Dynamic Curvature Optimization for Hyperbolic GCNs

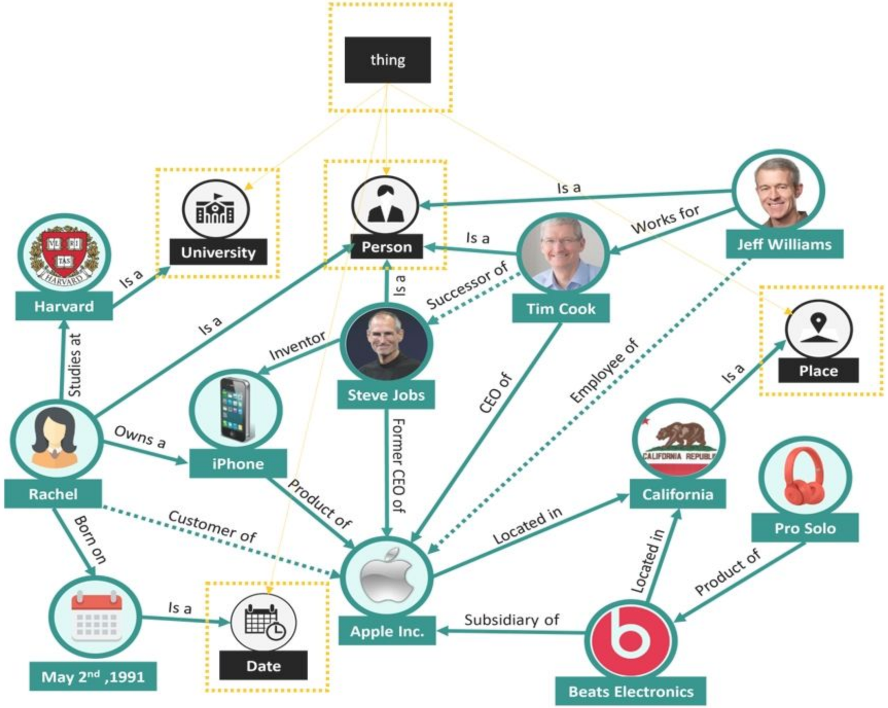
Leveraging Hyperbolic Geometry for Hierarchical Knowledge Graph Data

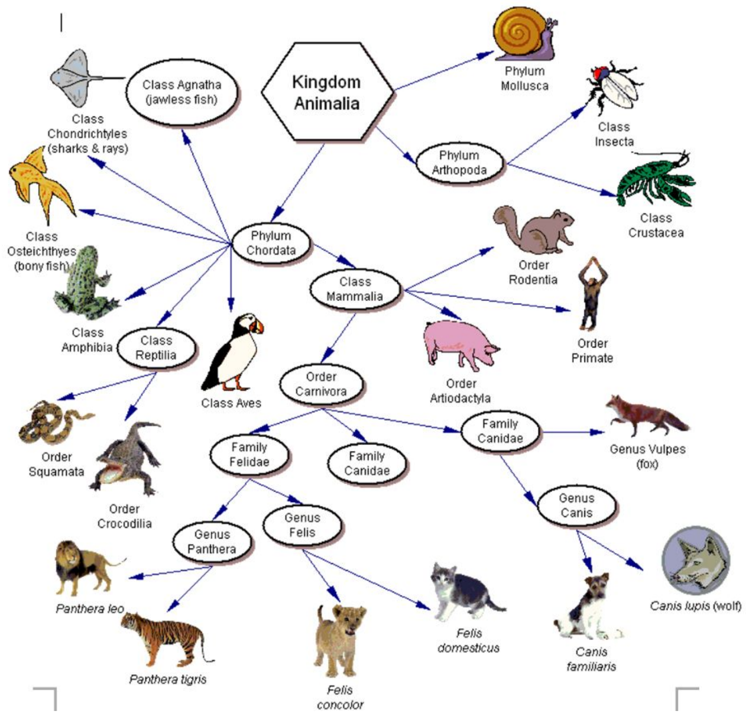
Optimization and Computation: S&DS 431

Valentina Simon and Arjan Kohli

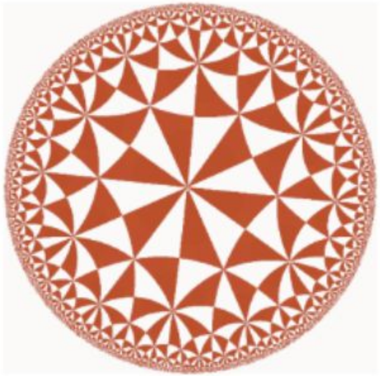
Introduction and Problem Statement

MOTIVATION: Real World Hierarchical Data:

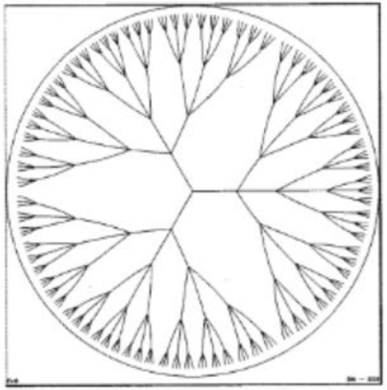




MOTIVATION: Trees and Hyperbolic Geometry



Poincaré: intuitive visualization



a hierarchical tree

In a tree, the number of nodes **grows exponentially** with the tree depth

The volume of a Poincare model in the hyperbolic space **grows exponentially** with its radius!

Thus, hierarchical relationships can be preserved more successfully and efficiently hyperbolically!

What is a Hyperbolic Graph Convolