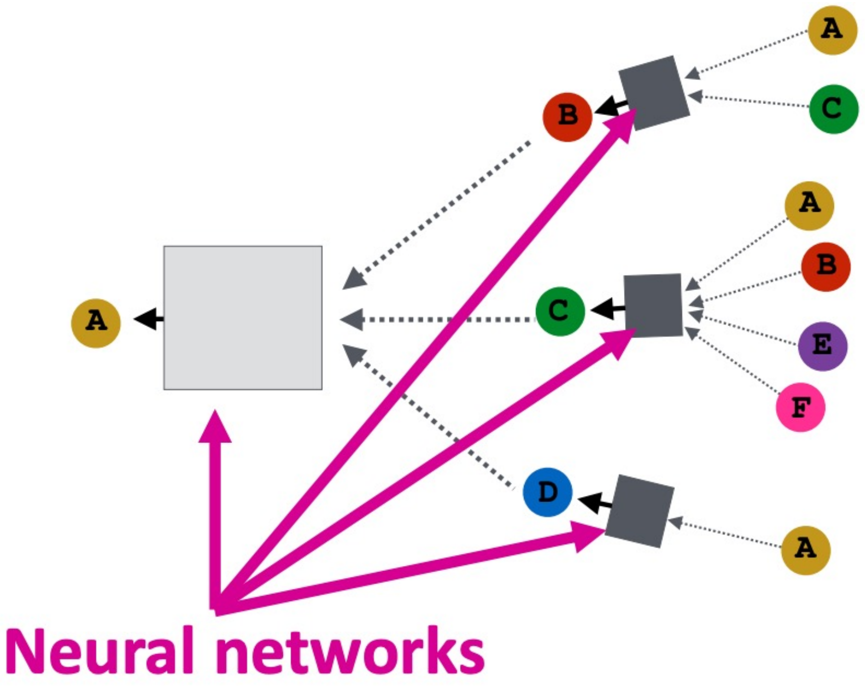
Through Möbius matrix associativity,

$$\widetilde{h_t} = \phi_c^{\otimes}((W(diag(r_t)) \otimes_c h_{t-1}) \oplus_c U \otimes_c x_t \oplus_c b)$$
$$h_t = h_{t-1} \oplus_c diag(z_t) \otimes_c (-h_{t-1} \oplus_c \widetilde{h_t})$$

Hyperbolic Neural Networks

Layer: Convolution Layer





Euclidean:

(feature transform)

$$h_i^{l,\mathbb{R}} = W^l x_i^{l-1,\mathbb{R}} + b^l$$

(neighborhood aggregation)

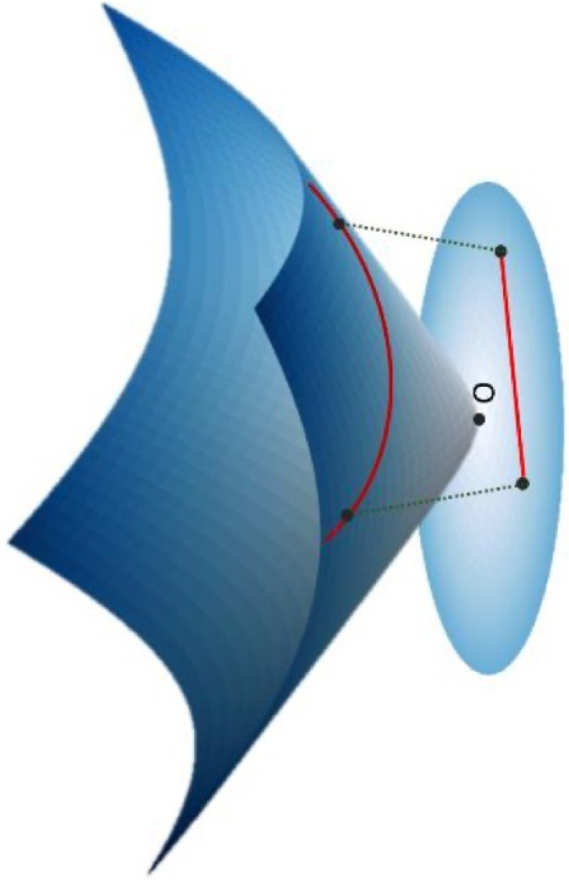
$$x_i^{l,\mathbb{R}} = \sigma\left(h_i^{l,\mathbb{R}} + \sum_{j \in \mathcal{N}(i)} w_{ij} h_j^{l,\mathbb{R}}\right)$$

Image Source: https://medium.com/neuralspace/graphs-neural-networks-in-nlp-dc475eb089de

Hyperbolic Neural Networks

Layer: Convolution Feature Transform

(feature transform)



Hyperbolic:

$$W^l \otimes_c x_i^{l-1,\mathbb{R}} = exp_0^c\left(W^l log_0^c\left(x_i^{l-1,\mathbb{R}}\right)\right)$$

Hyperbolic Transformation Notation

Map from tangent space to Poincaré Ball

Map from Poincaré Ball to tangent space

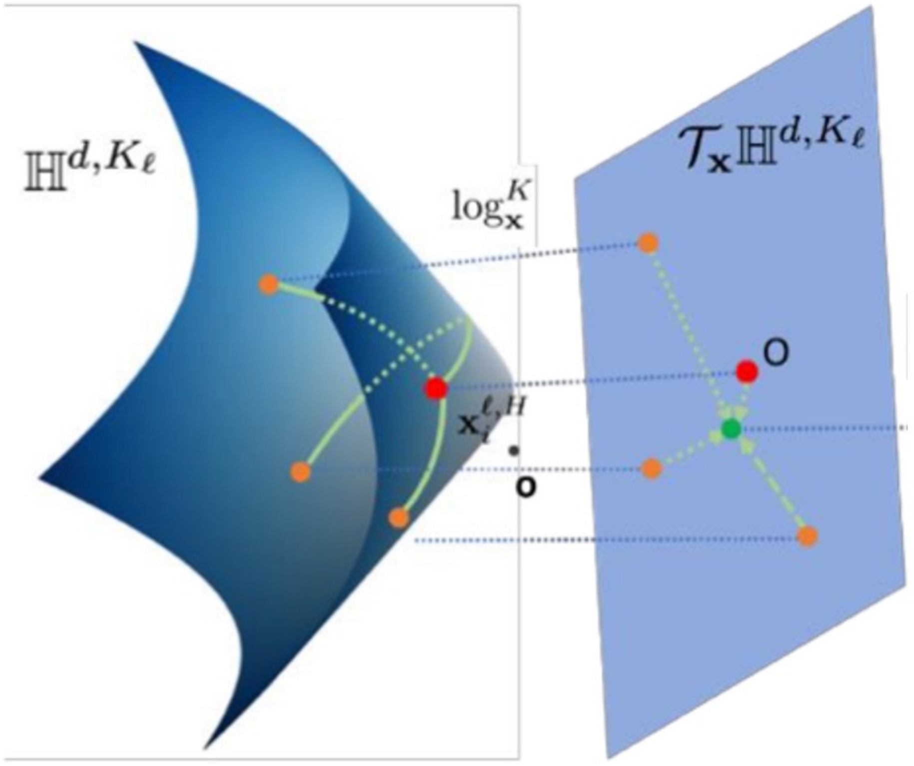
Implementation
Change "+" to `manifold.mobius_add` Change "matmul" to `manifold.mobius_matvec`

Image Source: Chami, Ines, et al. "Hyperbolic graph convolutional neural networks." Advances in neural information processing systems 32 (2019).

Hyperbolic Neural Networks

Layer: Convolution Feature Aggregation

(feature aggregation)



Hyperbolic:

$$w_{ij} = Softmax_{j \in \mathcal{N}(i)}\{MLP(log_0^c(x_i^{\mathbb{H}}) \parallel log_0^c(x_j^{\mathbb{H}}))\}$$

Attention of node i to node j

Concatenate in Tangent space because concatenation is not possible in hyperbolic space.

$$AGG(x^{\mathbb{H}})_i = exp_{x_i^{\mathbb{H}}}^c\left(\sum_{j \in \mathcal{N}(i)} w_{ij} log_{x_i^{\mathbb{H}}}^c(x_j^{\mathbb{H}})\right)$$

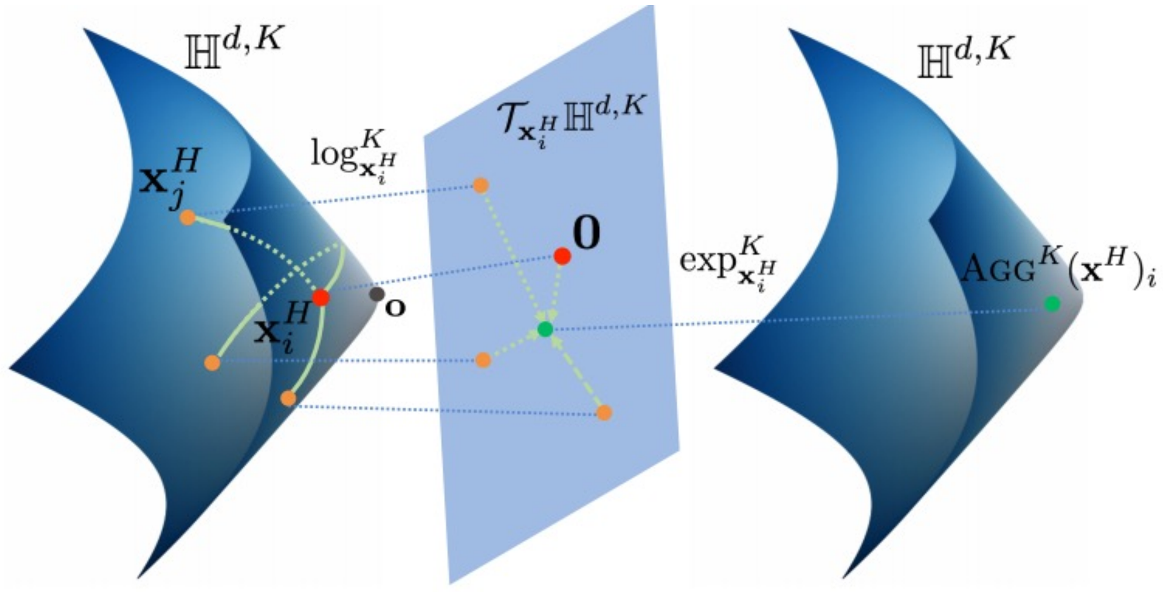
Implementation

1. manifold.logmap0 over all inputs.
2. Concatenate after transformation.
3. Apply Linear layer and Softmax.

Image Source: Chami, Ines, et al. "Hyperbolic graph convolutional neural networks." Advances in neural information processing systems 32 (2019).

Hyperbolic Neural Networks

Layer: Convolution Combined Pipeline



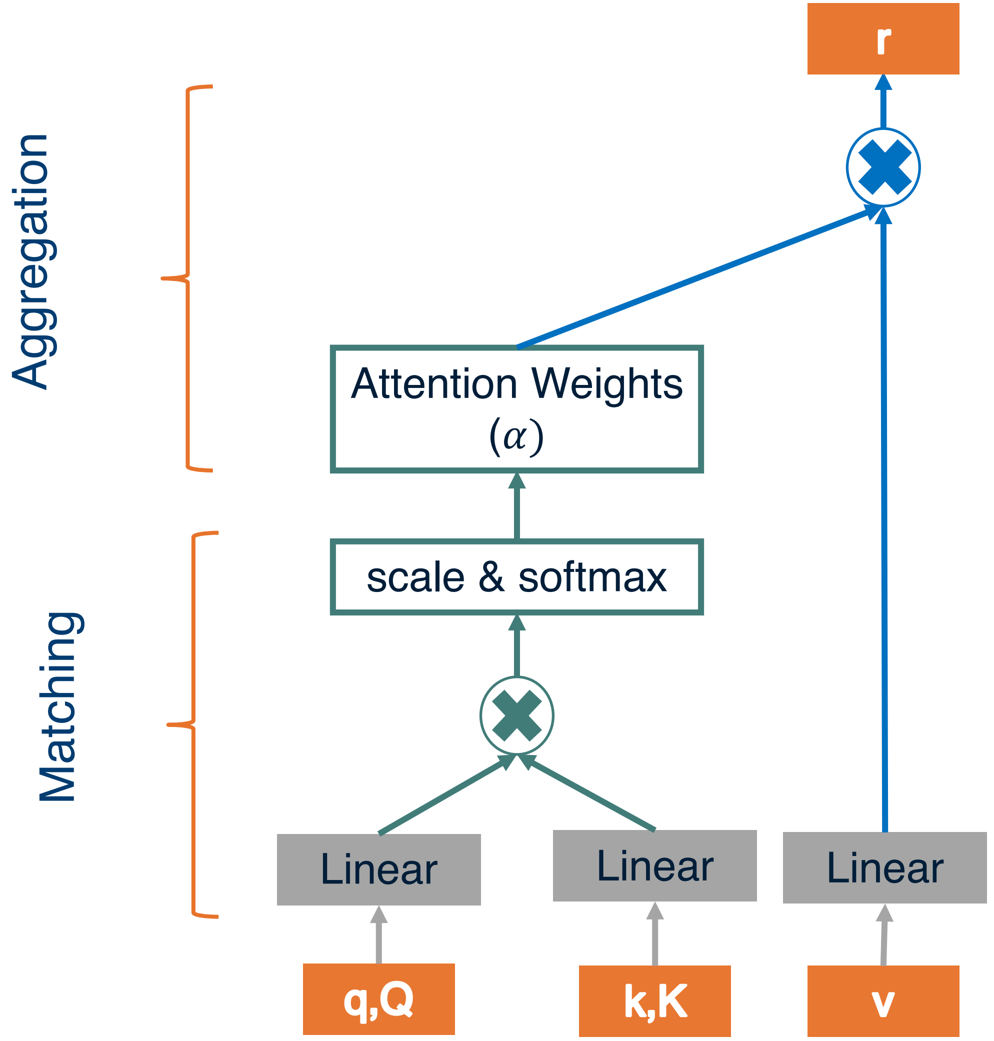
Feature Transform → Aggregate → Activation

Implementation
1. Apply hyperbolic linear layer over all inputs in the hyperbolic space. 2. Aggregate all inputs with aggregation layer in tangent space at origin. 3. Transform them back to hyperbolic space and apply hyperbolic activation.

Image Source: Chami, Ines, et al. "Hyperbolic graph convolutional neural networks." Advances in neural information processing systems 32 (2019).

Hyperbolic Neural Networks

Layer: Attention Layer



Euclidean:

(matching; calculating attention weights)

$$\alpha(q_i, k_j) = \alpha_{ij} = \frac{q_i k_j}{\sqrt{d}}$$

(aggregation)

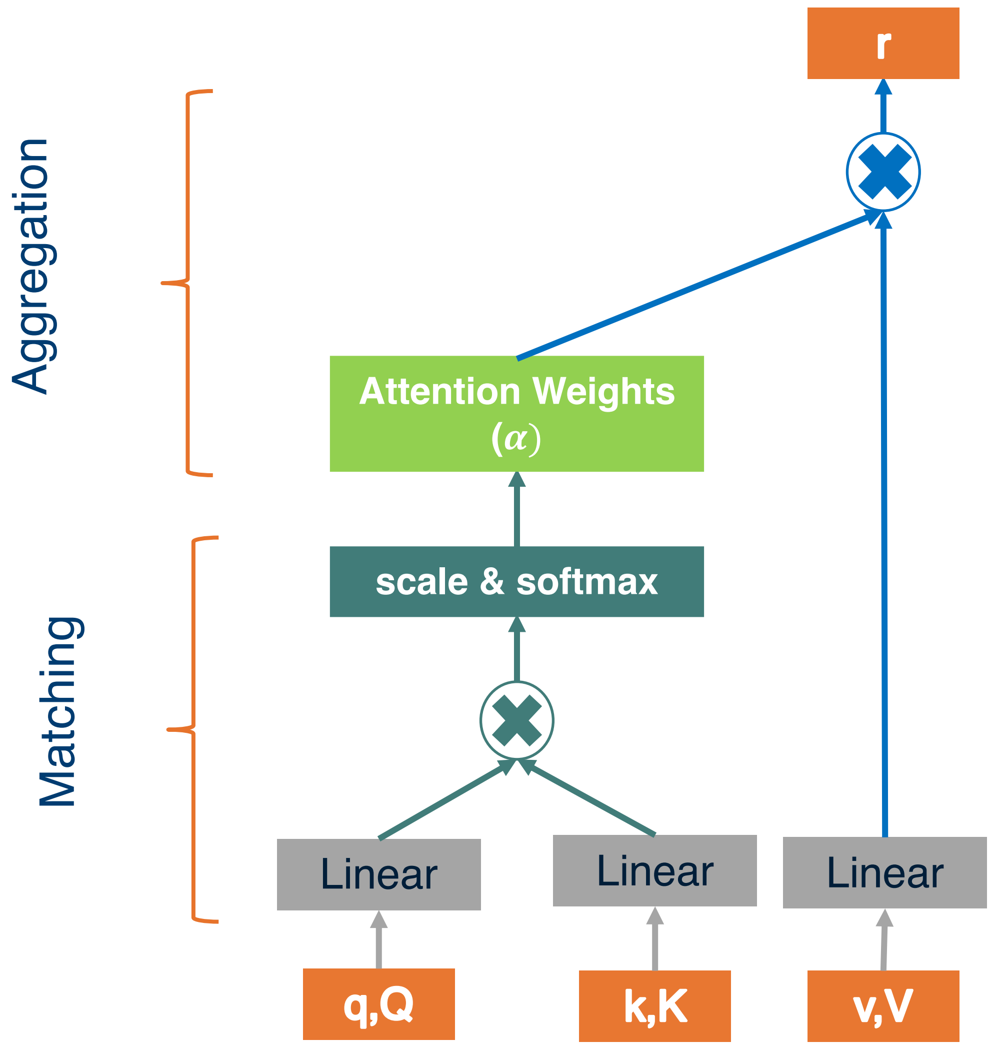
$$r_i\left(\{\alpha_{ij}\}_j, \{v_{ij}\}_j\right) = \sum_j softmax(\alpha_{ij})v_{ij}$$

Challenge:

- No direct replacement for softmax in Poincaré ball (division replacement not defined)
- Inner product in Euclidean is used as a distance metric. Not so in hyperbolic space
- Calculating midpoint for aggregation is not well defined.

Hyperbolic Neural Networks

Layer: Attention Layer



Euclidean:

(matching; calculating attention weights)

$$\alpha(q_i, k_j) = \alpha_{ij} = \frac{q_i k_j}{\sqrt{d}}$$

(aggregation)

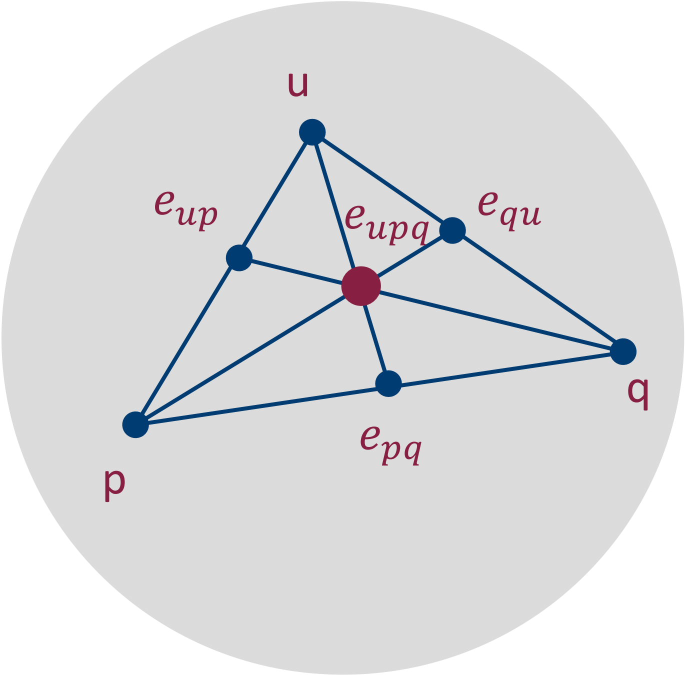
$$r_i\left(\{\alpha_{ij}\}_j, \{v_{ij}\}_j\right) = \sum_j softmax(\alpha_{ij}) v_{ij}$$

Solution:

- Use hyperbolic distance for matching
- Use Klein model and its definition of Einstein midpoint
- Softmax possible due to Euclidean nature of aggregation.

Hyperbolic Neural Networks

Layer: Attention Layer (Einstein Midpoint)



Lorentz Factor acts as weightage for calculating the mean; $\gamma(v) = \frac{1}{\sqrt{1-\|v\|^2}}$

$$e_{pq} = \frac{\gamma(p)p + \gamma(q)q}{\gamma(p) + \gamma(q)}$$

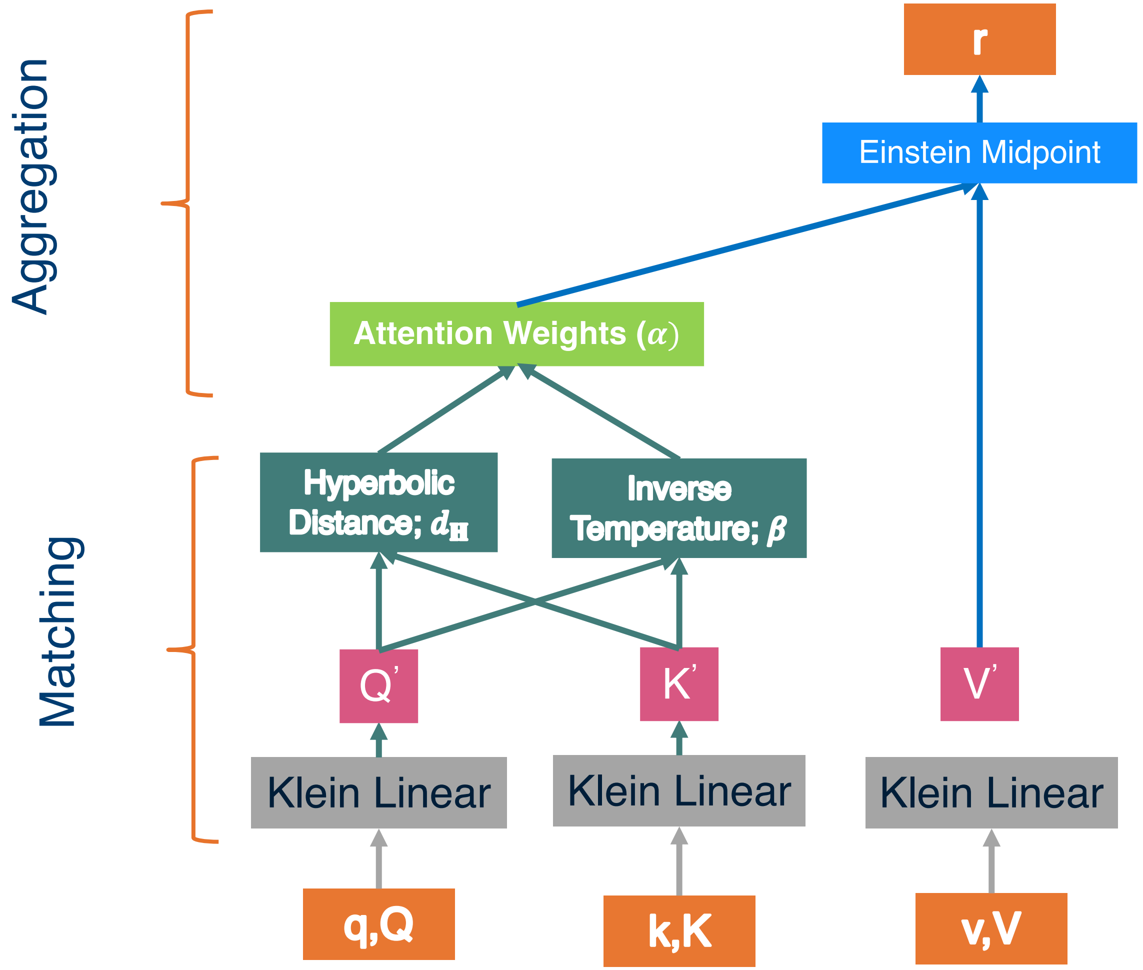
$$e_{qu} = \frac{\gamma(q)q + \gamma(u)u}{\gamma(q) + \gamma(u)}$$

$$e_{up} = \frac{\gamma(u)u + \gamma(p)p}{\gamma(u) + \gamma(p)}$$

$$e_{upq} = \frac{\gamma(u)u + \gamma(p)p + \gamma(q)q}{\gamma(u) + \gamma(p) + \gamma(q)}$$

Hyperbolic Neural Networks

Layer: Attention Layer



Hyperbolic:

(matching; calculating attention weights)

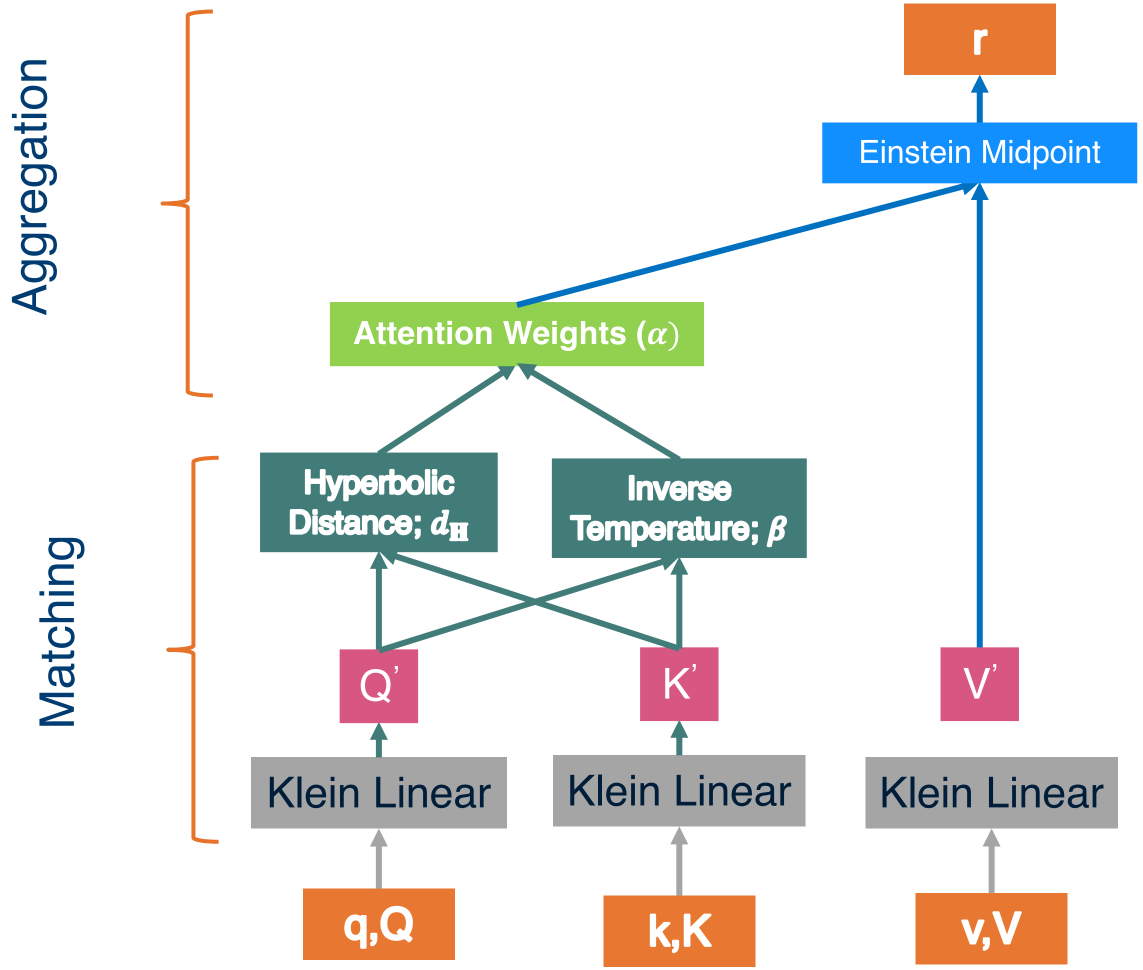
$$\alpha(q_i, k_j) = f(-\beta d_{\mathbb{H}}(q_i, k_j) - c)$$

(aggregation)

$$r_i\left(\{\alpha_{ij}\}_j, \{v_{ij}\}_j\right) = \sum_j \left[\frac{\alpha_{ij}\gamma(v_{ij})}{\sum_l \alpha_{il}\gamma(v_{il})}\right] v_{ij}; \gamma(v) = \frac{1}{\sqrt{1-\|v\|^2}}$$

Hyperbolic Neural Networks

Layer: Attention Layer



Implementation

1. Transform query, key and value with the Klein Linear layer.
2. Calculate hyperbolic distance (manifolds.lorentz.sqdist) and inverse temperature (manifolds.lorentz.beta) using the query and key.
3. Multiply them and apply a Linear layer with bias to get the attention weights.
4. Aggregate the attention weights and values using Einstein midpoint to get the final dense representation.

Hyperbolic Neural Networks

Practical Aspects

- Weights and biases can lie in either hyperbolic or Euclidean space, depending on the design choice.
  - Optimization algorithm (AdaM or RAdaM) needs to be chosen accordingly
- Applying incorrect operation to a corresponding manifold can cause instability in specific models.
- Exponential maps transform features non-linearly, activation may not be needed.
- Max pooling and Batch normalization are direct.
  - Mean pooling can be derived with gyro operations (Einstein midpoint for Lorentz manifold).

Hyperbolic Neural Networks

Practical Aspects

- In some cases one should use tangent space at origin or at a current point.
  - All transformation operations lead to loss of information, hence, it is preferred to operate in a tangent space at the exact point if possible. However, that is generally not feasible.
  - Hence, tangent space at a point is generally used when there is a source of operation, such as, root node for aggregation in convolution layers.
  - However, in most cases of transformations on singular points, tangent space at origin is sufficient.

Hyperbolic Neural Networks

Extensions of the hyperbolic operations

Due to the practical aspects of mixing Euclidean operations and hyperbolic operations, new approaches aim to unify the benefits into any one space.

- Move everything to hyperbolic space;

  Hyperbolic Neural Networks ++

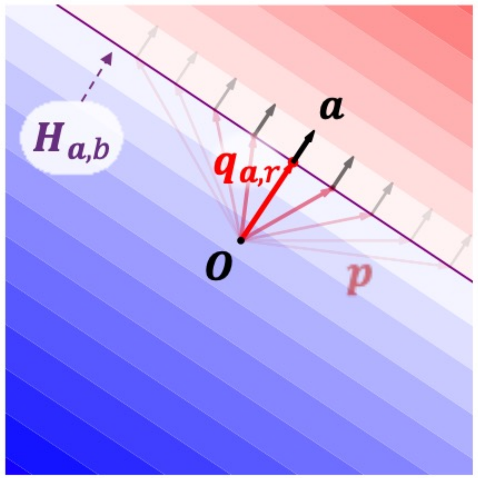
- Move everything to Euclidean space;

  Pseudo-Poincaré framework

Hyperbolic Neural Networks

Hyperbolic Networks ++

This is a new formalization of the Poincaré space. The current methods depend on the tangent space for several operations and the frequent back and forth mapping is both expensive and prone to a loss of data.



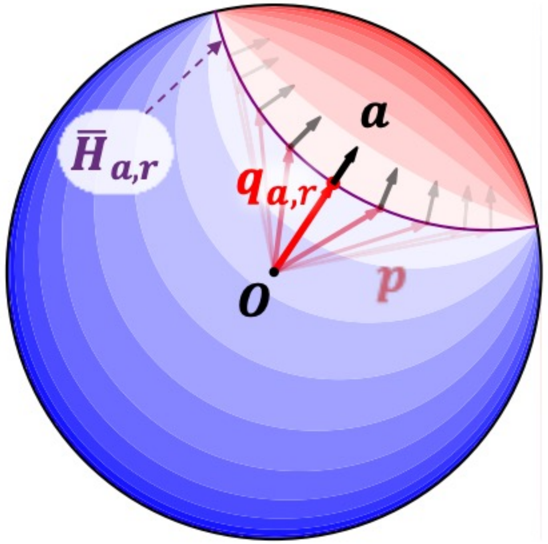


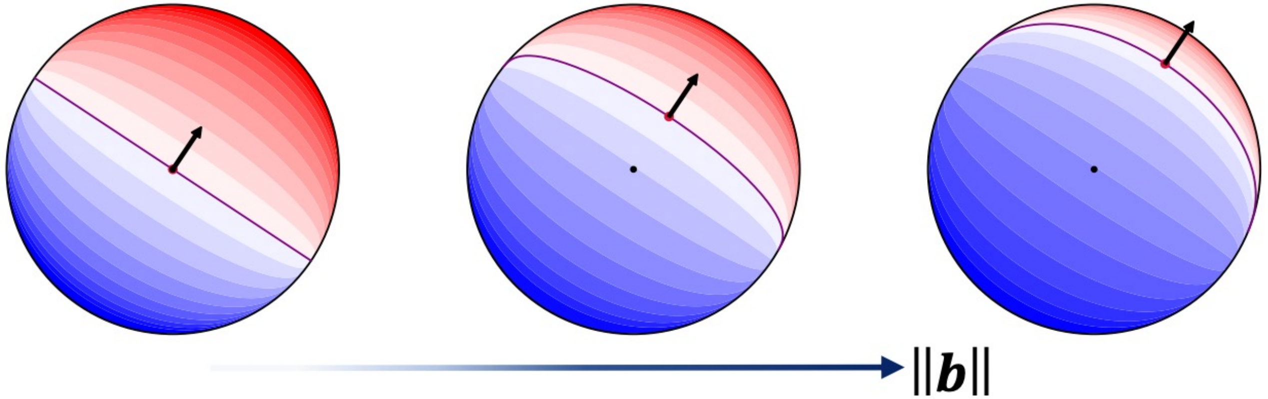
Image Source: Shimizu, Ryohei, Yusuke Mukuta, and Tatsuya Harada. "Hyperbolic neural networks++." ICLR (2021).

Hyperbolic Neural Networks

Hyperbolic Networks ++

Poincaré Ball

$$y = exp_0^c\left(W log_0^c(x)\right) \oplus_c b$$



Hyperbolic Neural Networks ++

$$y = \mathcal{F}^c(x; Z, r) = \mathrm{w}\left(1 + \sqrt{1 + c\|w\|^2}\right)^{-1}$$

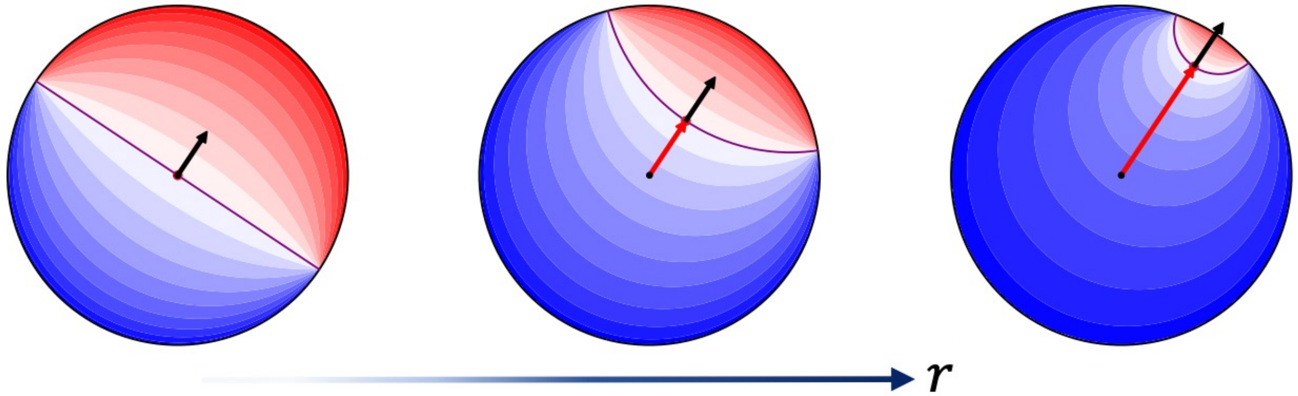


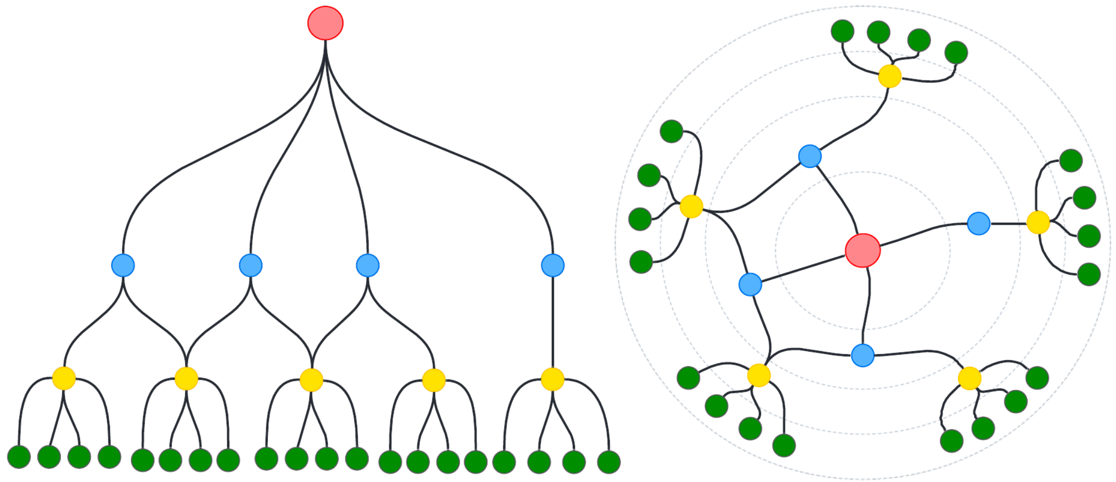
Image Source: Shimizu, Ryohei, Yusuke Mukuta, and Tatsuya Harada. "Hyperbolic neural networks++." ICLR (2021).

Hyperbolic Neural Networks

Pseudo-Poincaré framework

Pseudo-Poincaré space identifies the specific advantages of hyperbolic space and formulates it in the Euclidean space.

The primary advantage of Poincaré space is that it encodes hierarchical relations with exponential increase in volume. But it faces certain practical challenges;



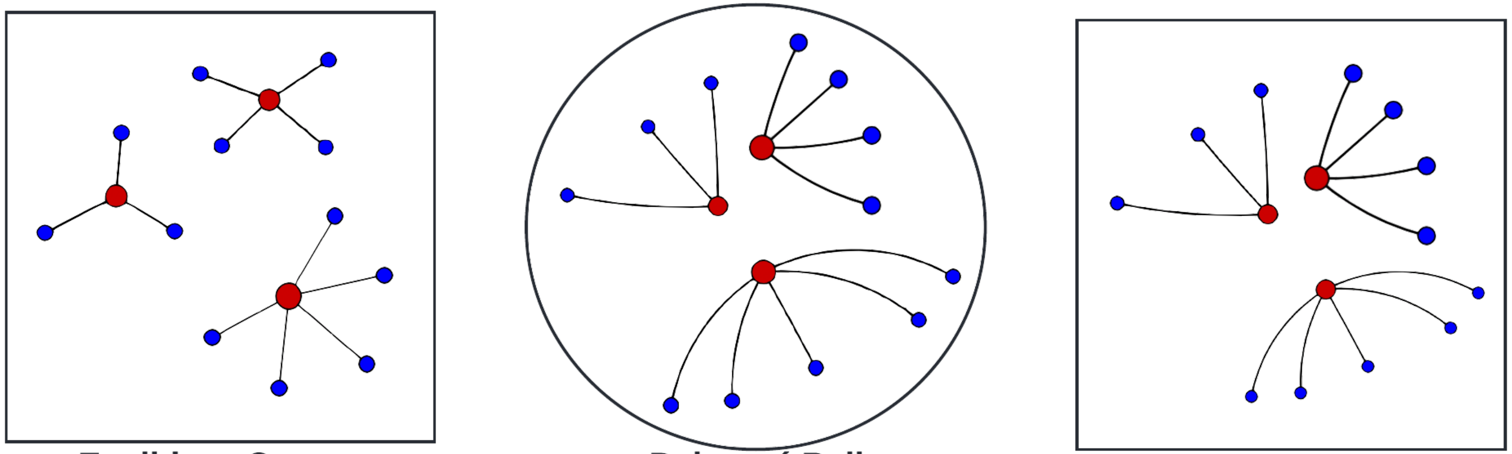
Hyperbolic Neural Networks
Pseudo-Poincaré framework

Challenges:

1. Non-scalability due to inverse hyperbolic functions.
2. Unstable training and gradient descent due to skips between hyperbolic and Euclidean space.
3. Non-closure of hyperbolic space.
4. Not-trivially extensible to complex Euclidean architectures.

Hyperbolic Neural Networks

Pseudo-Poincaré framework



Euclidean Space   Poincaré Ball   Pseudo-Poincaré

Solution:

- Use hyperbolic mapping as a normalization function.
- The use of hyperbolic space can be limited to the input and output for expanding and contracting the representational space.

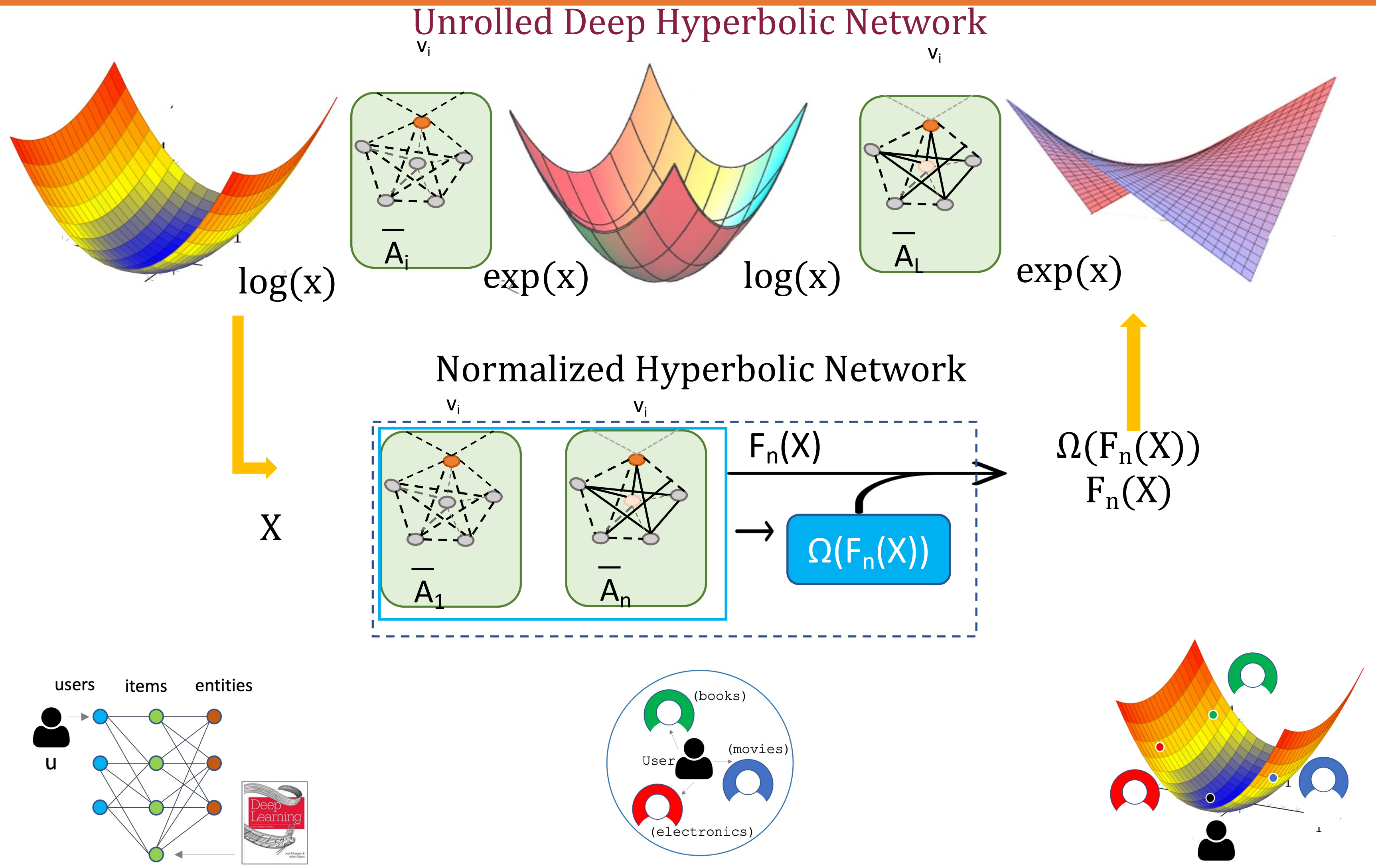
$$NF_n^{\otimes^c}(x) = \Omega\big(F_n(x)\big)F_n(x)$$

$$NGCN_n^{\otimes^c}(x) = \Omega\big(GCN_n(x)\big)GCN_n(x)$$

$$NGAT_n^{\otimes^c}(x) = \Omega\big(GAT_n(x)\big)GAT_n(x)$$

Hyperbolic Neural Networks

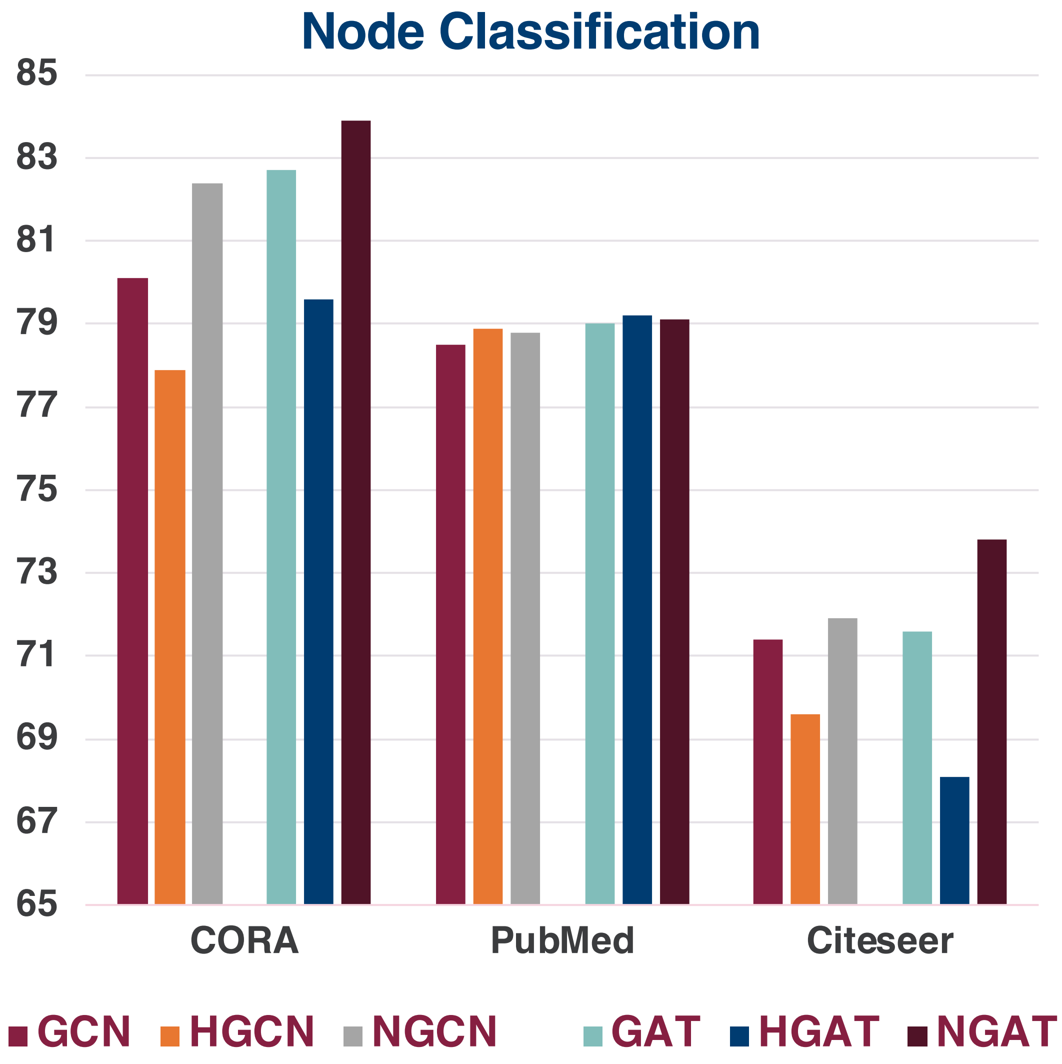
Pseudo-Poincaré: Hyperbolic Normalization

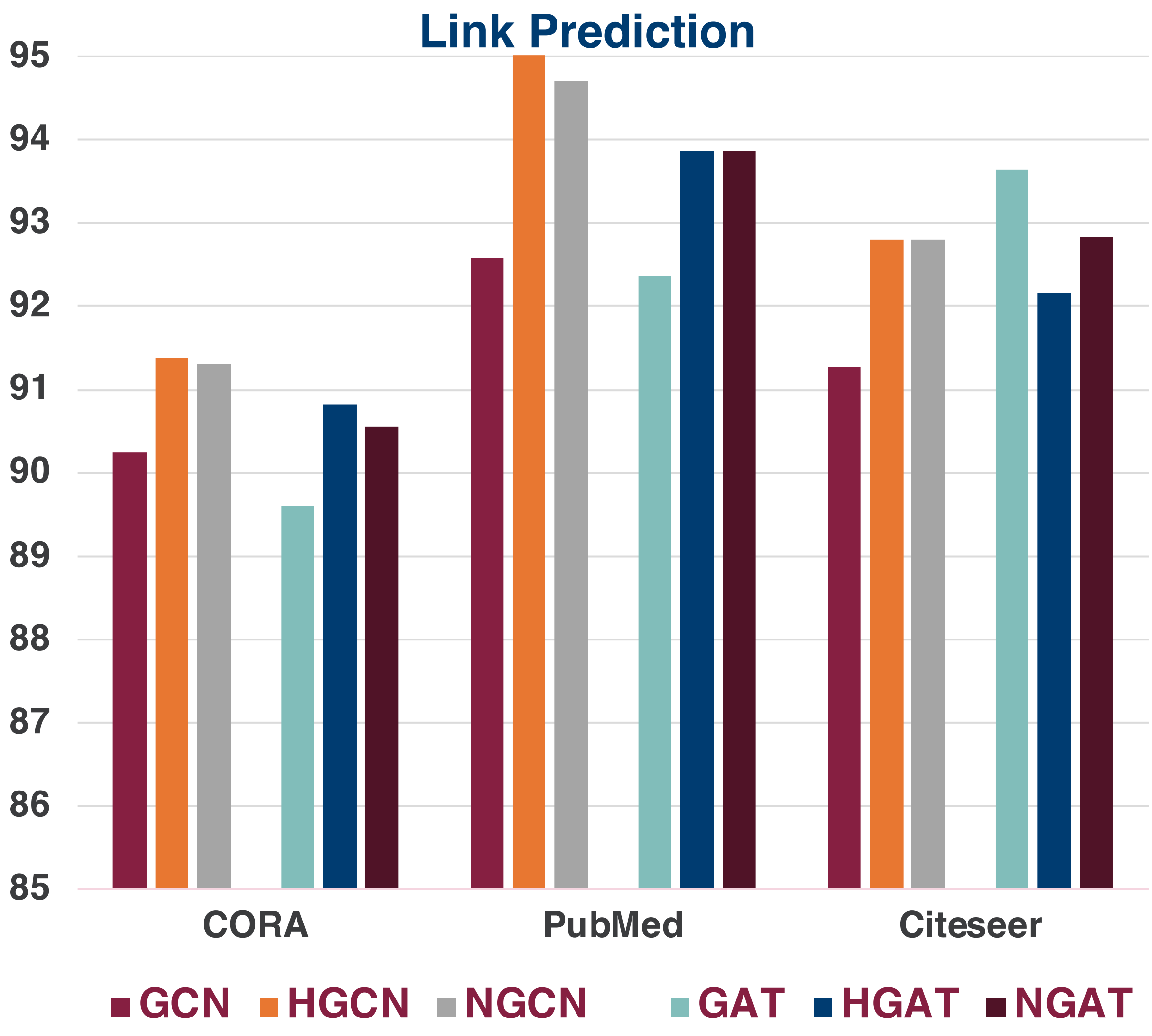


Multi-relational Graph   Poincaré Ball   Hyperbolic space

Hyperbolic Neural Networks

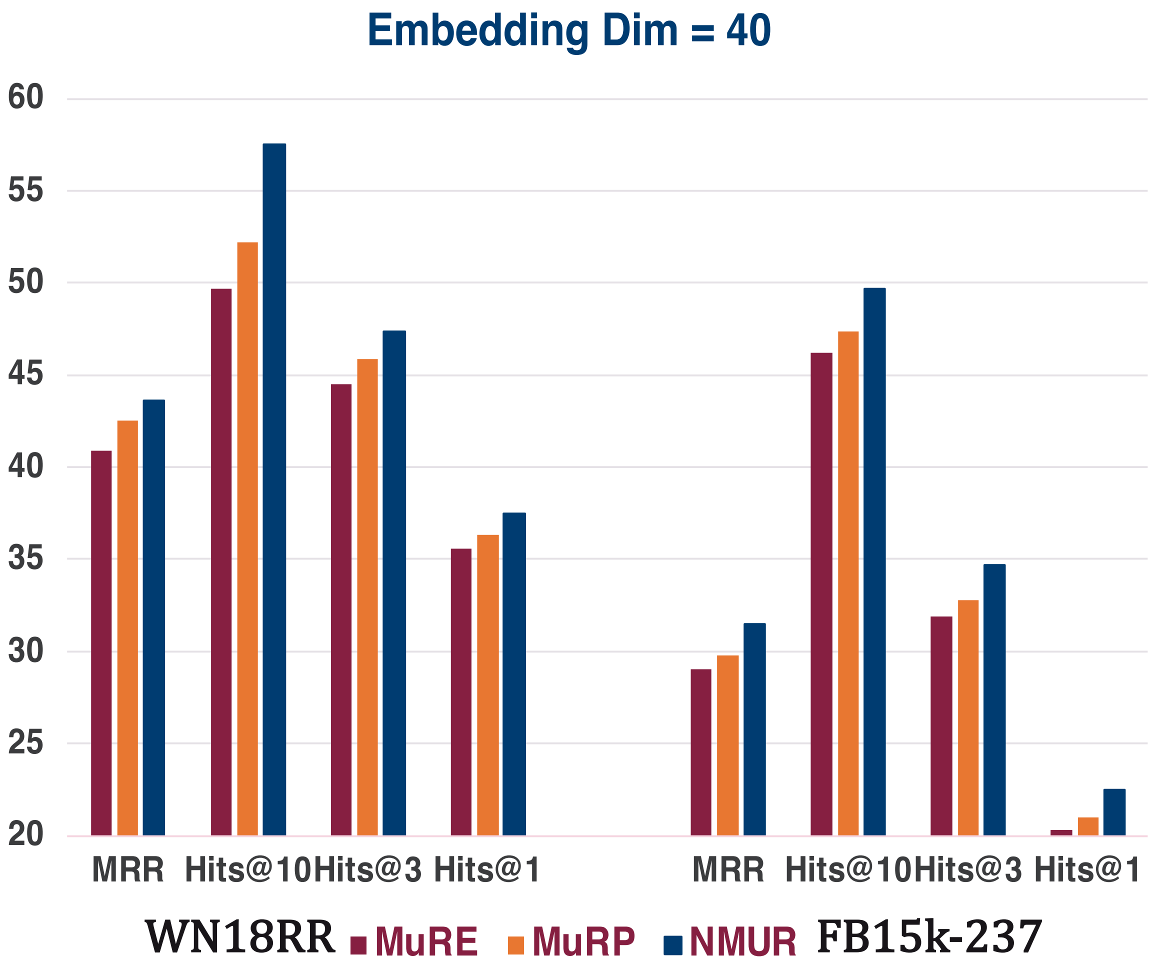
Pseudo-Poincaré: Results on Graph Processing tasks

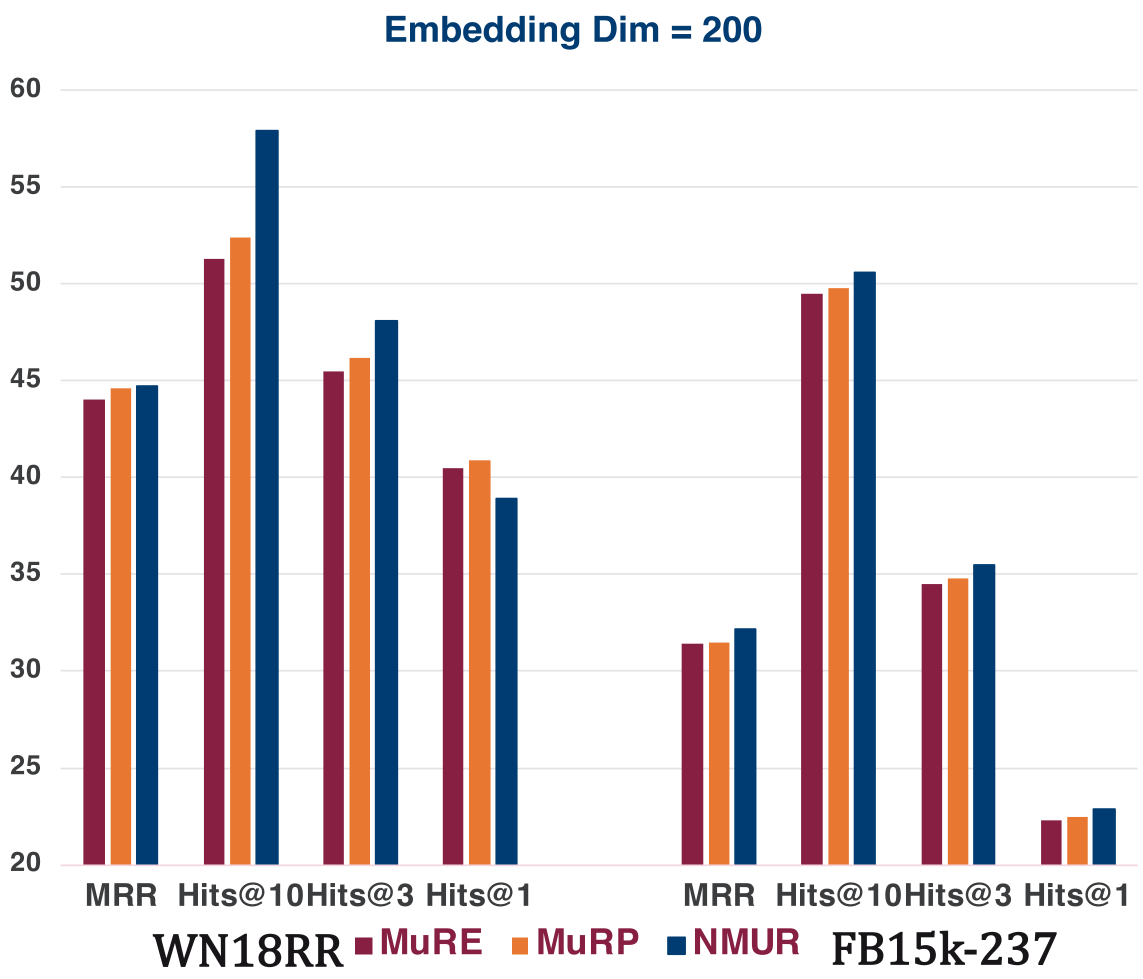




Hyperbolic Neural Networks

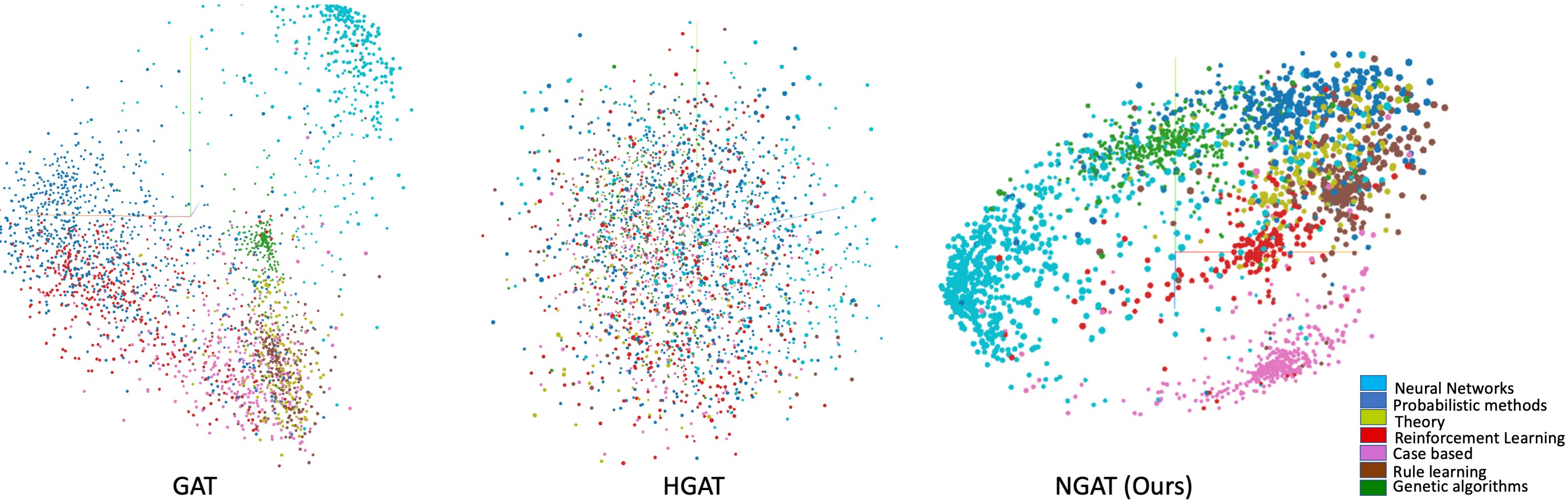
Pseudo-Poincaré: Multi-relational Representations





Hyperbolic Neural Networks

Pseudo-Poincaré: Multi-relational Representations

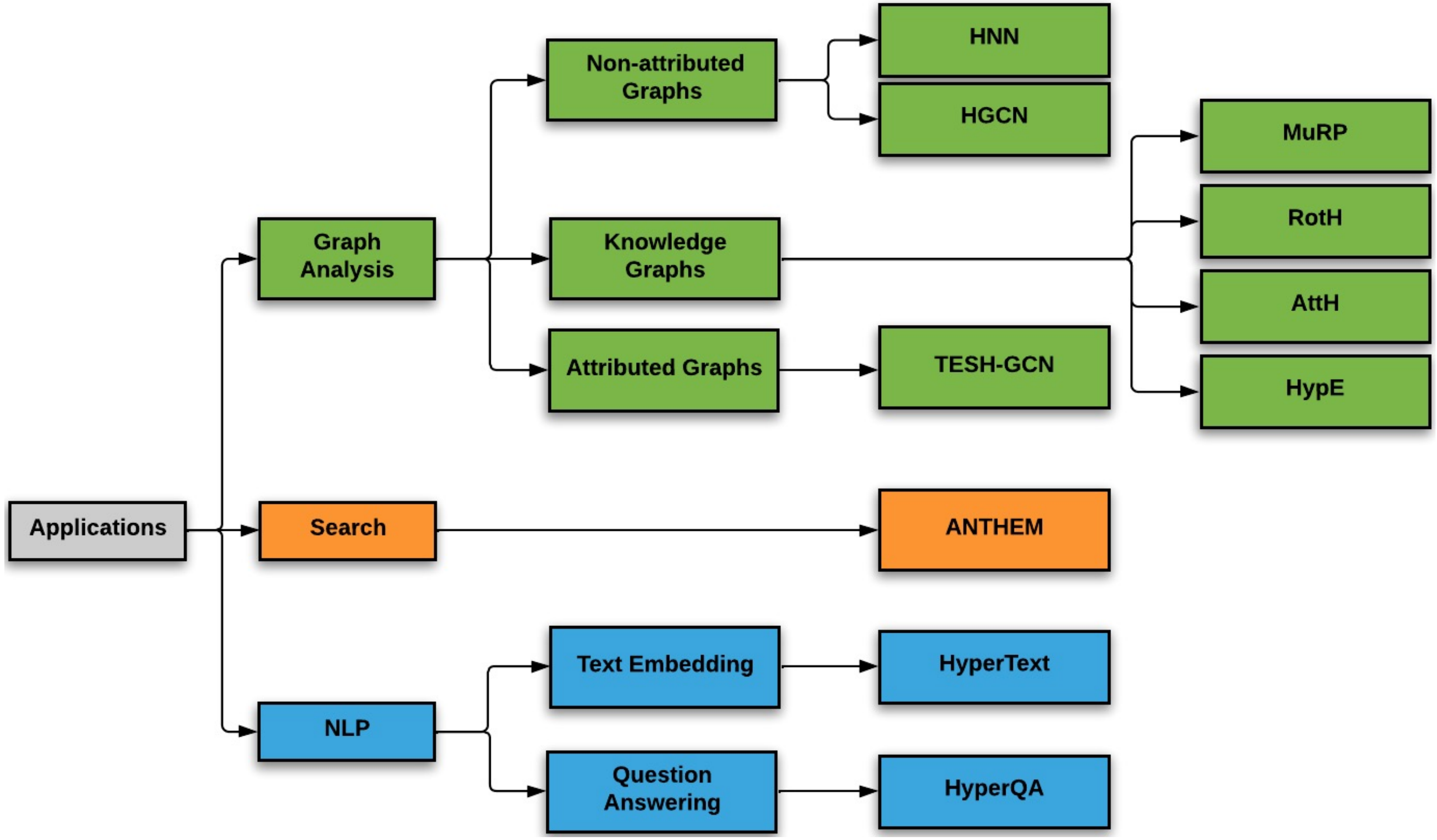


Q&A

Please provide your feedback at:

https://bit.ly/feedback-hyperbolic-tutorial

Part 4: Applications

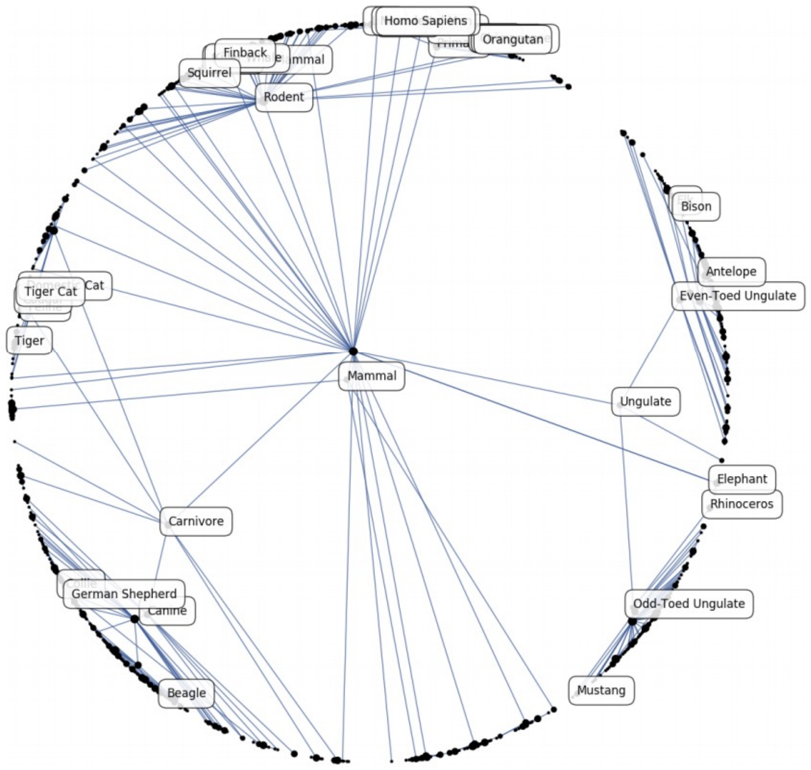


Overview of Applications

Graphs, Knowledge Graphs, Search, and NLP

Advantages of Hyperbolic Networks is primarily observed in cases when hierarchical relations exist in the underlying datasets.

1. Graph Analysis



Mammals in WordNet*

* Nickel, Maximillian, and Douwe Kiela. "Poincaré embeddings for learning hierarchical representations." *Advances in neural information processing systems* 30 (2017).

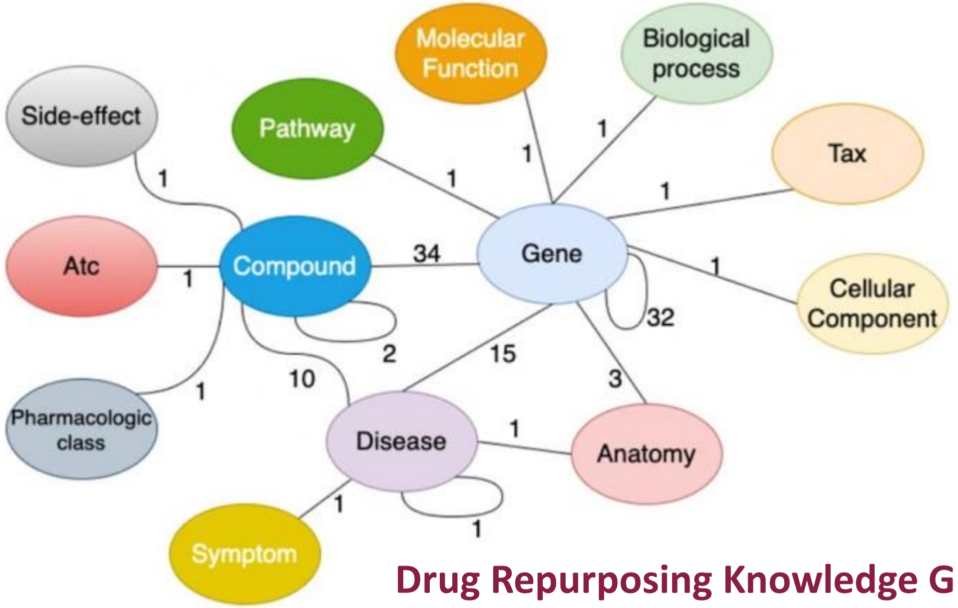
Overview of Applications

Graphs, Knowledge Graphs, Search, and NLP

Advantages of Hyperbolic Networks is primarily observed in cases when hierarchical relations exist in the underlying datasets.

While graph datasets are general candidates for hyperbolic networks, they have also been used to exploit the hierarchy in the following applications:

1. Graph Analysis
2. Knowledge Graphs



Drug Repurposing Knowledge Graph*

* Ioannidis, Vassilis N., et al. "Drkg-drug repurposing knowledge graph for covid-19." *arXiv preprint arXiv: 2010.09600* (2020).

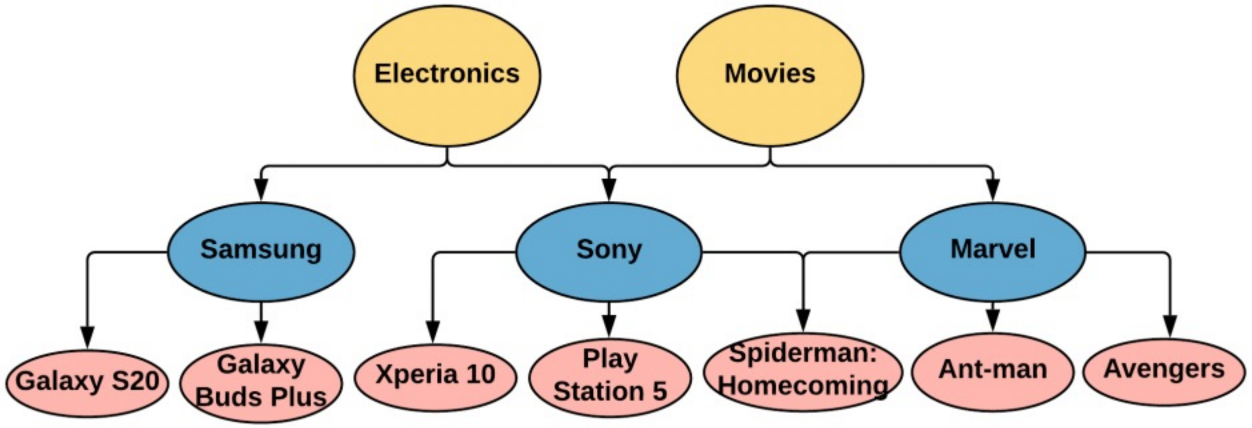
Overview of Applications

Graphs, Knowledge Graphs, Search, and NLP

Advantages of Hyperbolic Networks is primarily observed in cases when hierarchical relations exist in the underlying datasets.

While graph datasets are obvious candidates for hyperbolic networks, they have also been used to exploit the hierarchy in the following applications:

1. Graph Analysis
2. Knowledge Graphs
3. Search



Hierarchy in Product Catalogues*

*Choudhary, Nurendra, et al. "ANTHEM: Attentive Hyperbolic Entity Model for Product Search." WSDM 22 (2022): 21-25.

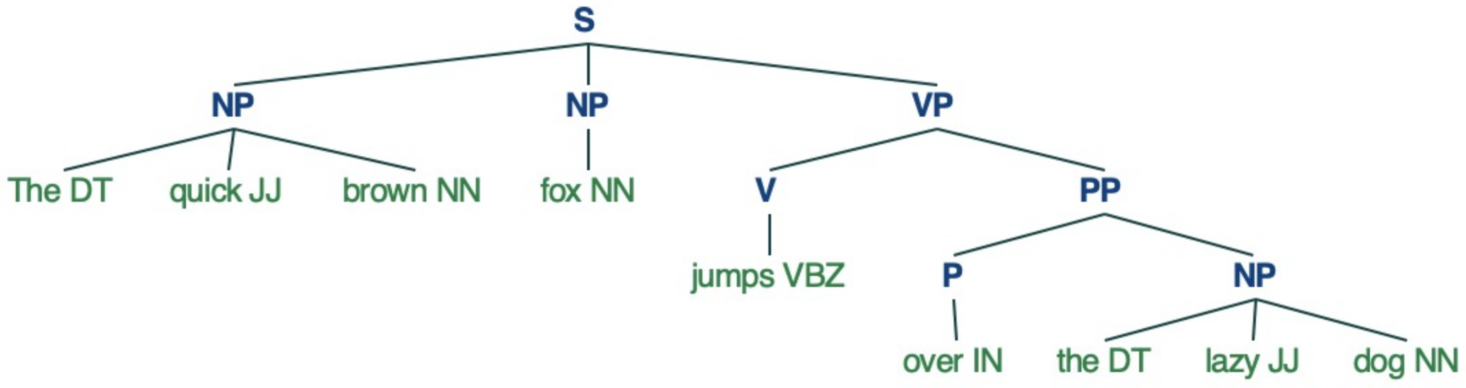
Overview of Applications

Graphs, Knowledge Graphs, Search, and NLP

Advantages of Hyperbolic Networks is primarily observed in cases when hierarchical relations exist in the underlying datasets.

While graph datasets are obvious candidates for hyperbolic networks, they have also been used to exploit the hierarchy in the following applications:

1. Graph Analysis
2. Knowledge Graphs
3. Search
4. Natural Language Processing



Parse trees of Sentences*

*https://www.geeksforgeeks.org/syntax-tree-natural-language-processing/

Overview of Applications

Graphs, Knowledge Graphs, Search and NLP

1. Graphs
   - ❑ Hyperbolic Neural Networks
   - ❑ Hyperbolic Graph Convolutional Neural Networks

2. Knowledge Graphs
   - ❑ Multi-relational Poincaré Embeddings
   - ❑ Low-Dimensional Hyperbolic Knowledge Graph Embeddings
   - ❑ Self-Supervised Hyperboloid Representations from Logical Queries over Knowledge Graphs

3. Search
   - ❑ ANTHEM: Attentive Hyperbolic Entity Model for Product Search

4. Natural Language Processing
   - ❑ Hyperbolic Representation Learning for Fast and Efficient Neural Question Answering
   - ❑ Text Enriched Sparse Hyperbolic Graph Convolution Network

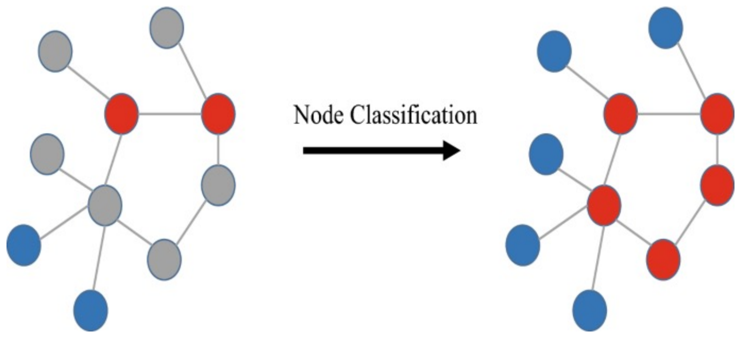
Applications

Graph Analysis: Introduction

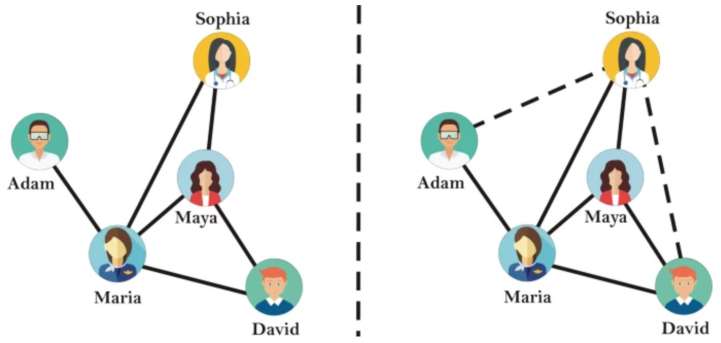
Graphs are essential data structures that contain attributed (or) non-attributed nodes connected by edges.

Several popular problems:

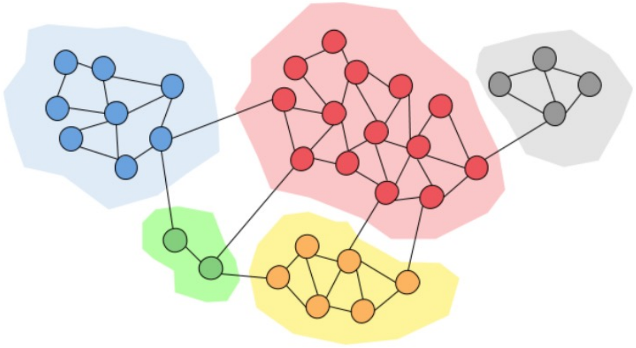
1. Node Classification
2. Link Prediction
3. Community Detection



Node Classification[1]



Link Prediction[2]



Community Detection[3]

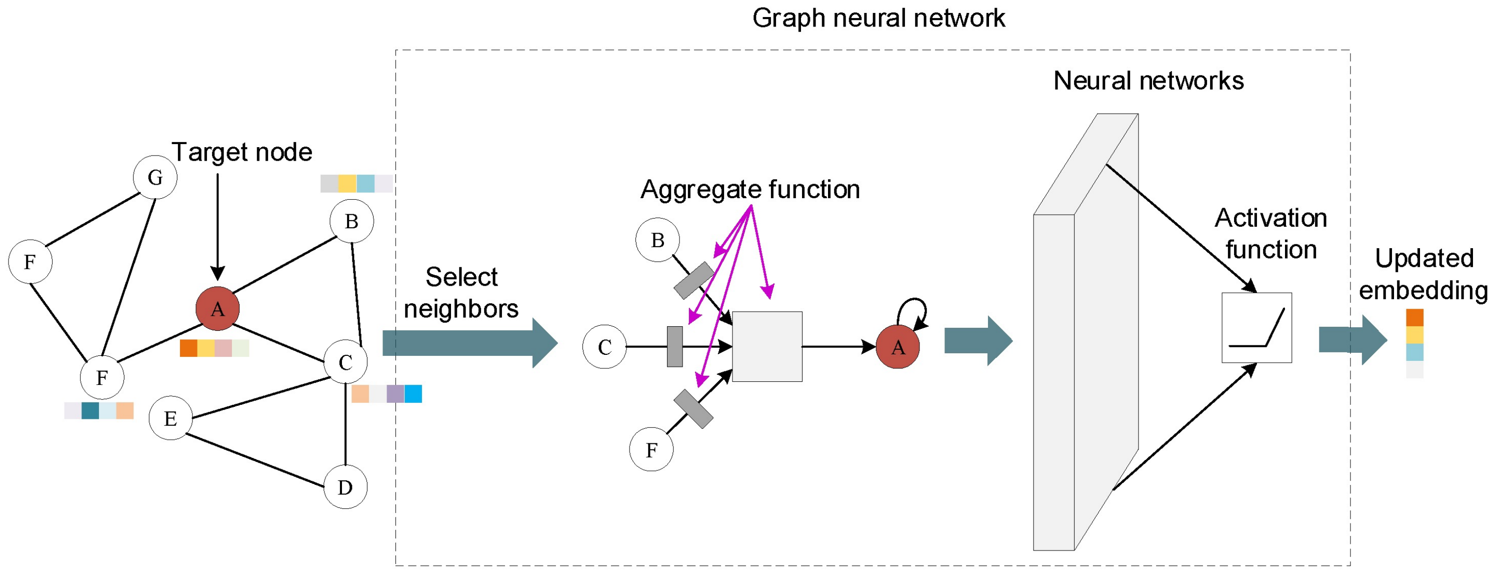
1. Chen, Linjun, Xingyi Liu, and Zexin Li. "Nonlinear Graph Learning-Convolutional Networks for Node Classification." Neural Processing Letters (2021): 1-10.
2. Ahmad, I., Akhtar, M. U., Noor, S., & Shahnaz, A. (2020). Missing link prediction using common neighbor and centrality based parameterized algorithm. Scientific reports.
3. Community Detection Algorithms, Thamindu Dilshan Jayawickrama, Towards Datascience.

Applications

Graph Analysis: Generic GNN

Generic Graph Neural Networks contains three components:

1. **Neighborhood Selection:**
   RandomWalk, DeepWalk, SkipWalk.
2. **Message Passing:**
   Linear, Convolution, Recurrent, Attention
3. **Message Aggregation:**
   Averaging, Max Pooling, MLP, Attention.

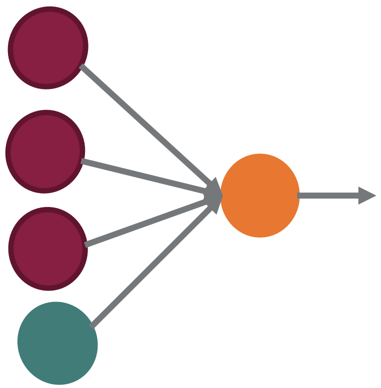


Generic Graph Neural Network*

* Zeng, Yufan, and Jiashan Tang. "Rlc-gnn: An improved deep architecture for spatial-based graph neural network with application to fraud detection." *Applied Sciences* 11.12 (2021): 5656.

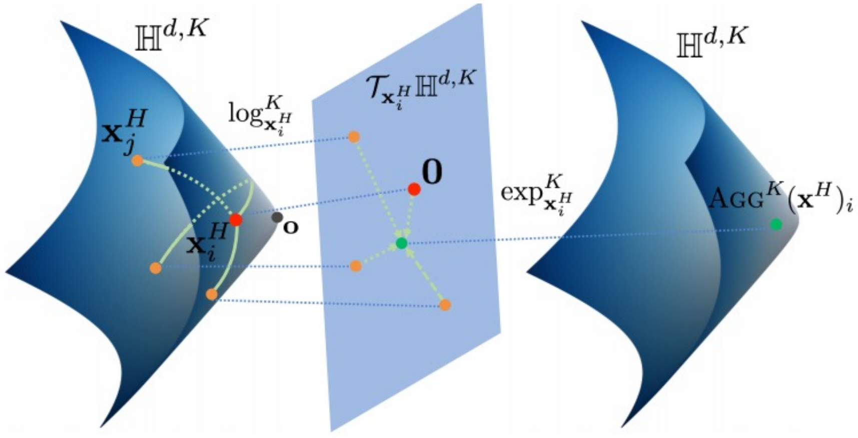
Applications

Graph Analysis: Hyperbolic Models (Previously discussed in Architectures)

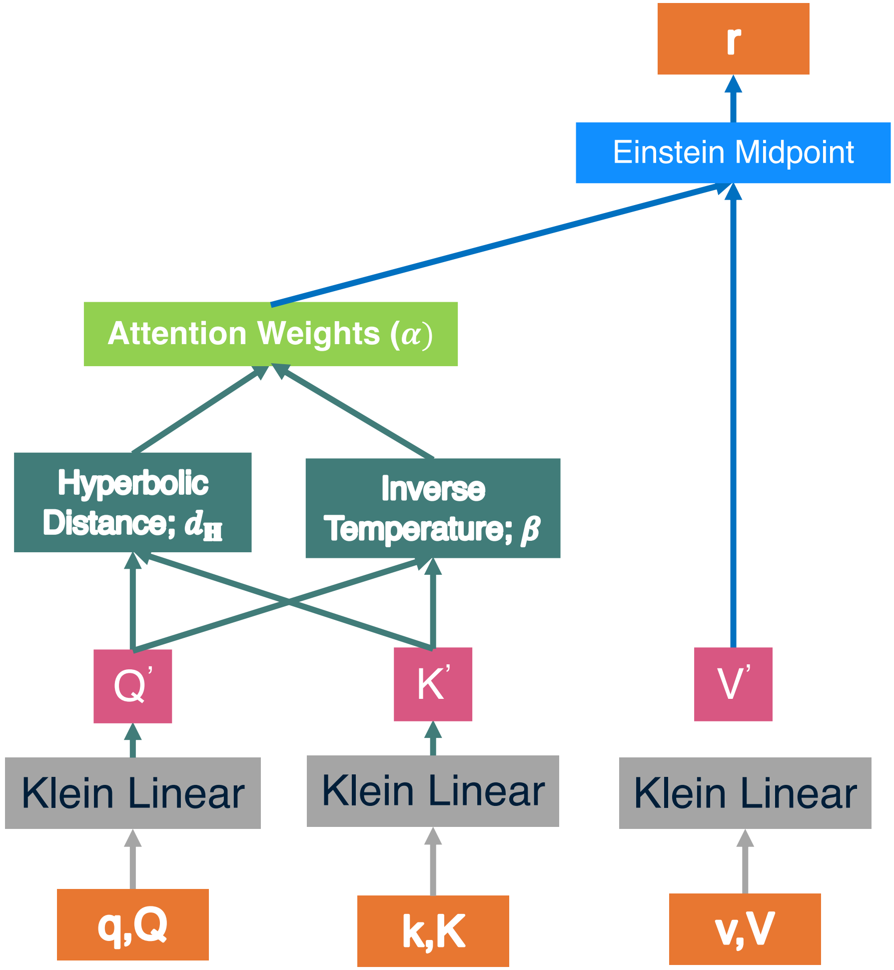


HNN

$w_1 \otimes_c x_1 \oplus_c w_2 \otimes_c x_2 \oplus_c w_3 \otimes_c x_3 \oplus_c \mathrm{b}$



HGCN



HGAT

Applications

Graph Analysis: Experimental Study

Evaluation Tasks:

Node Classification: Given a graph with nodes (attributed or non-attributed) and edges, estimate a model that predicts the class of unclassified existing (transductive) or new (inductive) nodes.

Link Prediction: Given a graph with nodes (attributed or non-attributed) and edges, estimate a model that predicts the probability of edges between existing (transductive) or new (inductive) node-pairs.

Datasets (δ is hyperbolicity, less δ implies more hierarchy in the dataset):

Area	Dataset	δ	Nodes	Edges	Labels	Description
Citation	CORA	11	Papers	Citations	Academic Subareas	Machine learning papers.
	PubMed	3.5	Papers	Citations	Academic Subareas	Medical papers.
Medicine	DISEASE	0	Testee	Interaction	Infected or not	SIR disease propagation model.
	PPI	1	Protein	Interaction	Stem cell growth rate	Union of ppi networks in human tissues.
Traffic	AIRPORT	1	Airports	Flight paths	Country population	Airline routes from OpenFlights.org.

Evaluation Metrics:

Node Classification: F1 score

Link Prediction: ROC-AUC score

Applications

Graph Analysis: Node Classification

Node Classification (F1 Score)										
		Shallow		Neural Nets		Graph Neural Nets				
Dataset	δ	EUC	HYP	MLP	HNN	GCN	GAT	SAGE	HGCN	HGAT
CORA	11	23.8 ± 0.7	22.0 ± 1.5	51.5 ± 1.0	54.6 ± 0.4	81.3 ± 0.3	83.0 ± 0.7	77.9 ± 2.4	79.9 ± 0.2	79.6 ± 0.3
PubMed	3.5	48.2 ± 0.7	68.5 ± 0.3	72.4 ± 0.2	69.8 ± 0.4	78.1 ± 0.2	79.0 ± 0.3	77.4 ± 2.2	80.3 ± 0.3	79.2 ± 0.3
DISEASE	0	32.5 ± 1.1	45.5 ± 3.3	28.8 ± 2.5	41.0 ± 1.8	69.7 ± 0.4	70.4 ± 0.4	69.1 ± 0.6	74.5 ± 0.9	73.4 ± 0.9
PPI	1	-	-	55.3+0.4	59.3 ± 0.4	69.7 ± 0.3	70.5 ± 0.4	69.1 ± 0.3	74.6 ± 0.3	73.6 ± 0.3
AIRPORT	1	60.9 ± 3.4	70.2 ± 0.1	68.6 ± 0.6	80.5 ± 0.5	81.4 ± 0.6	81.5 ± 0.3	82.1 ± 0.5	90.6 ± 0.2	89.4 ± 0.2

Applications

Graph Analysis: Link Prediction

Link Prediction (ROC AUC Score)										
		Shallow		Neural Nets		Graph Neural Nets				
Dataset	δ	EUC	HYP	MLP	HNN	GCN	GAT	SAGE	HGCN	HGAT
CORA	11	82.5 ± 0.3	87.6 ± 0.2	83.1 ± 0.5	89.0 ± 0.1	90.4 ± 0.2	93.7 ± 0.1	85.5 ± 0.6	92.9 ± 0.1	90.8 ± 0.2
PubMed	3.5	83.3 ± 0.1	87.5 ± 0.1	84.1 ± 0.9	94.9 ± 0.1	91.1 ± 0.5	91.2 ± 0.1	86.2 ± 1.0	96.3 ± 0.0	93.9 ± 0.2
DISEASE	0	59.8 ± 2.0	63.5 ± 0.6	72.6 ± 0.6	75.1 ± 0.3	64.7 ± 0.5	69.8 ± 0.3	65.9 ± 0.3	90.8 ± 0.3	88.6 ± 0.3
PPI	1	-	-	67.8 ± 0.2	72.9 ± 0.3	77.0 ± 0.5	76.8 ± 0.4	78.1 ± 0.6	84.5 ± 0.4	81.4 ± 0.3
AIRPORT	1	92.0 ± 0.0	94.5 ± 0.0	89.8 ± 0.5	90.8 ± 0.2	89.3 ± 0.4	90.5 ± 0.3	90.4 ± 0.5	96.4 ± 0.1	94.2 ± 0.2

Applications

Graph Analysis: Learnings

- ➔ Hyperbolic space are consistently better at simultaneously capturing hierarchical structure from more tree-like datasets where the hyperbolicity is lower.

- ➔ Hyperbolic networks are able to perform better on both message aggregation (node classification) and message passing (link prediction)

- ➔ HGCN and HGAT are also able to reduce the error margins compared to their Euclidean counterparts.

Applications

Knowledge Graphs: Introduction

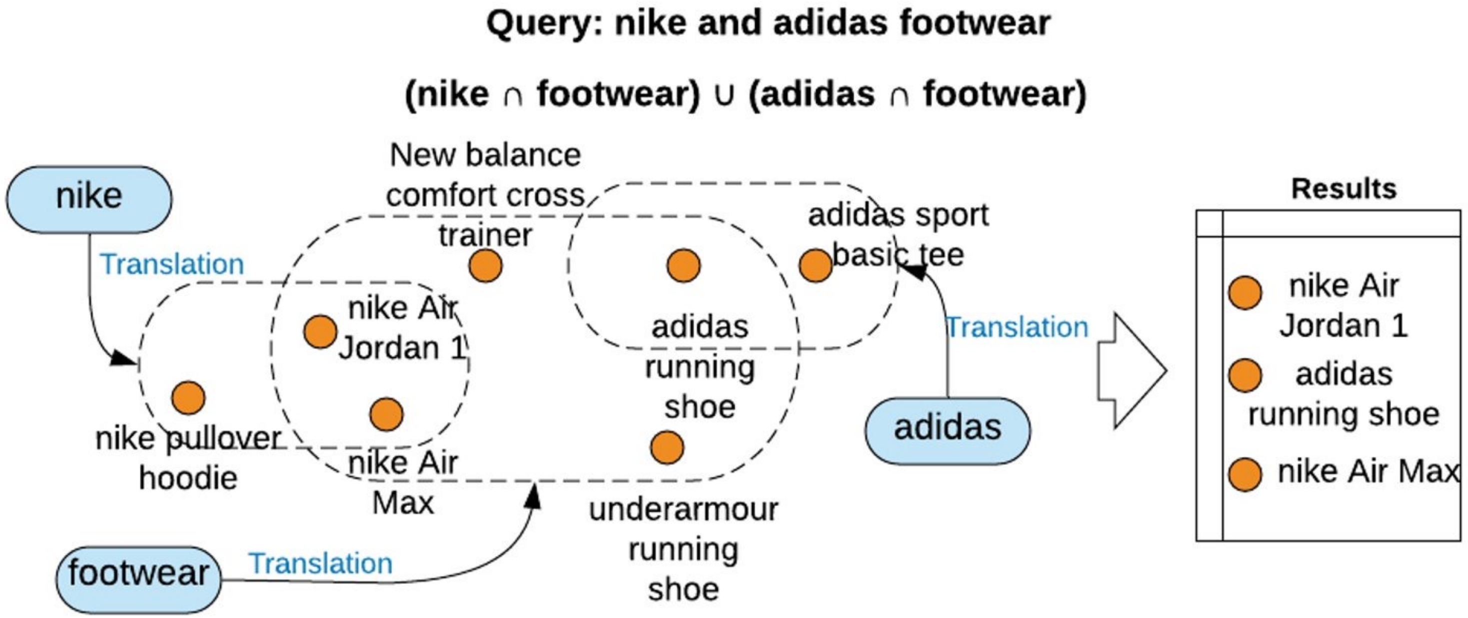
Knowledge Graphs are ubiquitous data structures.

KG querying is computationally expensive due to its size (≈10M nodes with trillions of relations).

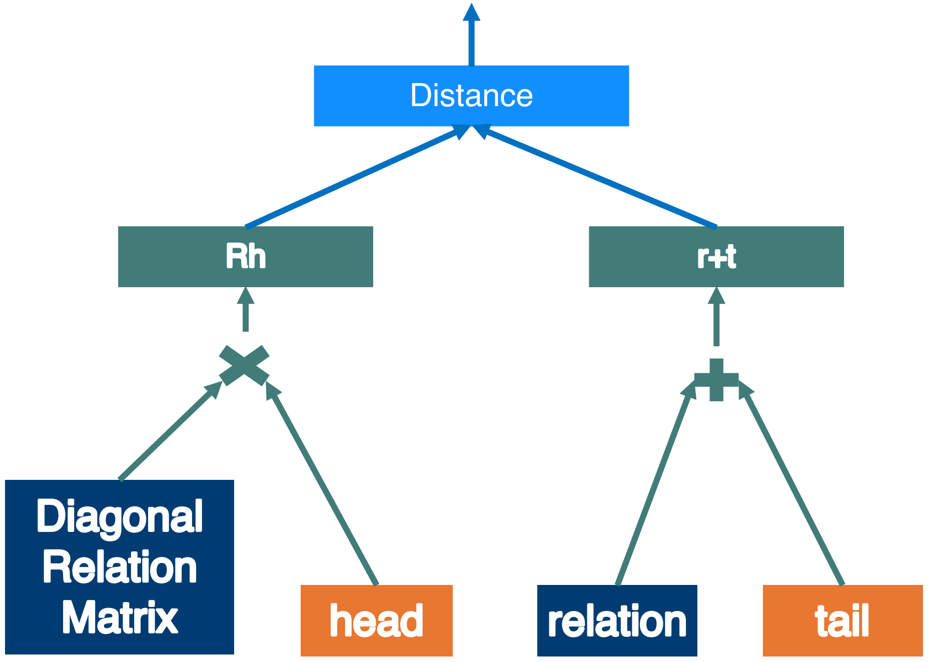
Representation Learning can help!

Learn representations of entities and relations in a latent space.

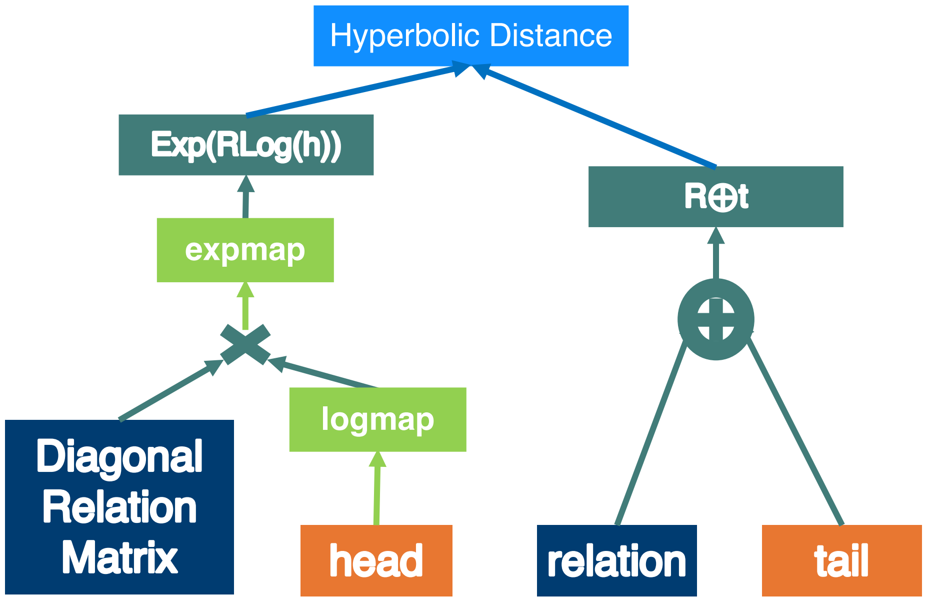
Apply logical operators to simulate querying behaviour.



Multi-relational Poincaré (MuRP) embedding model is an analogue to translational Euclidean representation model (MuRE).



MuRE



MuRP

Applications

MuRP: Experimental Study

Evaluation Tasks:

Relational Link Prediction: Given a graph with attributed nodes and attributed edges, estimate a model that predicts the relevance of other entities as tail answers for an input head entity and relation.

Datasets:

Dataset	# Entity	# Relation	Description
FB15k-237	14,541	237	Collection of real world facts from Freebase network.
WN18RR	40,943	11	Hierarchical collection of relations between words.

Evaluation Metrics:

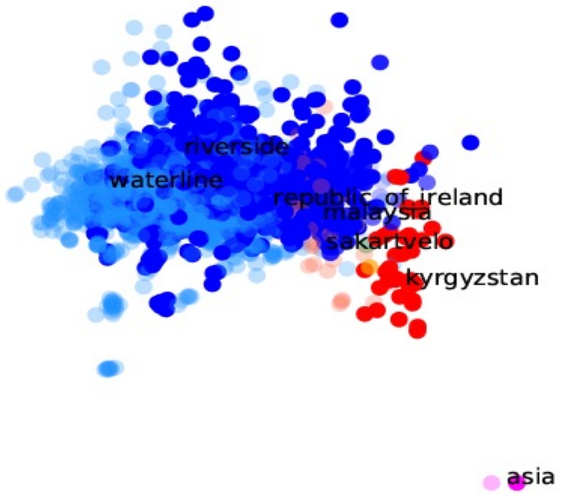
Relational Link Prediction : HITS@1,3,10 and Mean Reciprocal Rank (MRR).

Applications

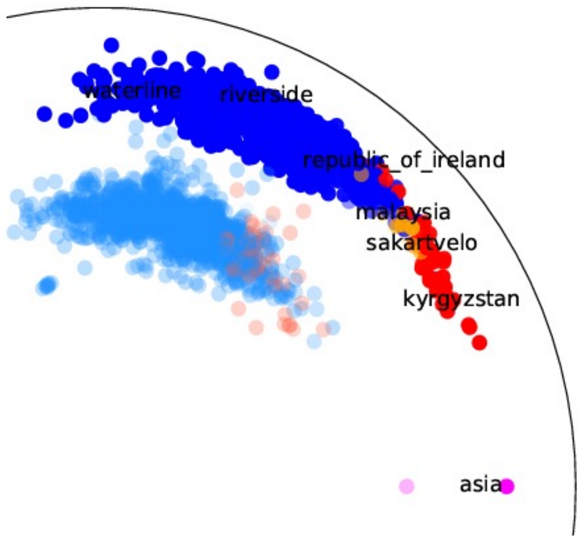
MuRP: Relational Link Prediction

Relational Link Prediction								
Dataset	WN18RR				FB15k-237			
Models	MRR	HITS@10	HITS@3	HITS@1	MRR	HITS@10	HITS@3	HITS@1
TransE	0.226	0.501	-	-	0.294	0.465	-	-
DistMult	0.43	0.49	0.44	0.39	0.241	0.419	0.263	0.155
ComplEx	0.44	0.51	0.46	0.41	0.247	0.428	0.275	0.158
Neural LP	-	-	-	-	0.25	0.408	-	-
MINERVA	-	-	-	-	-	0.456	-	-
ConvE	0.43	0.52	0.44	0.4	0.325	0.501	0.356	0.237
M-Walk	0.437	-	0.445	0.414	-	-	-	-
RotatE	-	-	-	-	0.297	0.48	0.328	0.205
MuRP	0.481	566	0.495	0.44	0.335	0.518	0.367	0.243

Applications
MuRP: Visualization



MuRE



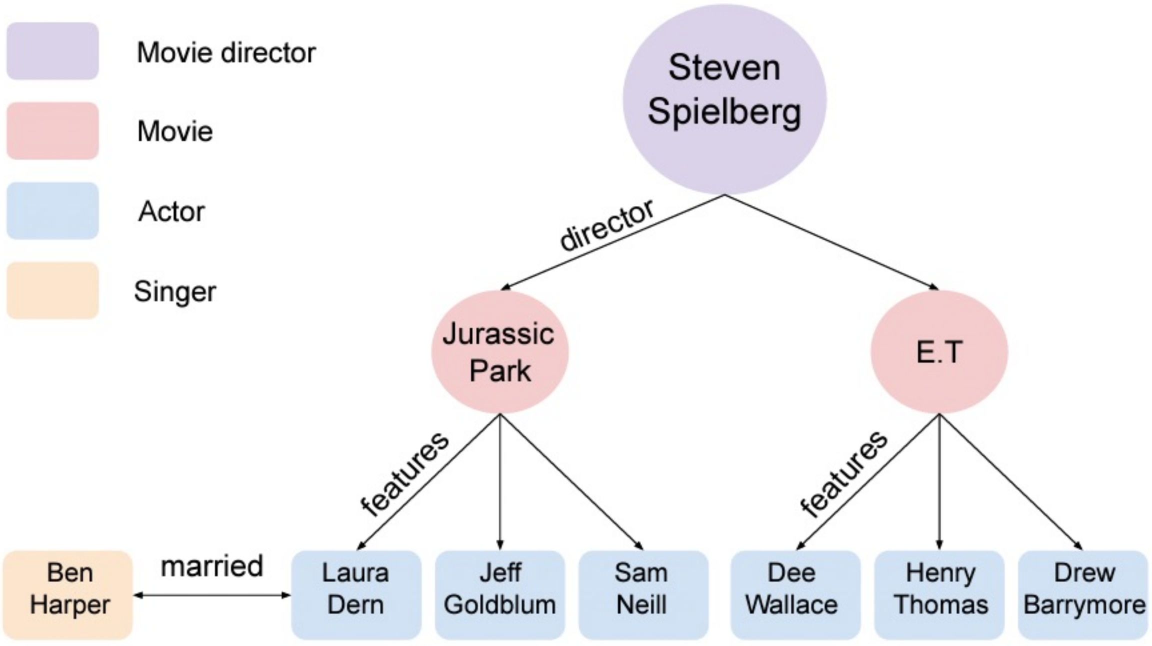
MuRP

Learned 40-dimensional MuRP and MuRE embeddings for WN18RR relation has_part.
Blue indicates **true positives** and **red** indicates **true negatives**. The lighter shades of the color indicate the entity before the relational translation.
Notice that the points are more separable in the Poincaré space when compared to the Euclidean space, where they are clustered together.

Applications

Knowledge Graphs: Rotation, Reflection and Attention

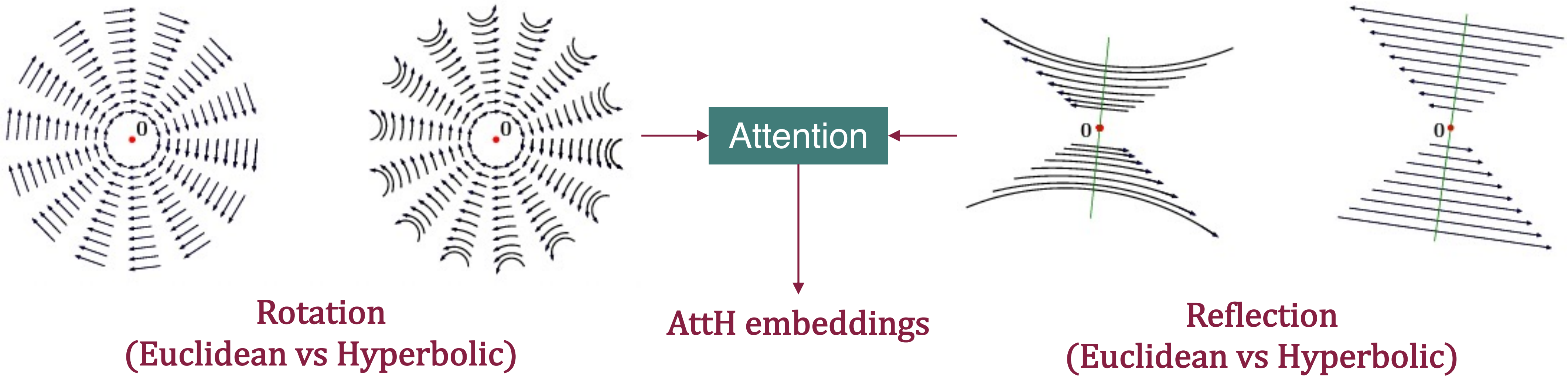
Knowledge Graphs have mixed topology, i.e., some relations are symmetric (*married*) while others are not (*director, features*).



Applications

Knowledge Graphs: Rotation, Reflection and Attention

Hence, the method proposes two sets of embeddings; rotational (RotH) and reflectional (RefH), which are finally combined with Attention (AttH).



Applications

RotH,RefH, AttH : Experimental Study

Evaluation Tasks:

Relational Link Prediction: Given a graph with attributed nodes and attributed edges, estimate a model that predicts the relevance of other entities as tail answers for an input head entity and relation.

Datasets:

Dataset	# Entity	# Relation	Description
FB15k-237	14,541	237	Collection of real world facts from Freebase network.
WN18RR	40,943	11	Hierarchical collection of relations between words.
YAGO3-10	123,182	37	Subset of YAGO3, a large semantic knowledge base, derived from Wikipedia, WordNet, WikiData, GeoNames, and other data sources.

Evaluation Metrics:

Relational Link Prediction : HITS@1,3,10 and Mean Reciprocal Rank (MRR).

Applications

RotH,RefH, AttH: Relational Link Prediction

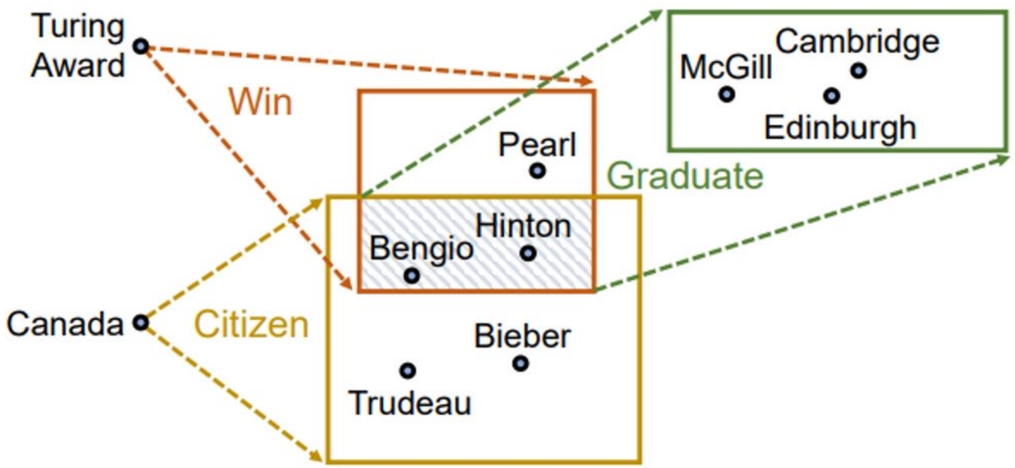
Relational Link Prediction													
	Dataset	WN18RR				FB15k-237				YAGO3-10			
Space	Models	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1
Euclidean	RotatE	0.387	0.33	0.417	0.491	0.29	0.208	0.316	0.458	-	-	-	-
	MuRE	0.458	0.421	0.471	0.525	0.313	0.226	0.34	0.489	0.283	0.187	0.317	0.478
	RefE	0.42	0.39	0.42	0.46	0.294	0.211	0.322	0.463	0.336	0.259	0.367	0.484
	RotE	0.465	0.42	0.484	0.544	0.323	0.235	0.353	0.501	0.23	0.15	0.247	0.392
	AttE	0.455	0.419	0.47	0.521	0.302	0.216	0.33	0.474	0.37	0.289	0.403	0.527
Hyperbolic	MuRP	0.463	0.426	0.477	0.529	0.307	0.22	0.337	0.482	0.381	0.295	0.417	0.548
	RefH	0.456	0.419	0.471	0.526	0.311	0.223	0.339	0.488	0.374	0.29	0.41	0.537
	RotH	0.447	0.408	0.464	0.518	0.312	0.224	0.342	0.489	0.381	0.302	0.415	0.53
	AttH	0.472	0.428	0.49	0.553	0.314	0.223	0.346	0.497	0.393	0.307	0.435	0.559

Applications

Knowledge Graphs: Euclidean Methods

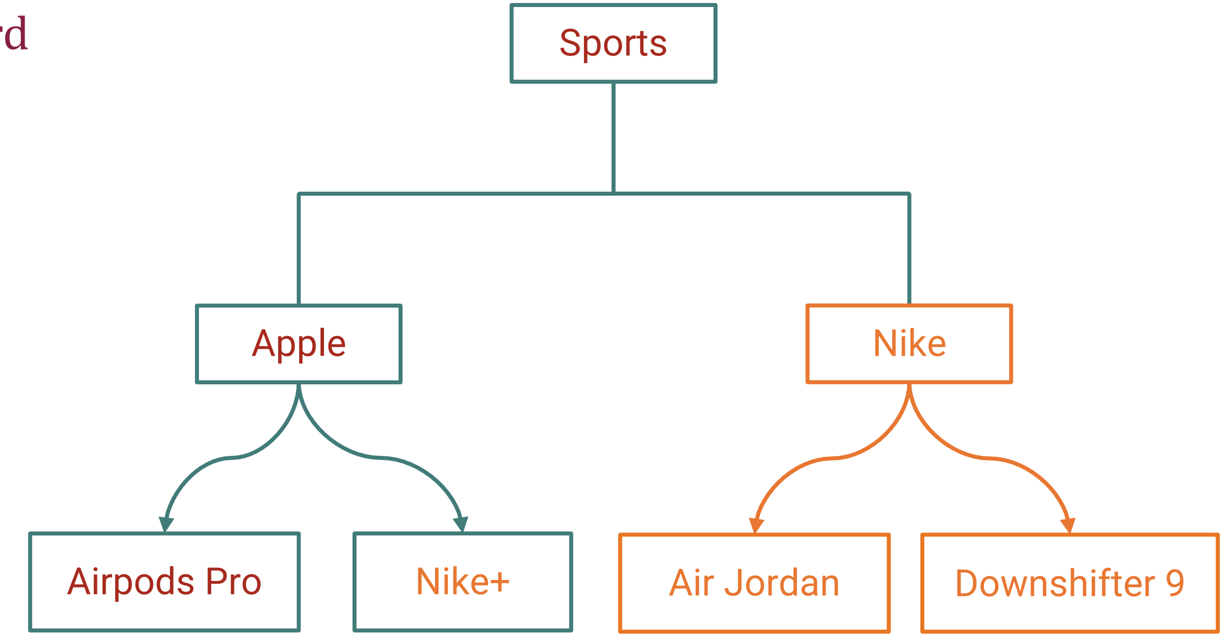
Spatial Representations are better for modelling Knowledge Graphs.

From which universities did the Canadian Turing Award winners graduate?



Box Embeddings in Euclidean Space

Knowledge Graphs possess inherent hierarchy.



Hierarchical Product Graph

Applications

Knowledge Graphs: Hyperboloid Embeddings

Capture spatial features of entities and relations.	Capture hierarchical features of entities and relations.
Handle different query types simultaneously	No labelled data (self-supervised models)

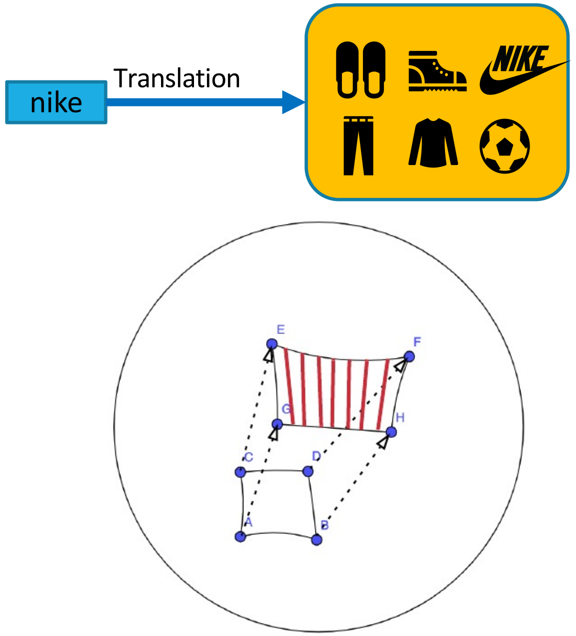
HypE

Applications

Knowledge Graphs: Reasoning Operations

Translation: Gives all children of a query

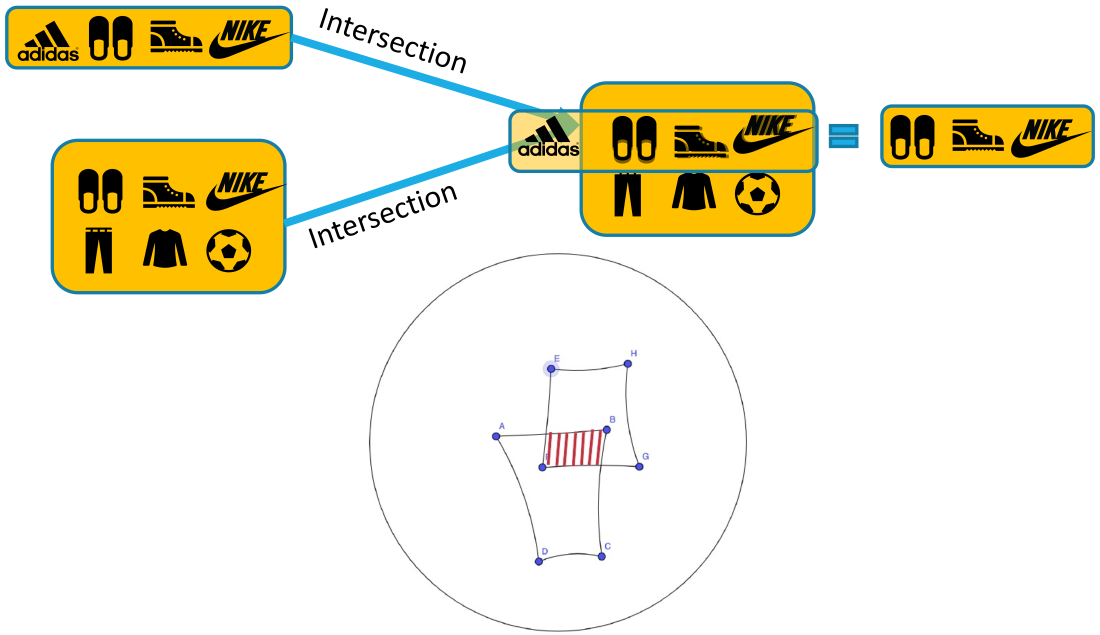
Q: nike



Hyperboloid Translation

Intersection: Gives intersection for two queries.

Q: nike footwear = nike ∩ footwear



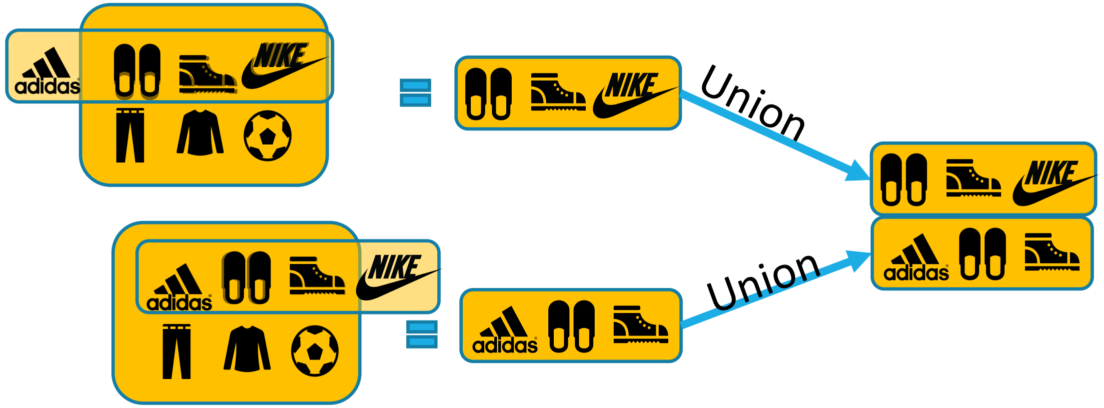
Hyperboloid Intersection

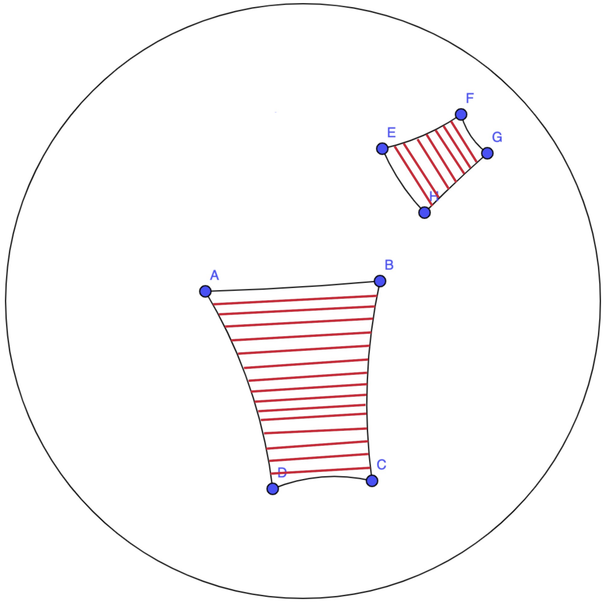
Applications

Knowledge Graphs: Hyperboloid Embeddings

Union Queries: Gives union of two queries.

Q: (nike ∪ adidas) ∩ footwear = (nike ∩ footwear) ∪ (adidas ∩ footwear)





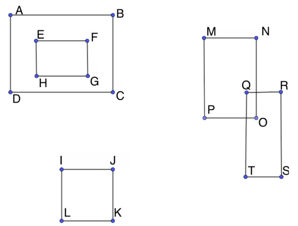
Hyperboloid Union

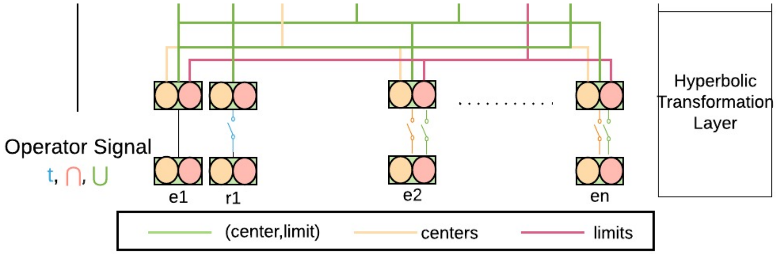
Applications

Knowledge Graphs: Hyperboloid Embeddings

Initialize

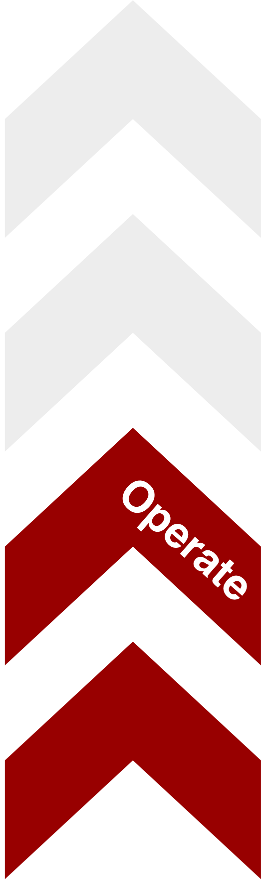
Initialize entities and relations with Random Euclidean Boxes



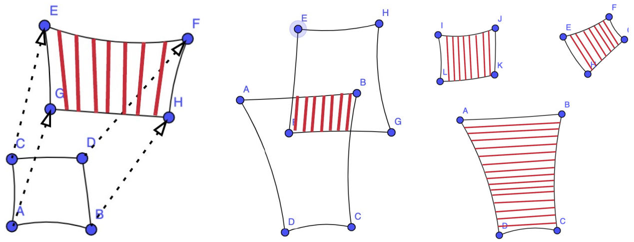


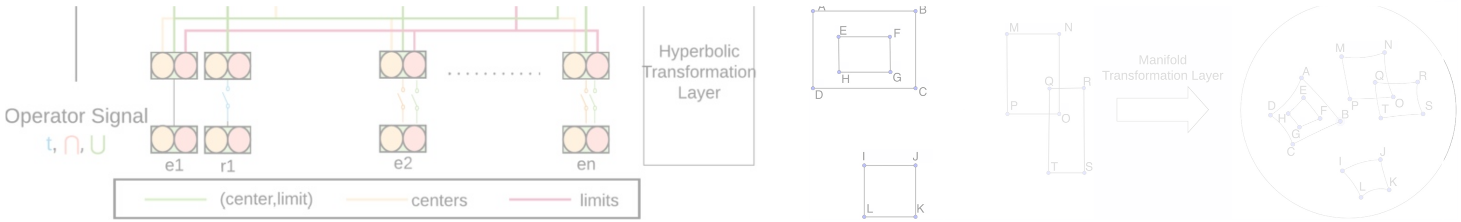
Applications

Knowledge Graphs: Hyperboloid Embeddings



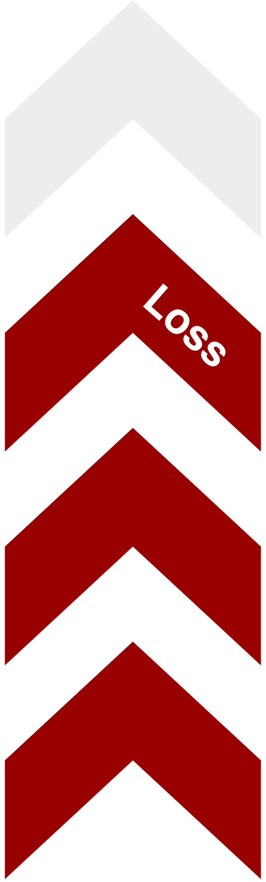
Operate according to signal



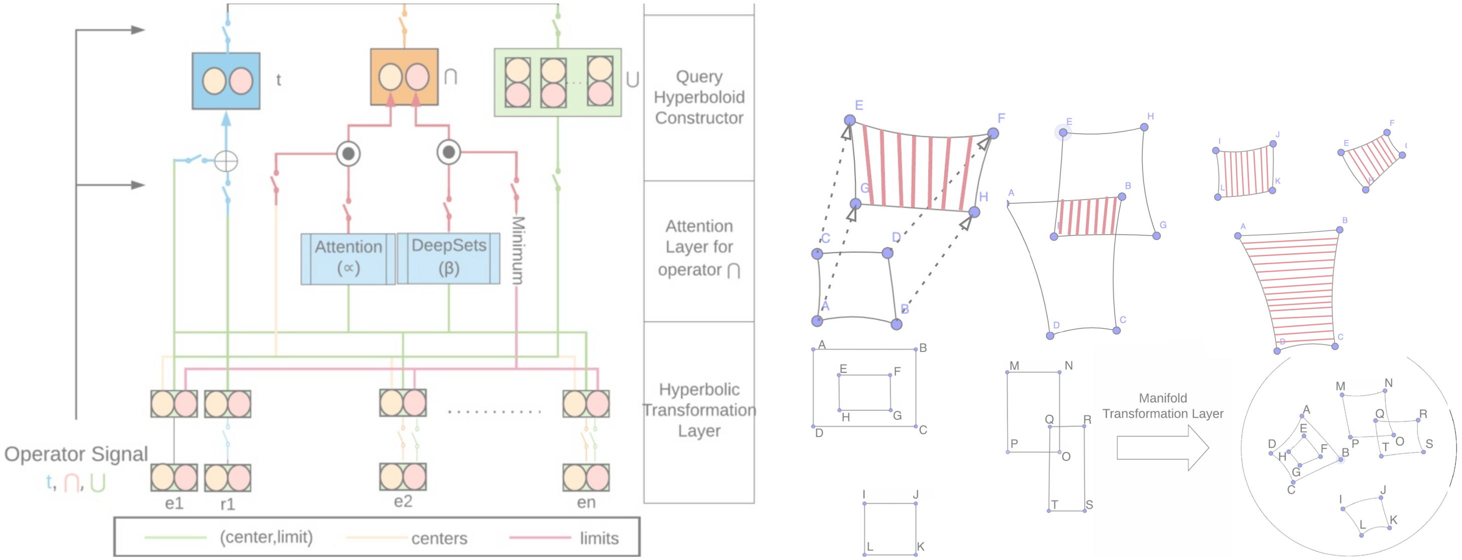


Applications

Knowledge Graphs: Hyperboloid Embeddings

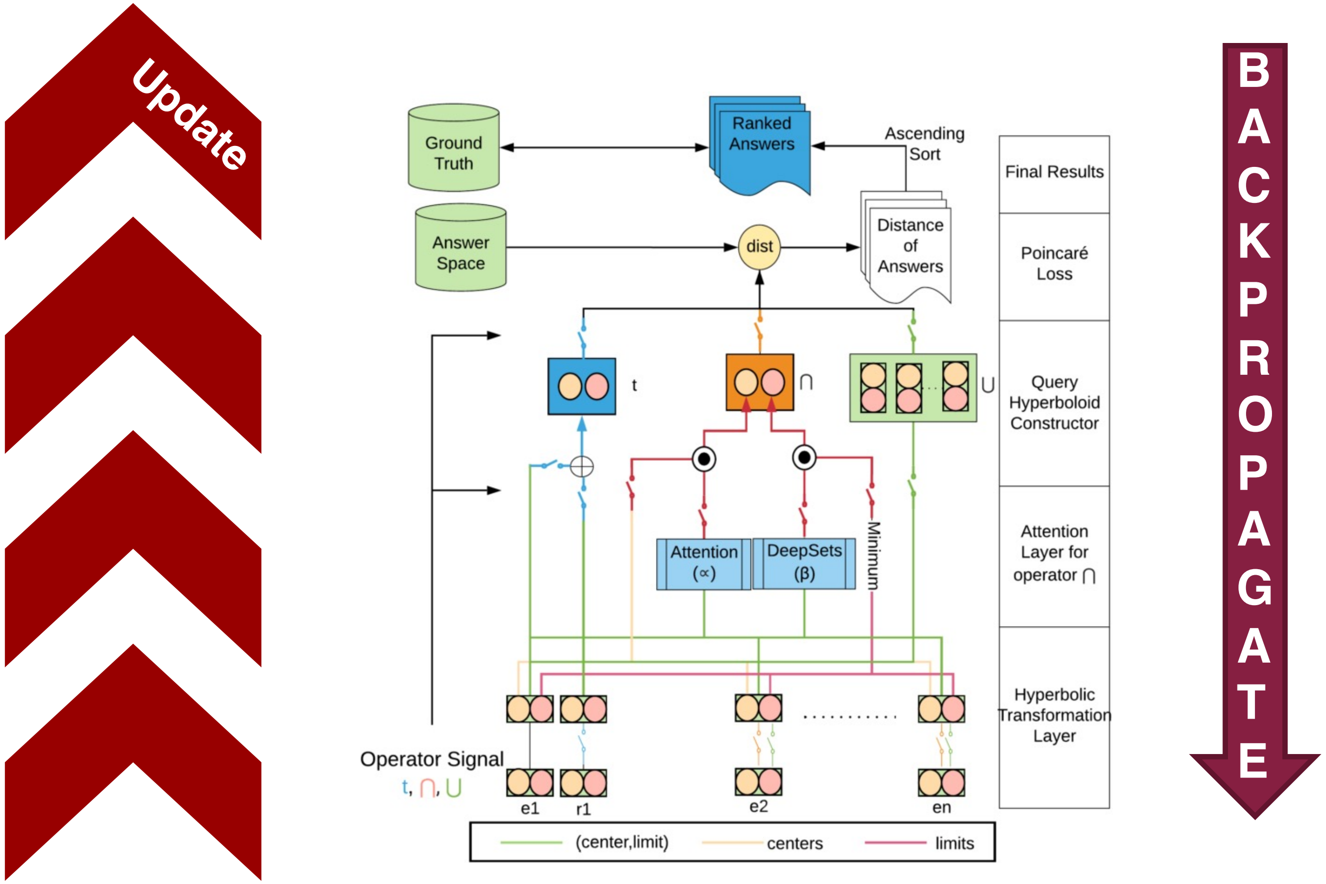


Calculate Loss based on distance of samples from the query space.



Applications

Knowledge Graphs: Hyperboloid Embeddings

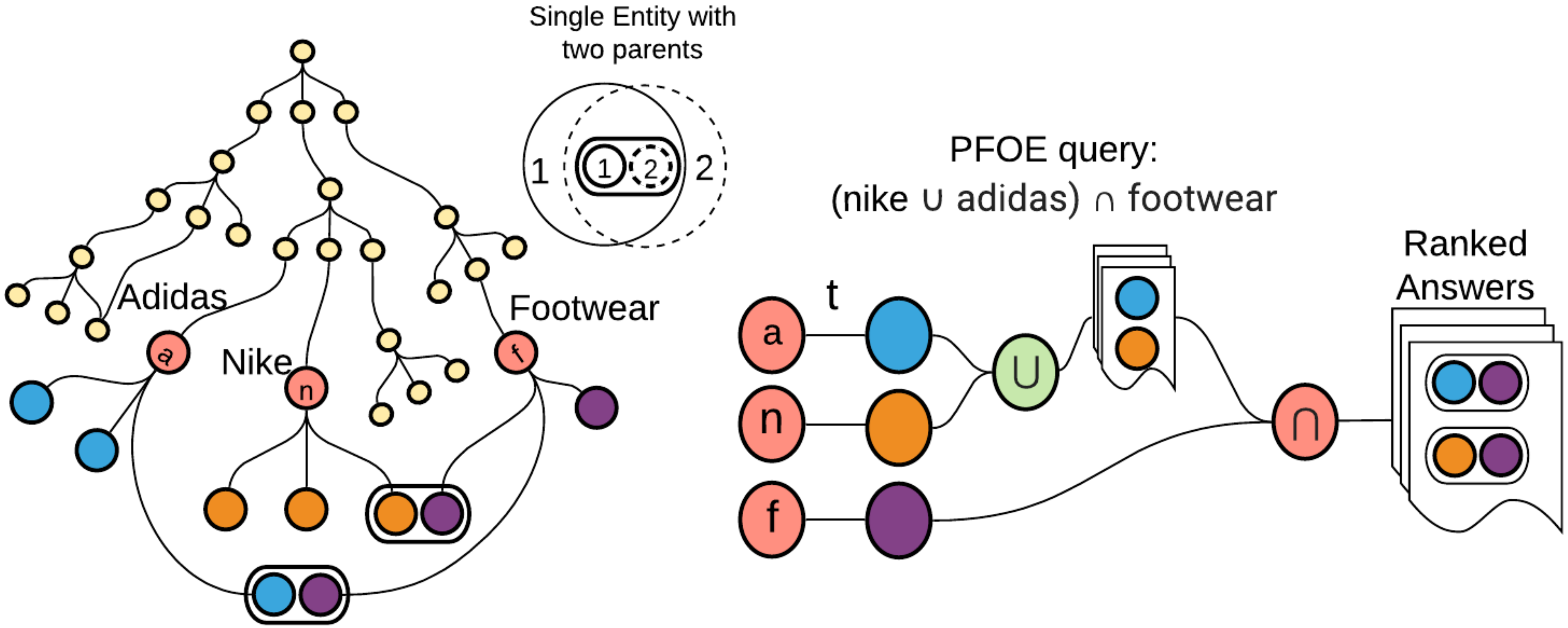


Update entity and relation representations based on the loss.

Applications

Knowledge Graphs: Self-supervised Learning

Generate pseudo-queries by using the structure/relations in the training graph.



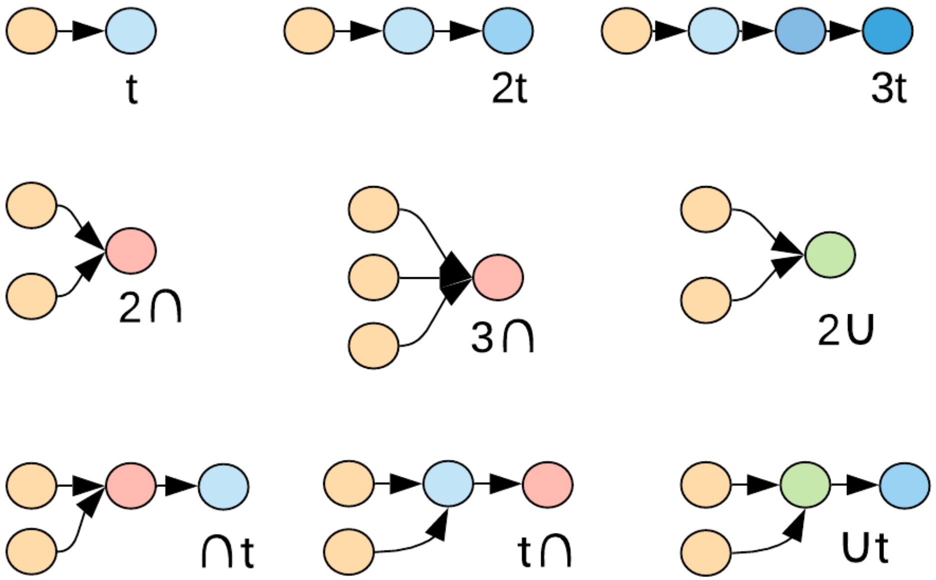
Generation of Pseudo-query from a Product Graph

Applications
Knowledge Graphs: Evaluation

1. Efficacy of the Query Search Space
2. Performance of Anomaly Detection
3. Visualization

Applications

Knowledge Graphs: Efficacy of the Query Space



Translation (t):

1t: "nike", "shoes", "adidas"

2t: "women shoes" ("shoes" → "women")

3t: "furniture" ("furniture"→"chair","table","dining", etc→"ikea","wayfair", etc)

Intersection (∩):

2∩: "nike shoes" ("nike" AND "shoes")

3∩: "nike jordan laces" ("nike" AND "jordan" AND "laces")

∩t: "nike shoes" ("nike" AND "shoes" →products in the space)

t∩: "furniture ikea" ("furniture"→"chair","table","dining", etc AND "ikea")

Union (∪):

2∪: "nike and adidas" ("nike" OR "adidas")

∪t: "nike and adidas shoes" ("nike" OR "adidas" →products in the space)

Applications

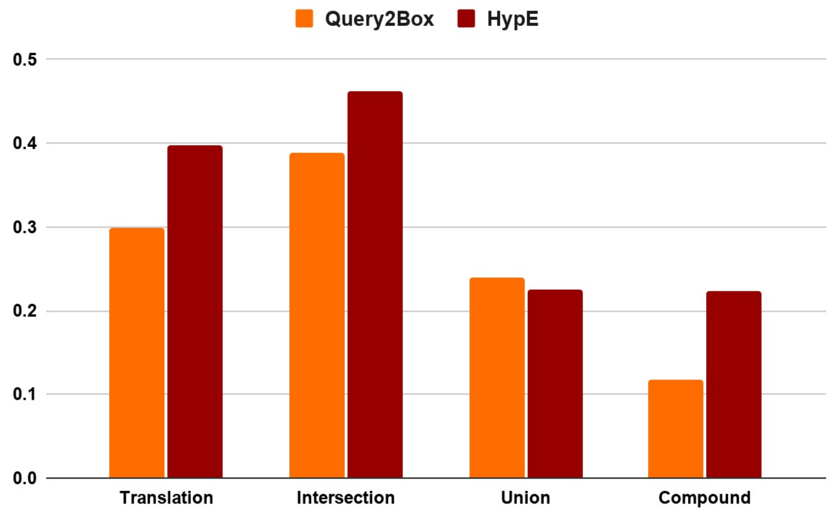
Knowledge Graphs: Efficacy of the Query Space

- Logical Query Reasoning
  - Dataset: FB15K, FB15K-237, NELL995, DBPedia, E-commerce Product Graph
  - Primary Baseline: Query2Box (ICLR 2020)
  - Evaluation Metrics: HITS@3, Mean Reciprocal Rank

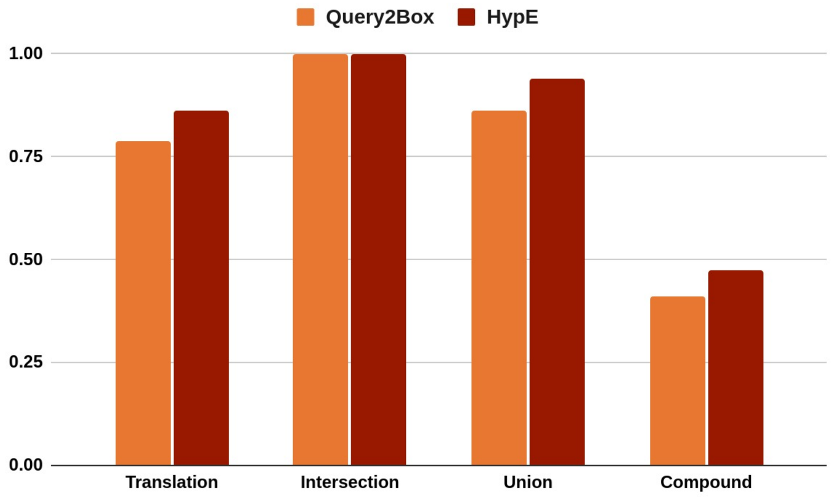
Applications

Knowledge Graphs: Efficacy of the Query Space

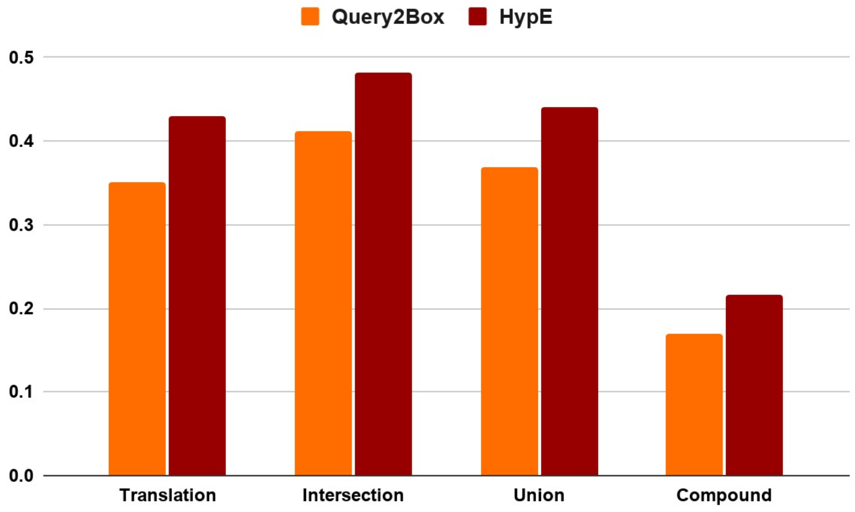
FB15K-237



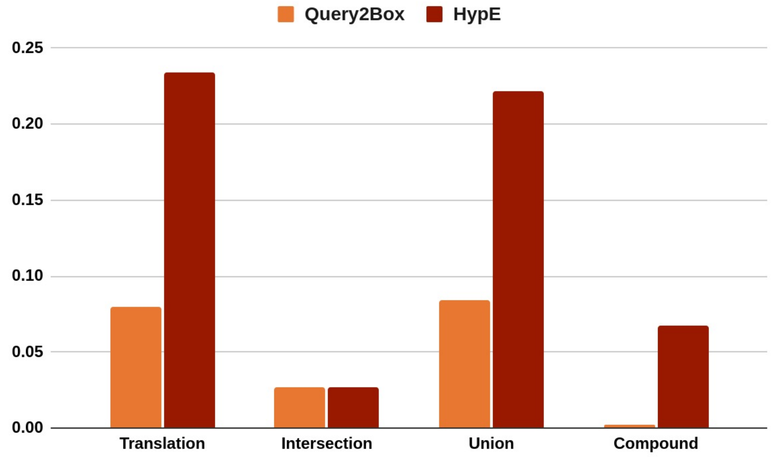
DBPedia



NELL995

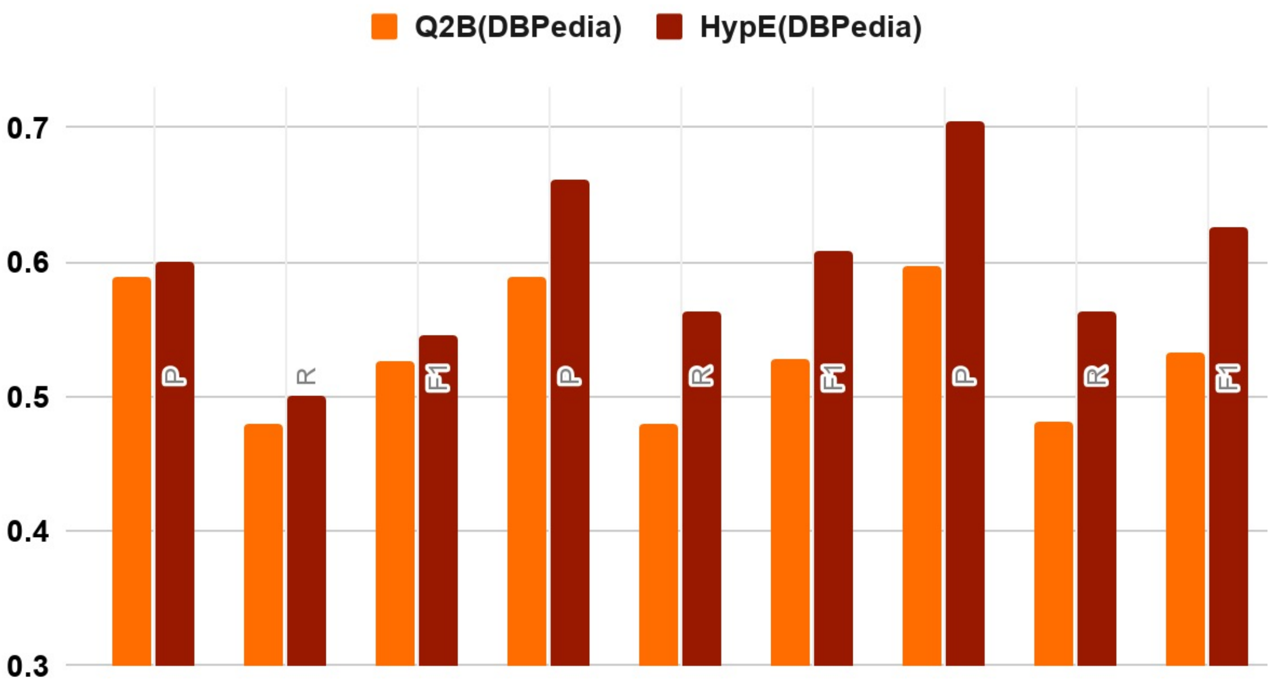


E-commerce (Relative)



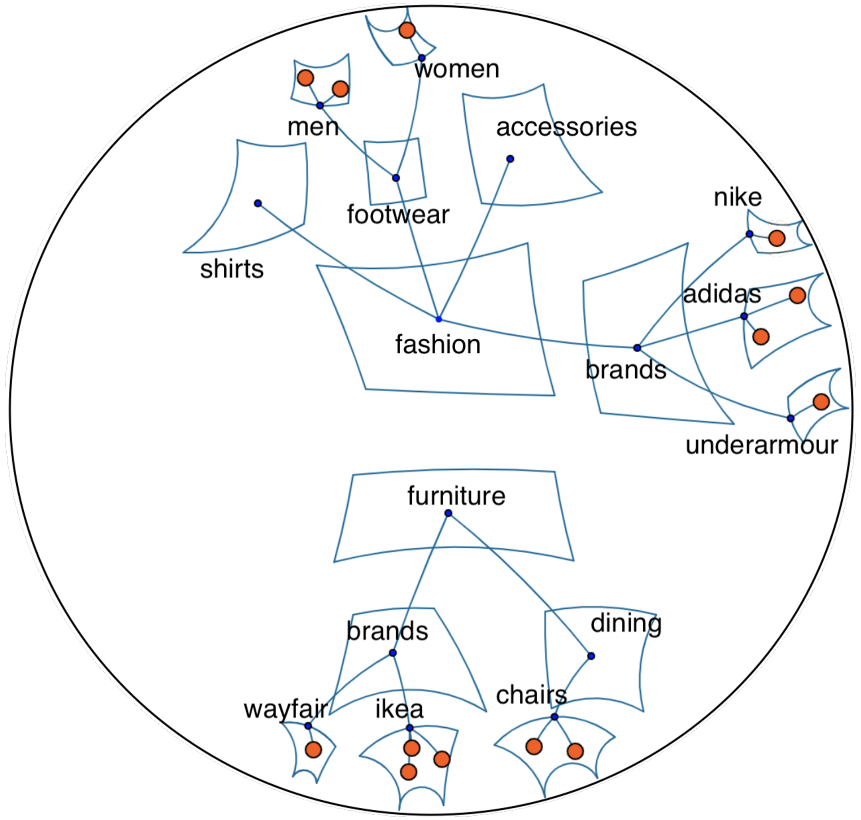
Applications

Knowledge Graphs: Anomaly Detection (Downstream task)

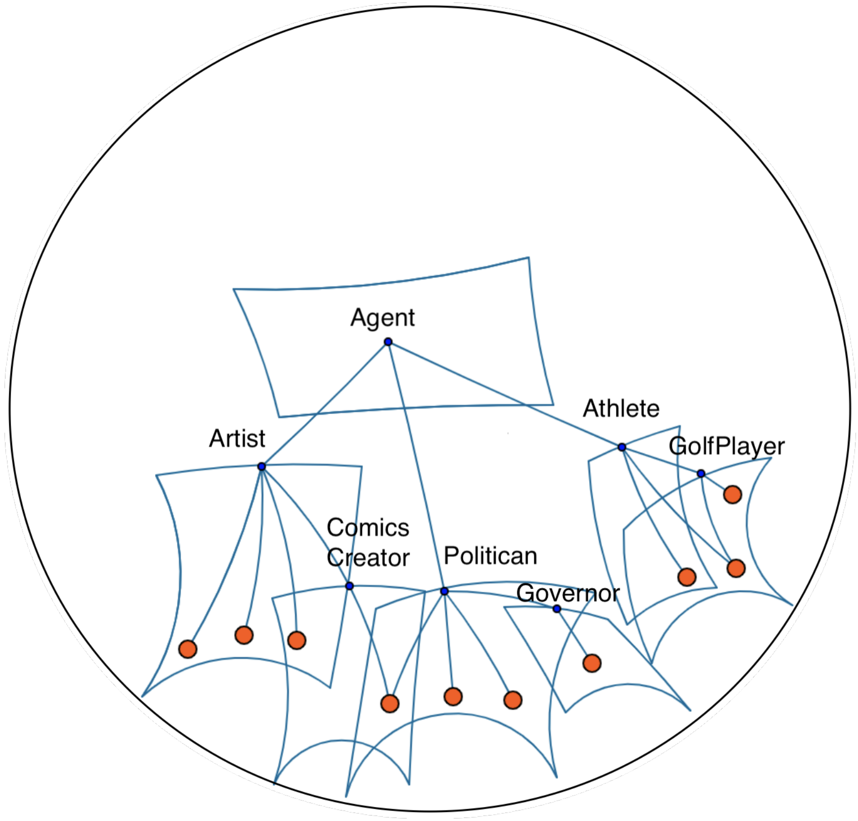


- Miscategorized Nodes detection
- Evaluation Metrics: Precision, Recall, F1-score
- Different levels of parents considered: Parent, Grandparent and Great-grandparent to the child nodes.

Applications
Knowledge Graphs: Visualization



Hyperboloids of the E-commerce Graph



Hyperboloids of DBPedia

More results can be found in the paper: Choudhary, N., Rao, N., Katariya, S., Subbian, K., & Reddy, C. K. (2021, April). Self-Supervised Hyperboloid Representations from Logical Queries over Knowledge Graphs. (WWW 2021)

Applications

Knowledge Graphs: Learnings

- ➔ Hyperbolic space is better at simultaneously capturing spatial and hierarchical structure information by pseudo-querying the knowledge graphs.
- ➔ The ablation study shows the clear importance of using relatively complex queries such as intersection and union in enhancing HypE's performance.
- ➔ HypE's representation, in congruence with/without additional information such as semantics, can also be used for downstream tasks (anomaly detection).
- ➔ The hyperboloids can also be visualized in a Poincaré ball for better human comprehension.

Applications

Extending Logical Reasoning to Product Search

Challenges:

- Search queries are presented in Natural Language and not logical queries.
- Reasoning needs logical steps.
- The products lie in a hierarchical catalogue.

Can we learn the logical reasoning steps from the natural language queries to retrieve products from a hierarchical catalogue?

Applications

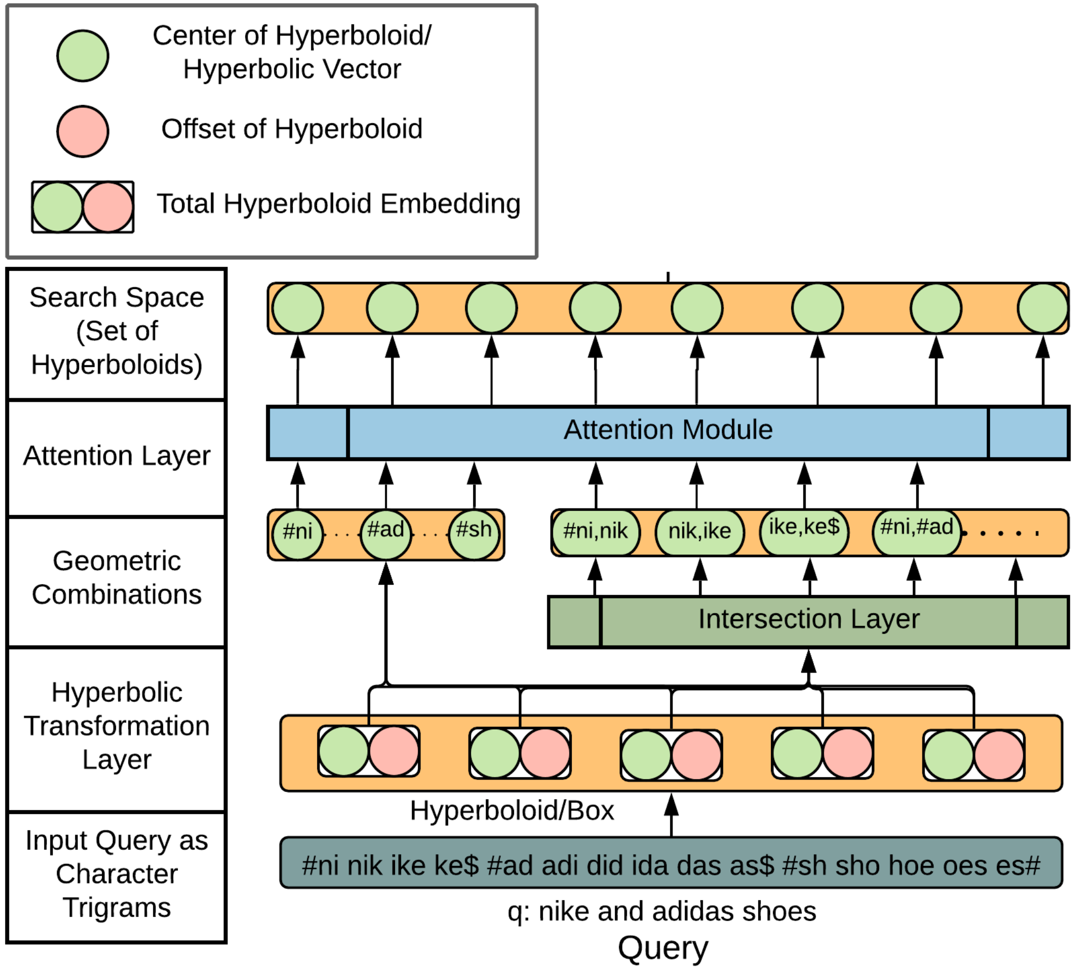
Product Search: ANTHEM

ANTHEM

- Learn query representations in the Euclidean space.
- Learn query operations through Attentive aggregation.
- Encode hierarchically catalogued products in hyperbolic space.
- Use a joint-learning architecture to match queries and products.

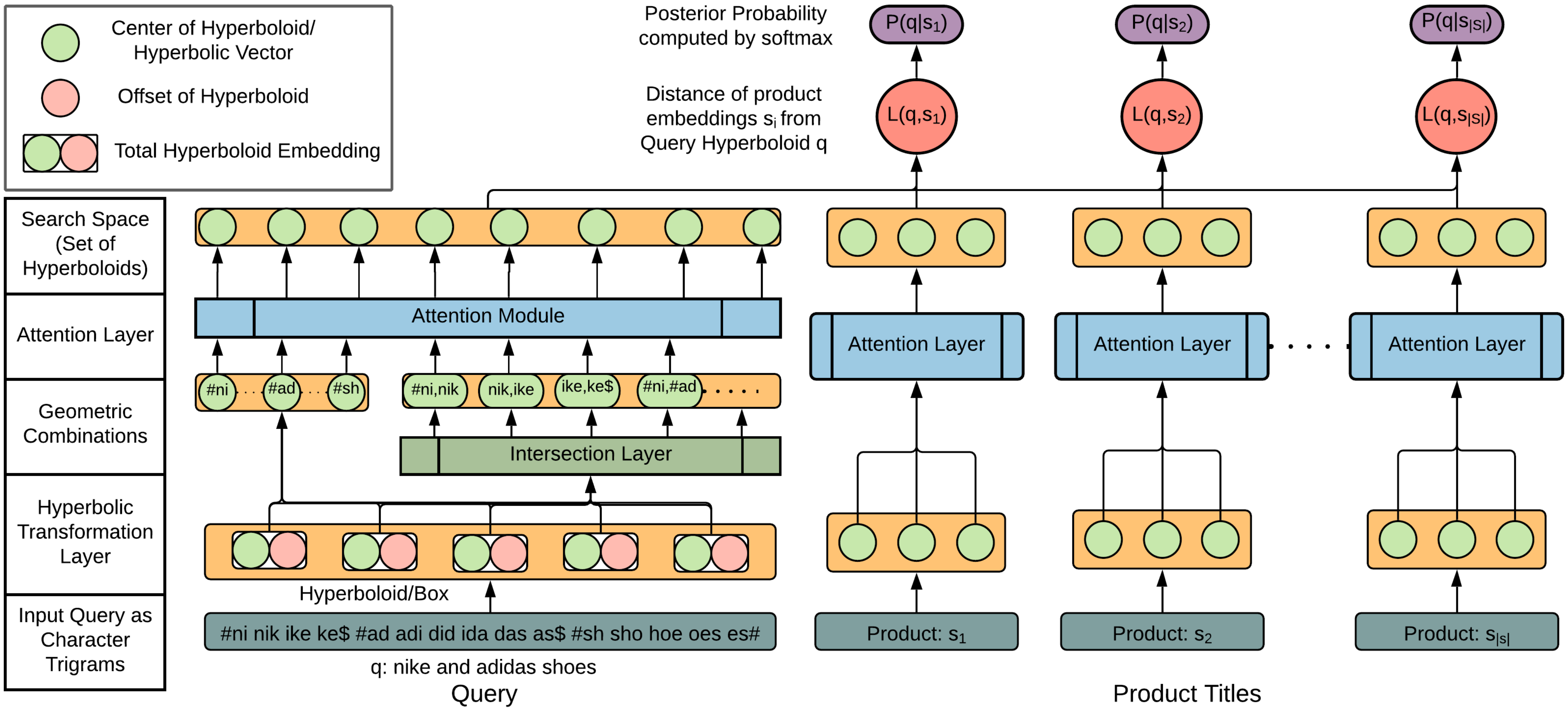
Applications

Product Search: ANTHEM



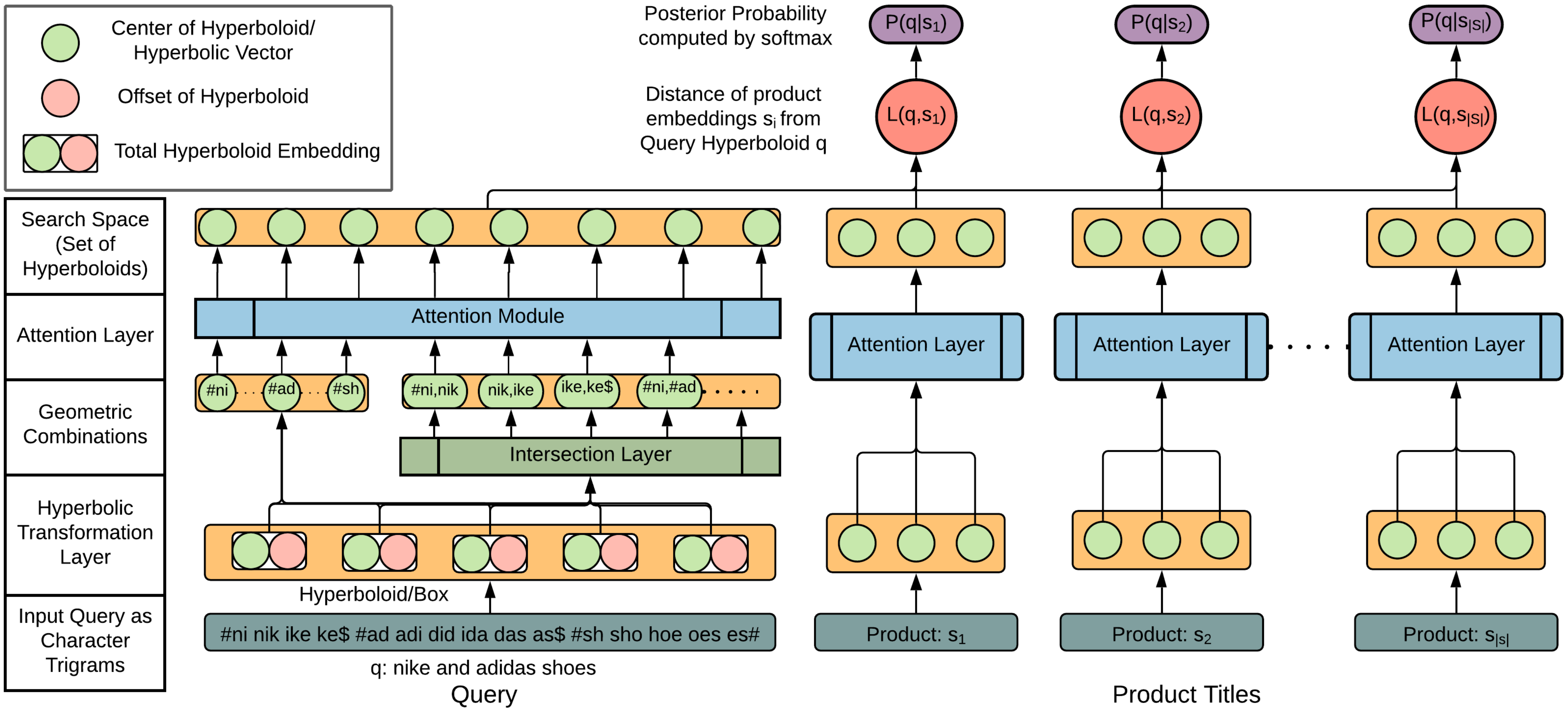
Applications

Product Search: ANTHEM



Applications

Product Search: ANTHEM



Applications

Product Search: Evaluation

1. Performance on Product Search
2. Performance on Query Matching
3. Explainability Study

Applications

Product Search: Performance on Product Search

- Dataset for Product Search: E-commerce (Amazon) Product Search, Public E-commerce Search Relevance
- Dataset for Query Matching: E-commerce (Amazon) ESCI Query Matching
- Baselines: ARC-II, KNRM, DUET, DRMM, aNMM, MatchPyramid, C-DSSM, MV-LSTM, BERT.
- Evaluation Metrics: NDCG@3, NDCG@5, NDCG@10, Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR)

Applications

Product Search: Performance on Product Search

Datasets	E-commerce Product Search (in %)					Public E-commerce Search Relevance (in %)				
Models	NDCG@3	NDCG@5	NDCG@10	MAP	MRR	NDCG@3	NDCG@5	NDCG@10	MAP	MRR
ARC-II	0	0	0	0	0	59.2	58.1	54.4	58.2	48.5
KNRM	12.5	12.8	15.0	12.8	16.7	66.6	65.5	62.6	65.6	56.6
Duet	13.1	13.3	15.4	13.4	14.7	66.9	65.8	62.8	66.0	55.6
DRMM	20.5	22.8	24.4	20.3	21.7	71.3	71.3	67.7	70.0	59.0
aNMM	21.0	23.9	26.8	20.3	22.9	71.6	72.0	69.0	70.0	59.6
MatchPyramid	25.6	25.6	26.8	25.0	34.7	74.3	72.9	69.0	72.8	65.3
C-DSSM	31.3	29.4	44.7	32.6	27.8	77.7	75.2	78.7	77.1	61.9
MV-LSTM	34.7	33.3	55.7	34.3	37.7	79.7	77.5	84.7	78.2	66.8
BERT	38.6	37.2	65.9	40.1	51.0	82.1	79.7	90.2	81.5	73.2
E-ANTHEM	49.4	46.7	66.7	51.2	62.9	88.5	85.2	90.7	88.0	79.0
ANTHEM	**51.1**	**48.9**	**80.9**	**53.5**	**65.4**	**89.5**	**86.5**	**98.4**	**89.3**	**80.2**

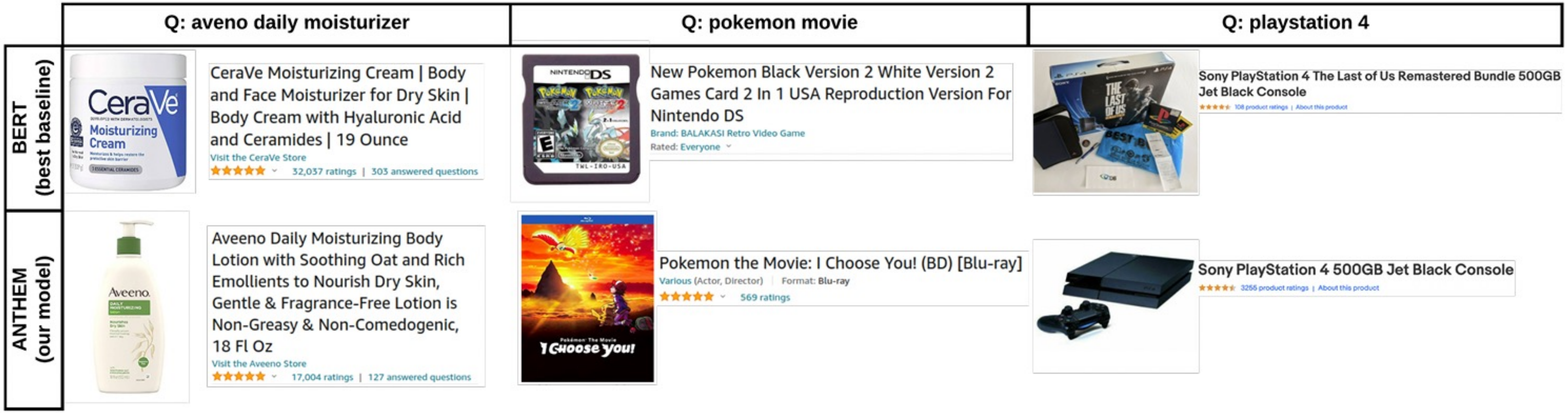
Applications

Product Search: Performance on Query Matching

Models	Accuracy (in %)	F-score (in %)	AUC (in %)
ARC-II	0.0	0.0	0.0
KNRM	-4.1	-24.9	-19.1
Duet	-2.3	-4.7	0.9
DRMM	25.1	15.4	33.1
aNMM	-1.3	-8.6	4.0
MatchPyramid	-14.5	-17.8	-9.7
C-DSSM	21.2	21.7	30.1
MV-LSTM	41.1	21.2	48.9
BERT	40.1	33.5	54.7
E-ANTHEM	43.2	40.3	61.4
ANTHEM	**43.9**	**40.8**	**62.6**

Applications

Product Search: Explainability Study



ANTHEM detects misspellings due to the use of char trigrams.

ANTHEM gives less false positives due to hierarchical information.

ANTHEM results have higher diversity due to information gain from different product hierarchies.

More results can be found in the paper: Choudhary, N., Rao, N., Katariya, S., Subbian, K., & Reddy, C. K. (2022, February). ANTHEM: Attentive Hyperbolic Entity Model for Product Search. (WSDM 2022)

Applications

Product Search: Learnings

- ➔ Hyperbolic space is better at simultaneously capturing spatial and hierarchical structure information by pseudo-querying the knowledge graphs.
- ➔ The ablation study shows the clear importance of using attention mechanism to capture intersection and union operations in product search queries.
- ➔ HypE's product representation, in congruence with matching architecture can be utilized for downstream tasks (product search and query matching).
- ➔ The attention weights in ANTHEM can also be visualized for better human comprehension.

Applications

Natural Language Processing: Text Classification

- Natural Language Inference: Given two sentences, a premise (e.g. "Little kids A. and B. are playing soccer.") and a hypothesis (e.g. "Two children are playing outdoors."), the binary classification task is to predict whether the second sentence can be inferred from the first one.
  - Standard SNLI dataset; 570K training, 10K validation and 10K test sentence pairs.
- PREFIX: Given two sentences, model has to decide if the second sentence is a noisy prefix of the first, or a random sentence.
  - PREFIX-Z% (Z=10, 30 or 50): For each random first sentence and one random prefix of it, a second positive sentence is generated by randomly replacing Z% of the words of the prefix, and a second negative sentence of same length is randomly generated. 500K training, 10K validation and 10K test pairs.

Applications

Text Classification: Evaluation

Metric: Accuracy	SNLI	PREFIX-10%	PREFIX-30%	PREFIX-50%
FULLY EUCLIDEAN RNN	79.34	89.62	81.71	72.10
HYPERBOLIC RNN+FFNN, EUCL. MLR	79.18	96.36	87.83	76.50
FULLY HYPERBOLIC RNN	78.21	96.91	87.25	62.94
FULLY EUCLIDEAN GRU	**81.52**	95.96	86.47	75.04
HYPERBOLIC GRU+FFNN, EUCL. MLR	79.76	**97.36**	**88.47**	**76.87**
FULLY HYPERBOLIC GRU	81.19	97.14	88.26	76.44

Applications

Text Classification: Evaluation

- Hyperbolic architectures, on average, perform better than the Euclidean variants.
- The improvement is better on hierarchical task; PREFIX
- No apparent benefit of using Hyperbolic MLR.

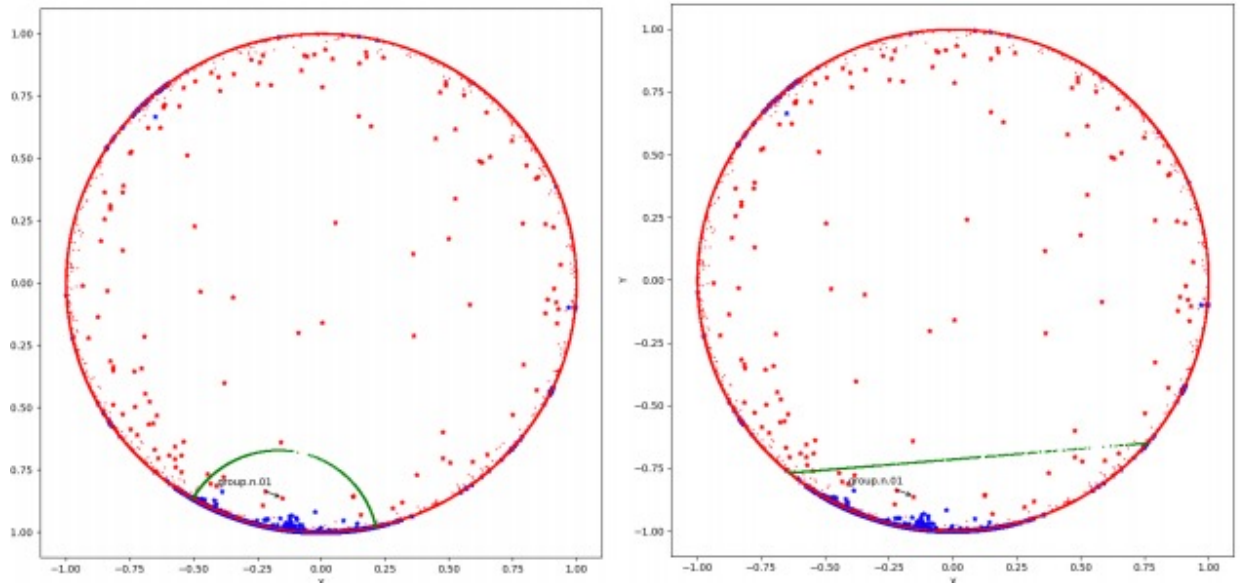
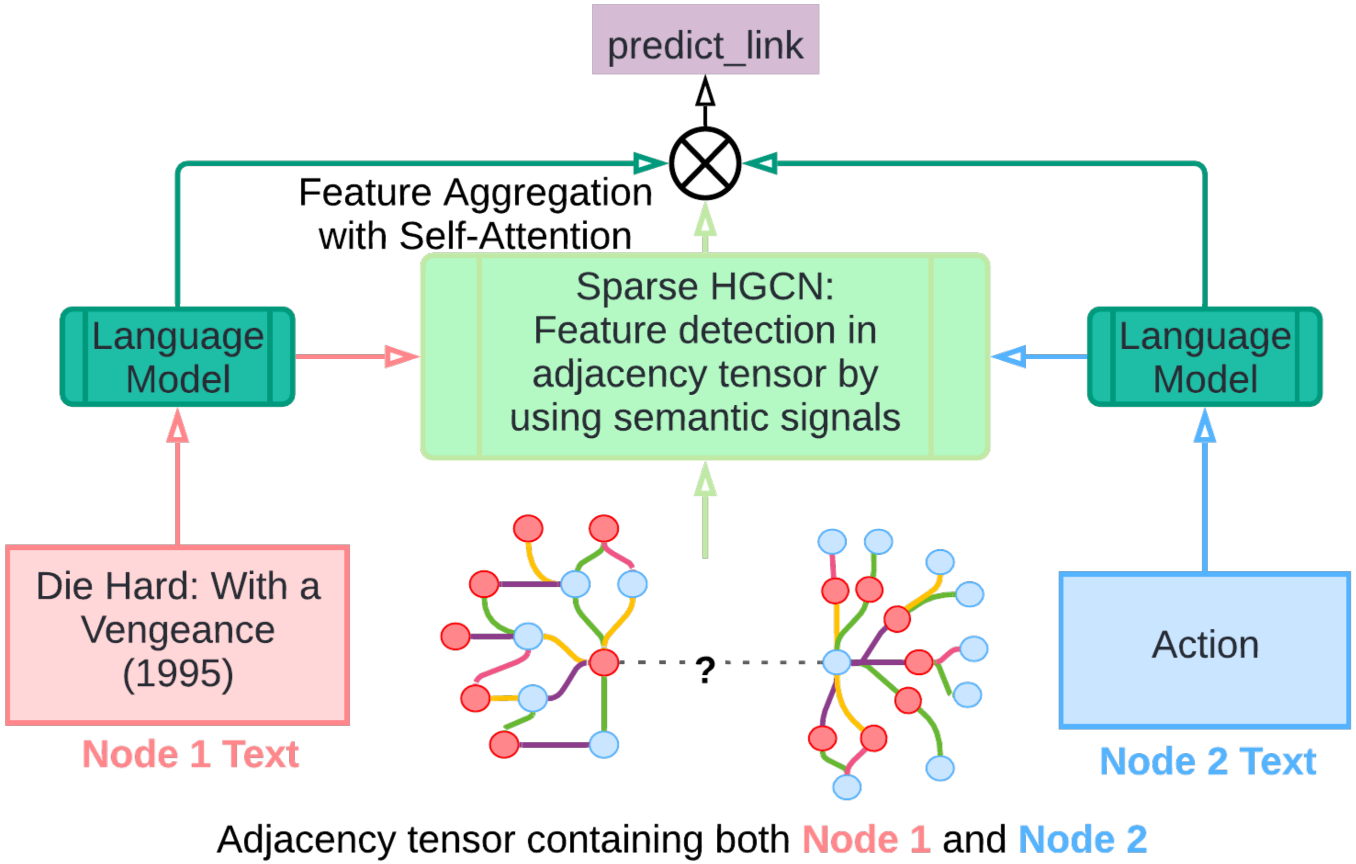


Image Source: Ganea, Octavian, Gary Bécigneul, and Thomas Hofmann. "Hyperbolic neural networks." Advances in neural information processing systems 31 (2018).

Applications

Multi-modal Graph Processing: TESH-GCN

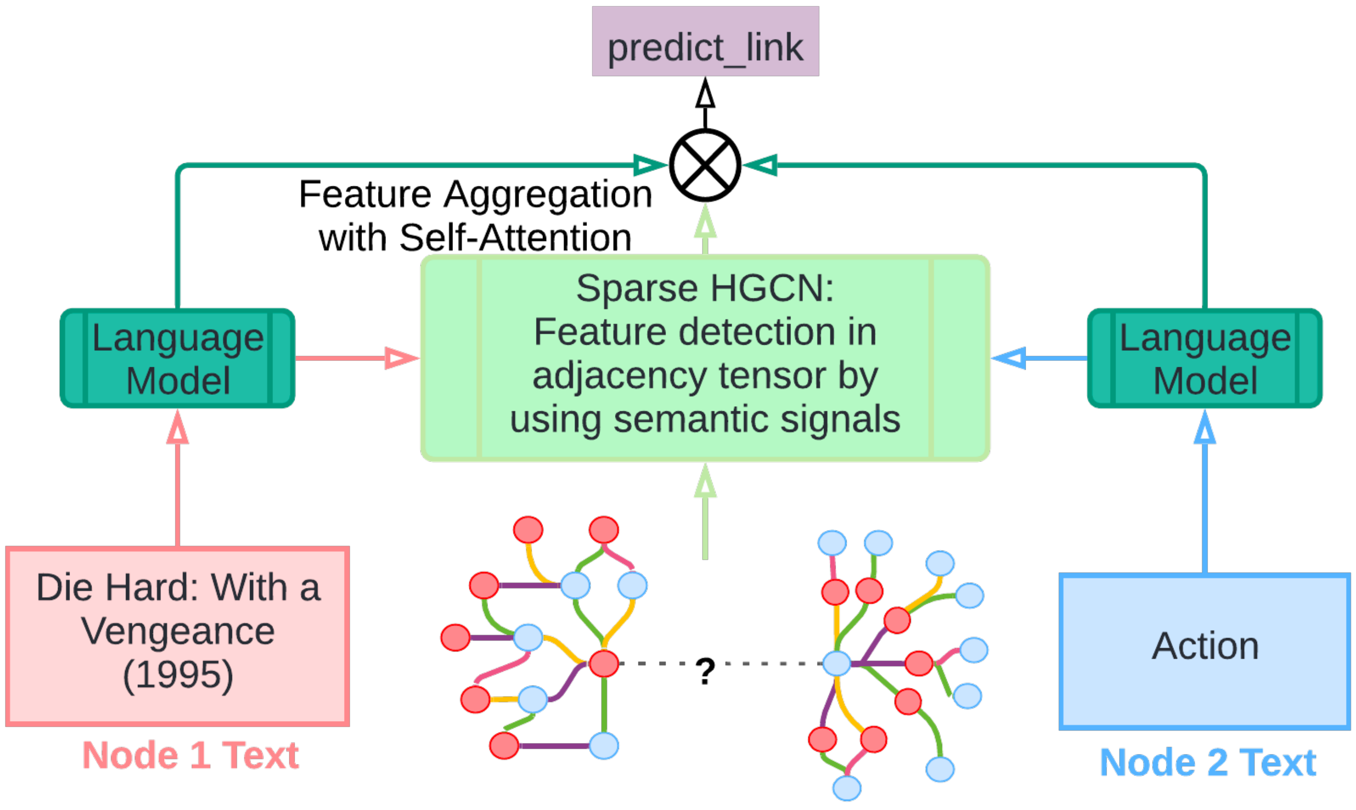
- Several graphs have inherent node information. However, the structure information is handled independently.
- Real-world graphs are also large and hence, it is difficult to capture global information.
- TESH-GCN is a hybrid graph-text based approach that aims to solve these issues.



Applications

Multi-modal Graph Processing: TESH-GCN

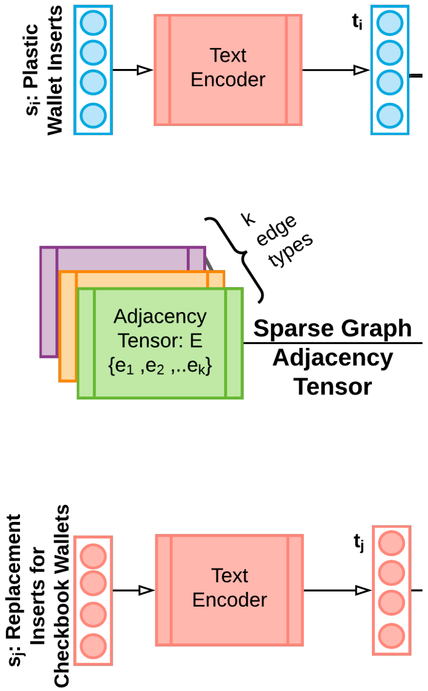
In TESH-GCN, we try to utilize the inherent node information to extract both local and global graph features and integrate it with the nodes' information to finally solve a downstream task.



Adjacency tensor containing both Node 1 and Node 2

Applications

Multi-modal Graph Processing: TESH-GCN



We use Text and Graph-based Link Prediction as a running example in this work. But the method is extensible to any other modality and downstream task.

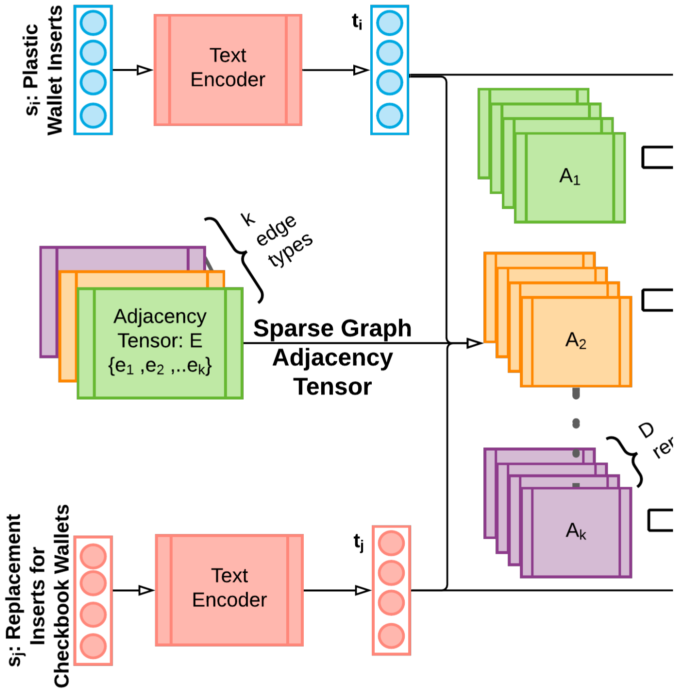
The inputs to TESH-GCN are:

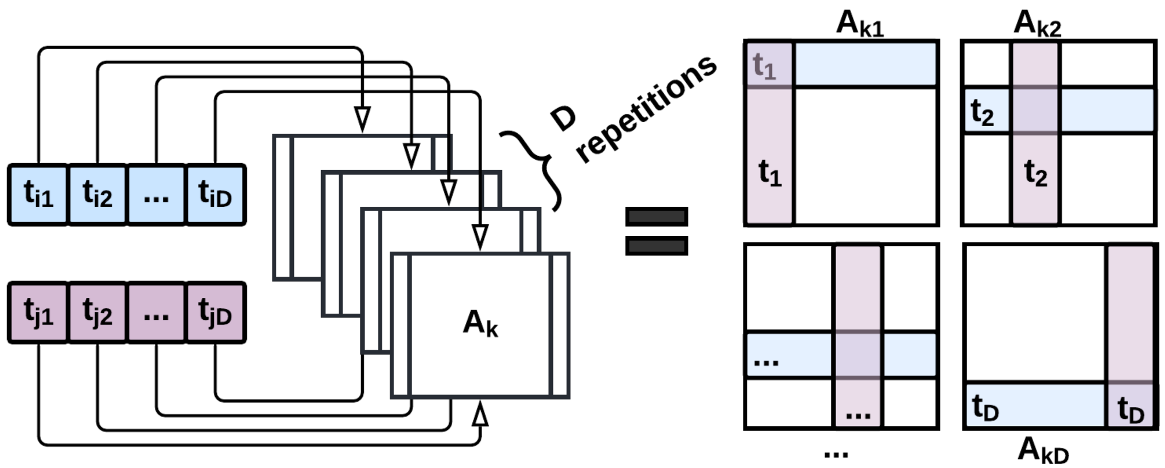
- Text from Node 1.
- Text from Node 2.
- Sparse Adjacency Tensor of the graph.
  - Graph is a tensor for multiple-relations and a matrix for the case of single-relation.

The text from Node 1 and Node 2 are encoded using a trainable language model.

Applications

Multi-modal Graph Processing: TESH-GCN

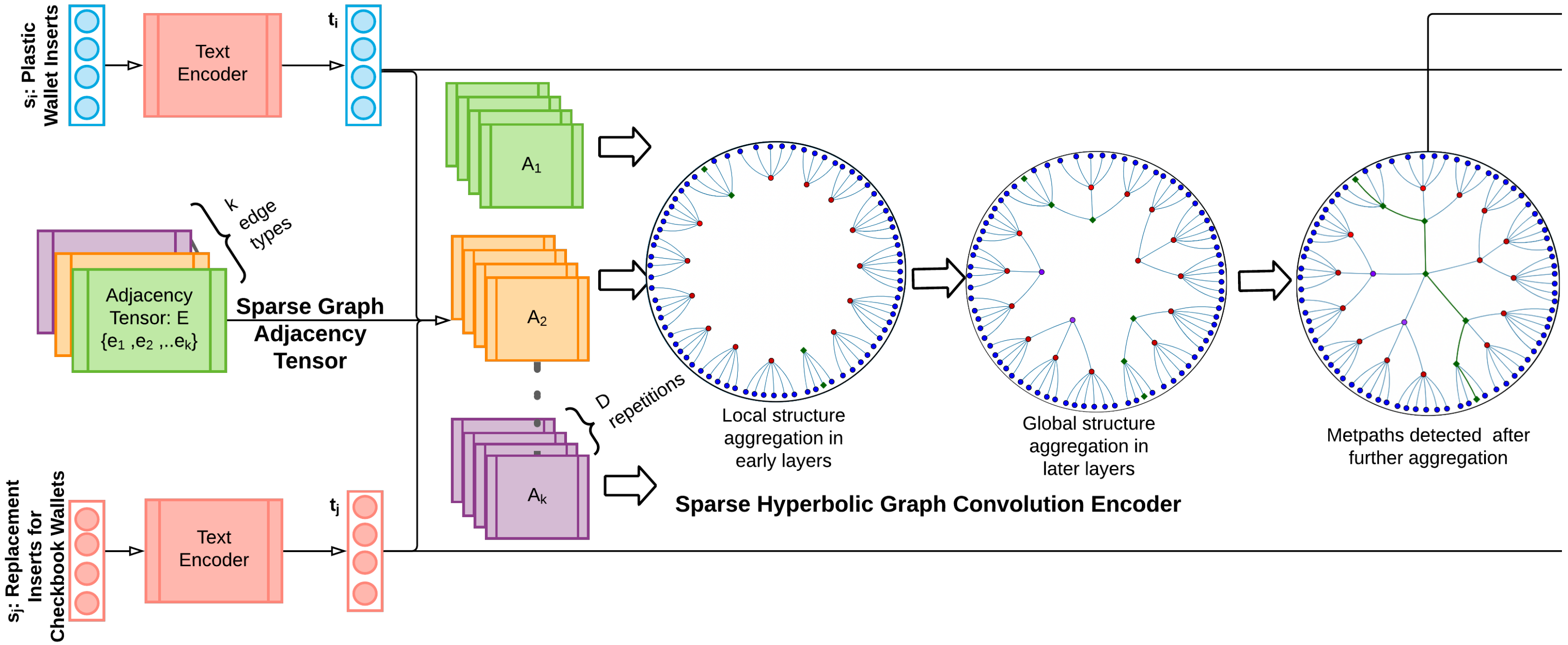




The text encoding from Node 1 and Node 2 are added to the Adjacency tensor to capture both semantic information and location of the nodes.

Applications

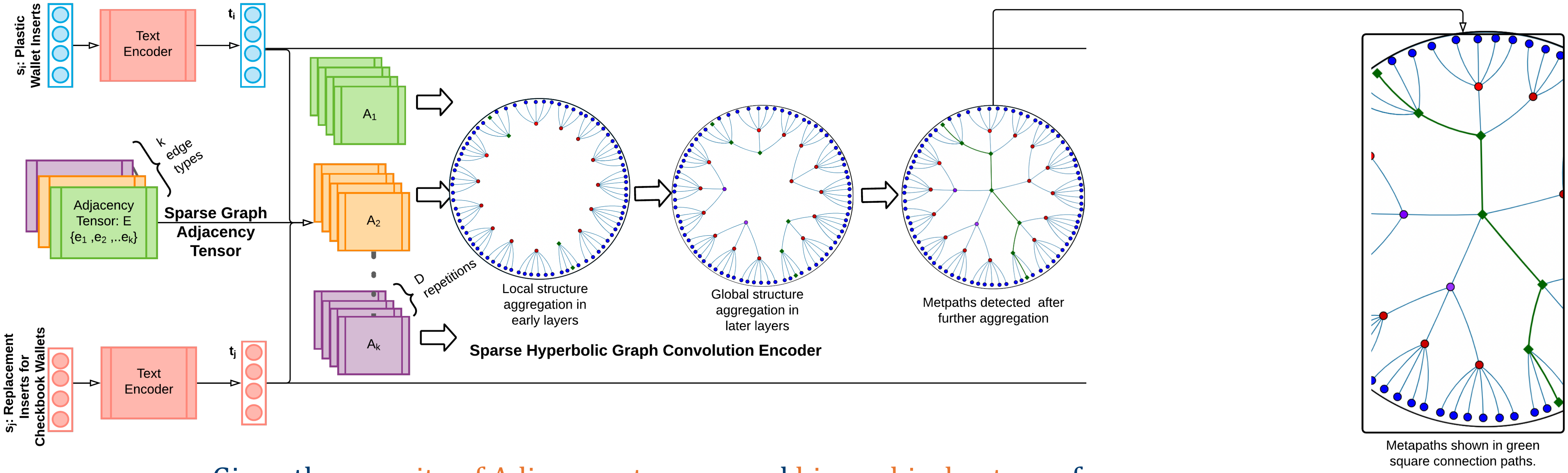
Multi-modal Graph Processing: TESH-GCN



Given the sparsity of Adjacency tensors and hierarchical nature of graph structure, we use an 8-layer Sparse HGCN encoder.
We notice that in most of the cases and 8-layer network is able to identify the metapath between the input nodes.

Applications

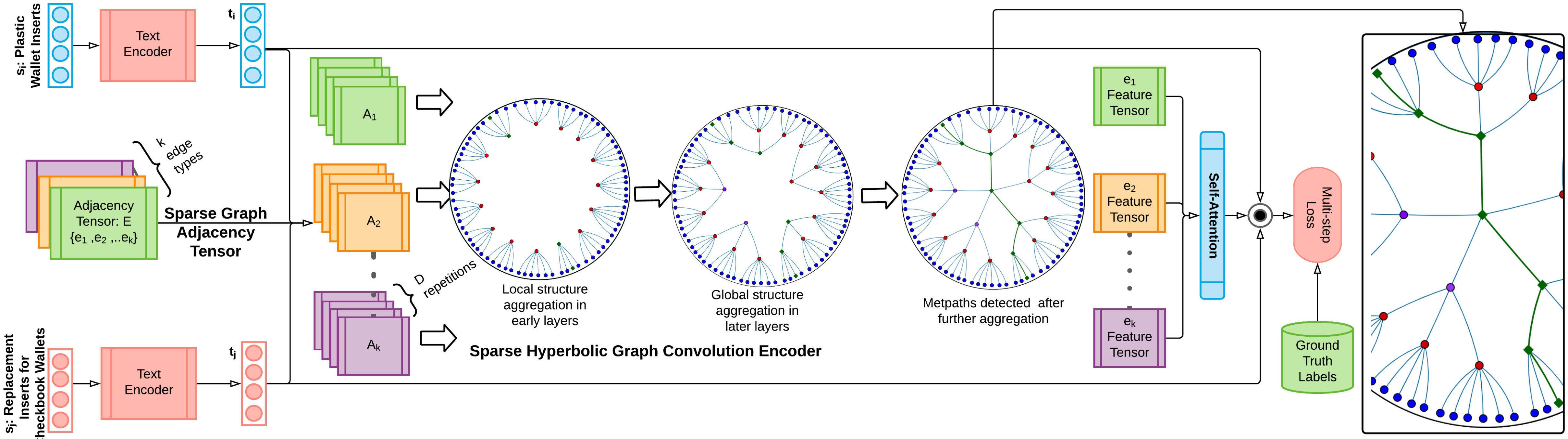
Multi-modal Graph Processing: TESH-GCN



Given the sparsity of Adjacency tensors and hierarchical nature of graph structure, we use an 8-layer Sparse HGCN encoder.
We notice that in most of the cases and 8-layer network is able to identify the metapath between the input nodes.

Applications

Multi-modal Graph Processing: TESH-GCN



Metapaths shown in green square connection paths.

Finally, to aggregate the multi-modal information, we use a self-attention network over the text and graph features.

Applications

Multi-modal Graph Processing: Evaluation

1. Performance on Link Prediction
2. Ablation Study
3. Time and Memory Complexity
4. Robustness against Noise
5. Example Metapaths

- Task: Link Prediction
  - Dataset: Amazon, DBLP, Twitter, Cora, MovieLens
  - Baselines: Text-based (BERT), Graph-based (HGCN), Hybrid approach (TextGCN)
  - Evaluation Metrics: Accuracy, F1

Applications

Multi-modal Graph Processing: Evaluation

	Amazon		DBLP		Twitter		Cora		MovieLens	
Models	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
BERT	0.787	0.784	0.604	0.603	0.667	0.641	0.757	0.751	0.76	0.752
HGCN	0.71	0.703	0.547	0.533	0.608	0.598	0.929	0.923	0.685	0.677
TextGCN	0.817	0.809	0.624	0.616	0.671	0.669	0.862	0.856	0.789	0.78
TESH-GCN	**0.829**	**0.836**	**0.636**	**0.640**	**0.709**	**0.670**	**0.940**	**0.918**	**0.806**	**0.801**

Applications

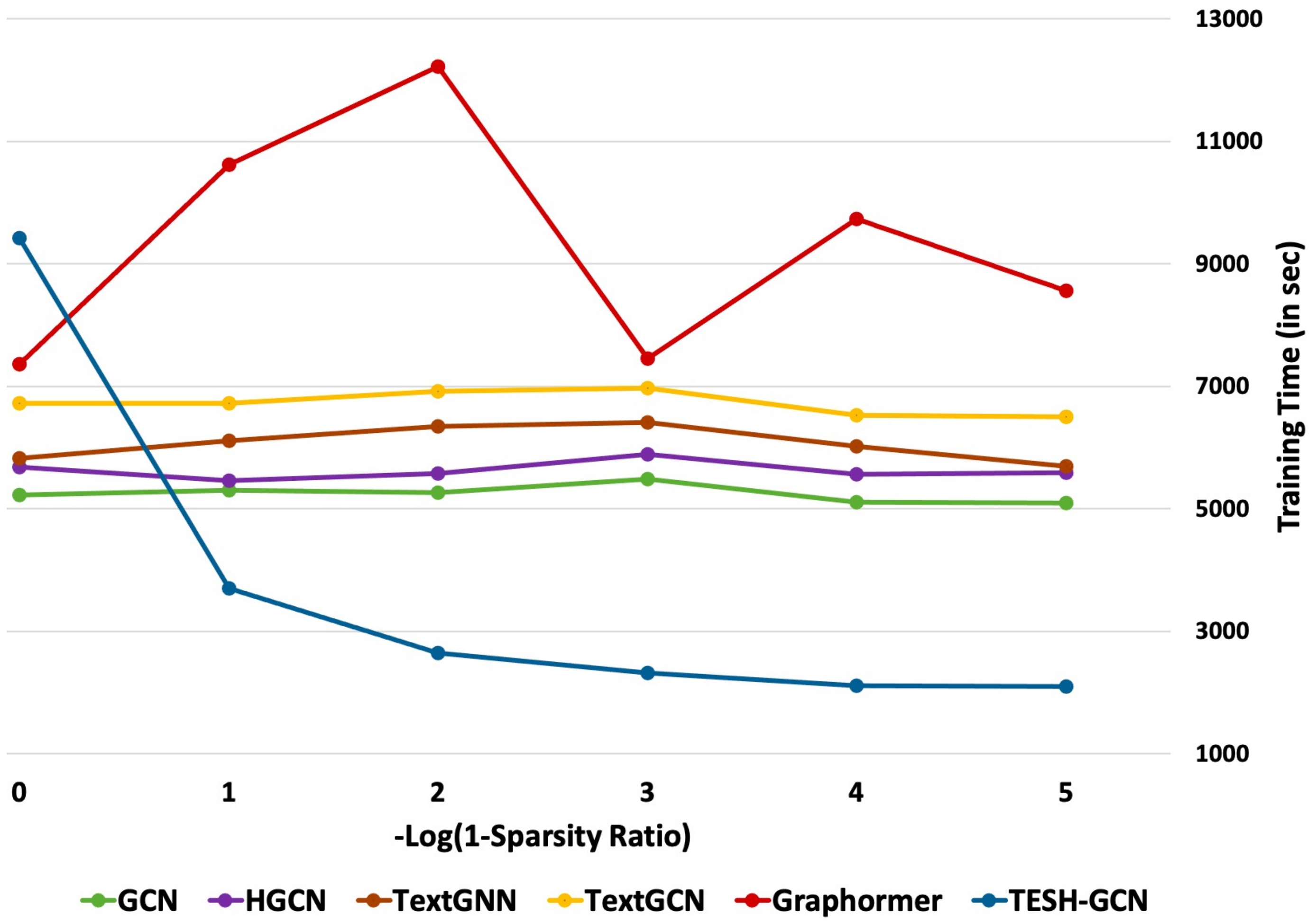
Multi-modal Graph Processing: Ablation Study

	Amazon		DBLP		Twitter		Cora		MovieLens	
Models	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
TESH-GCN	0.829	0.836	0.636	0.64	0.709	0.67	0.94	0.918	0.806	0.801
w/o Text	0.784	0.784	0.599	0.599	0.645	0.622	0.854	0.824	0.759	0.748
w/o Hyperbolic	0.677	0.678	0.522	0.516	0.577	0.585	0.787	0.757	0.655	0.66
w/o Residual	0.826	0.829	0.629	0.632	0.699	0.658	0.937	0.913	0.796	0.795

Applications

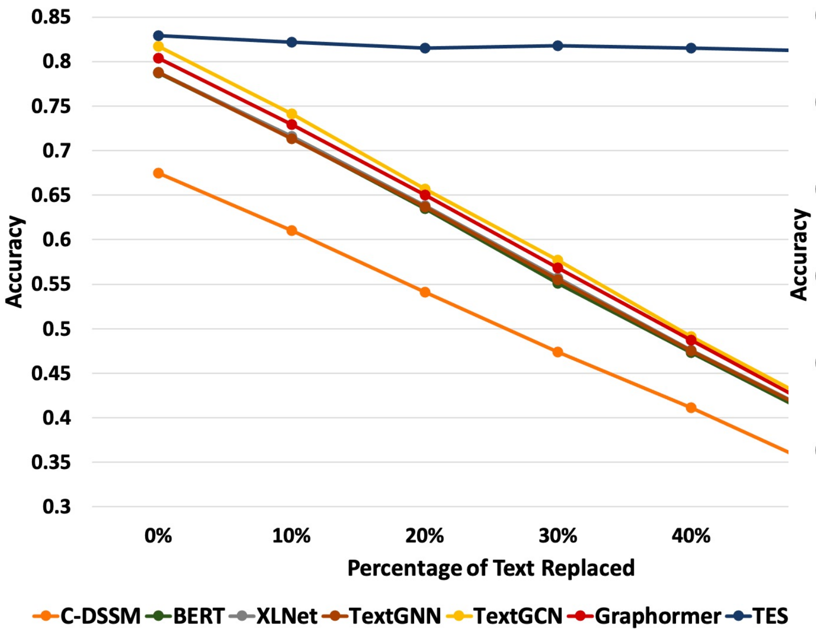
Multi-modal Graph Processing: Time and Memory Complexity

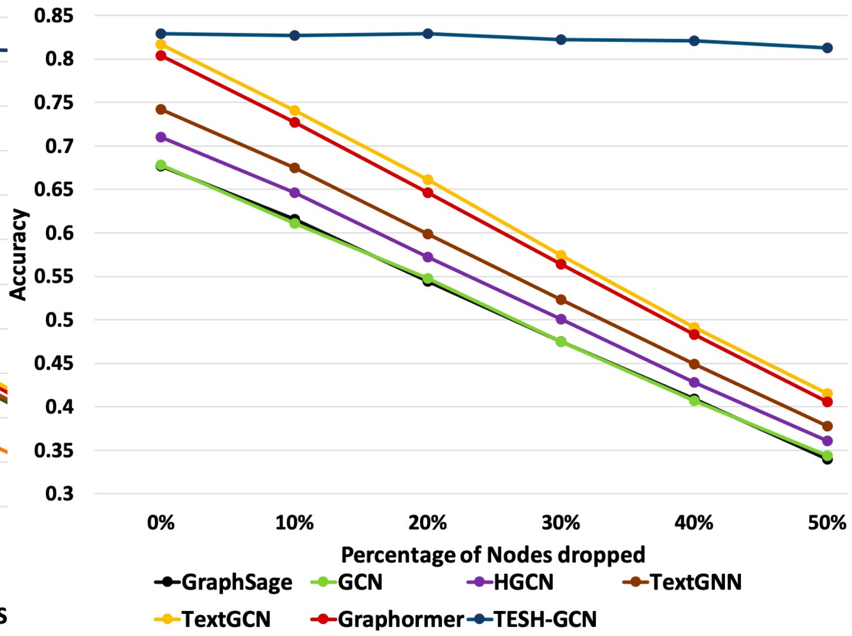
TESH-GCN's training time is more than other methods for dense graphs, but as the graph's get sparser, the training time reduces drastically.

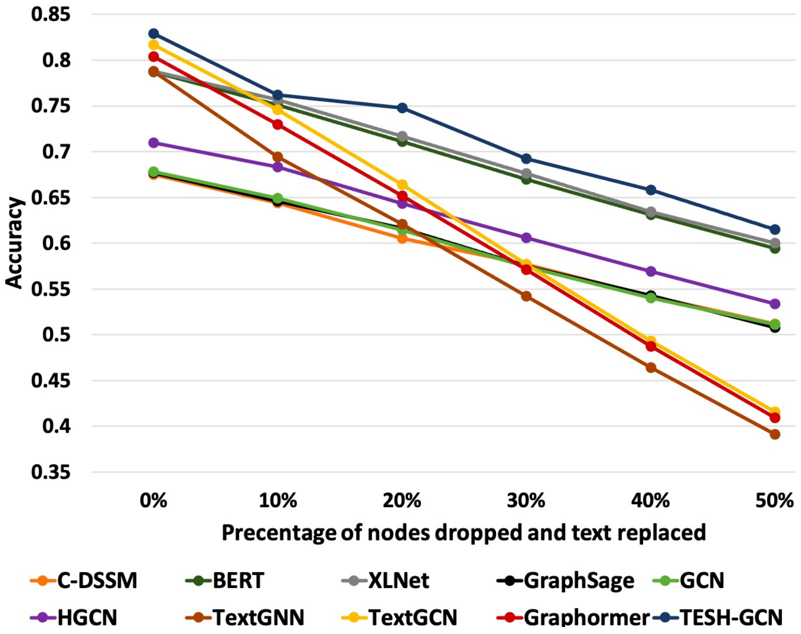


Applications

Multi-modal Graph Processing: Robustness



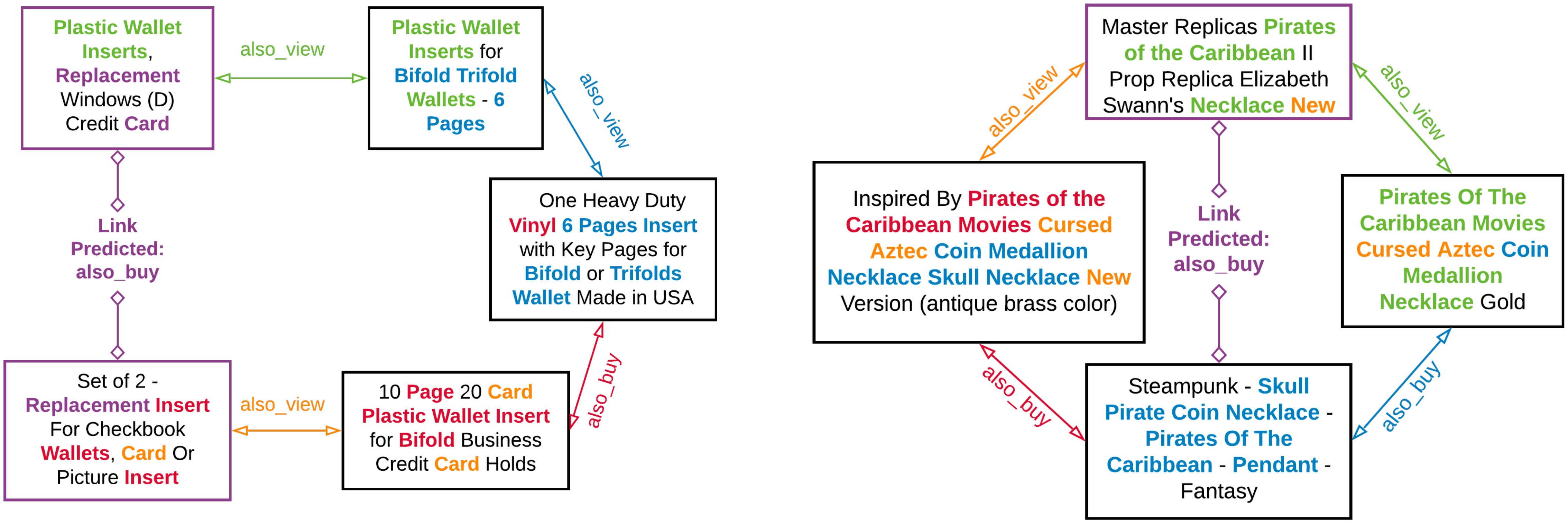




Robustness is tested by text replacement, node drops and a hybrid node drop with text replacement.

Applications

Multi-modal Graph Processing: Visualization of Metapaths



Q&A

Please provide your feedback at:

https://bit.ly/feedback-hyperbolic-tutorial

Part 5:
Future Directions

Conclusion
Hyperbolic Neural Networks

- Hyperbolic space is better at capturing hierarchical features due to its exponential growth in volume (depth ≈ radius).
- Hyperbolic space can simultaneously capture spatial and hierarchical structure information by pseudo-querying the knowledge graphs (HyPE).
- Attention mechanism can be used to capture intersection and union operations in search queries.
- HypE's representation, in congruence with matching architecture can be utilized for downstream tasks (query matching).
- The hyperbolic space can also be visualized for better human comprehension.

However, further study in the field shows certain interesting challenges.

Challenges

Hyperbolic Neural Networks

- As we mentioned previously, hyperbolic networks suffer from practical implementation challenges.
  - Non-availability of specific objective functions (or) normalization layers.
  - Unstable training and non-closure of hyperbolic networks.
- Application to complex tasks in Natural Language Processing and Computer Vision
  - Currently only limited to representation learning
- Development of scalable pre-trained architectures.
  - Hyperbolic networks are not able to leverage the GPU operations.
- Standardization of hyperbolic architectures as libraries for easy access.
  - Currently, a lot of theoretical background is required for development of hyperbolic networks, certain level of abstraction and standardization will help the initiation of new researchers in the area.

Future Directions

Implementational Challenges

We see development of new models of hyperbolic formulation such as; Pseudo-Poincaré and HNN++.

However, the following problems remain:

- Objective functions remains limited to hyperbolic distance, which is equivalent to L1-norm. There is scope for development of more complex classification and regression objectives.
- Along the lines of development of AdaM for Euclidean spaces, further study into the hyperbolic gradient descent is needed to provide new techniques for stable training.
- Not all Euclidean points are hyperbolic points. There is scope for practical development of specific hyperbolic libraries or theoretical development to satisfy the bound limitation of hyperbolic space.

Future Directions

Extension to Complex Applications

While graph research into hyperbolic spaces has been extended to complex problems such as logical reasoning. The scope in other domains has been limited to representation learning. Following are the possibilities in other domains;

- Natural Language Processing: The dependency tree structure of sentences can be used towards machine translation, question-answering, and document search.
- Computer Vision: The power of exponential volume growth can be used to hierarchically preserve both high-level features, such as object type, and low-level details for object identification.
- Networks: Hyperbolic space can hierarchically aggregate information from network clusters to process high-level details.

Future Directions

Scalable Pre-trained architectures

Due to the complexity of addition and multiplication operations, hyperbolic networks cannot properly utilize the power of GPUs. This limits our ability to develop architectures for effective training paradigms such as transfer learning and curriculum learning.

- Development of scalable formulation for GPUs is required.

- Subsequently, further development of large networks that can then utilize;
  - pretraining and finetuning approach of transfer learning and,
  - continuous improvements through curriculum learning

Future Directions

Standardization of Hyperbolic Neural Networks

Abstraction and Standardization of hyperbolic network theory and architectures is required.

GraphZoo* and this tutorial are a step in that direction.

The abstraction, especially, helps application-oriented research that would prefer black-box frameworks.

Interpretability frameworks such as LINE and SHAP also need to be developed for hyperbolic spaces to improve the trust in hyperbolic networks.

This will also improve human comprehension of the networks, so researchers can conduct targeted studies into the problem areas of hyperbolic networks.

* https://github.com/reddy-lab/GraphZoo

References

1. Choudhary, N., Rao, N., Katariya, S., Subbian, K., & Reddy, C. K. (2022, February). ANTHEM: Attentive Hyperbolic Entity Model for Product Search. In Proceedings of the International Conference on Web Search and Data Mining 2022 (WSDM '22).
2. Vyas, A., Choudhary, N., Khatir, M., & Reddy, C. K. (2022). GraphZoo: A Development Toolkit for Graph Neural Networks with Hyperbolic Geometries. Companion Proceedings of the Web Conference 2022 (WWW '22).
3. Choudhary, N., Rao, N., Katariya, S., Subbian, K., & Reddy, C. (2021). Probabilistic Entity Representation Model for Reasoning over Knowledge Graphs. Advances in Neural Information Processing Systems, 34 (NeurIPS '21).
4. Choudhary, N., Rao, N., Katariya, S., Subbian, K., & Reddy, C. K. (2021, April). Self-Supervised Hyperboloid Representations from Logical Queries over Knowledge Graphs. In Proceedings of the Web Conference 2021 (pp. 1373-1384) (WWW '21).
5. Shimizu, R., Mukuta, Y., & Harada, T. (2021). Hyperbolic neural networks++. International Conference on Learning Representations (ICLR '21).
6. W. Peng, T. Varanka, A. Mostafa, H. Shi and G. Zhao, Hyperbolic Deep Neural Networks: A Survey, in IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI 2021).
7. Sun, J., Cheng, Z., Zuberi, S., Pérez, F., & Volkovs, M. (2021, April). Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference 2021* (pp. 593-601) (WWW '21).
8. Chami, I., Wolf, A., Juan, D. C., Sala, F., Ravi, S., & Ré, C. (2020). Low-dimensional hyperbolic knowledge graph embeddings, In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL '20).*
9. Chami, I., Ying, Z., Ré, C., & Leskovec, J. (2019). Hyperbolic graph convolutional neural networks. Advances in neural information processing systems, 32, 4868-4879 (NeurIPS '19).
10. Balazevic, I., Allen, C., & Hospedales, T. (2019). Multi-relational poincaré graph embeddings. *Advances in Neural Information Processing Systems, 32 (NeurIPS '19).*
11. Ganea, O. E., Bécigneul, G., & Hofmann, T. (2018). Hyperbolic neural networks. Advances in neural information processing systems, 5345-5355 (NeurIPS '18).
12. Tay, Y., Tuan, L. A., & Hui, S. C. (2018, February). Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (pp. 583-591) (WSDM '18).

Thanks!

Any questions?

Find me at:

https://nurendra.me

nurendra@vt.edu

The toolkit is available at:

https://github.com/reddy-lab/GraphZoo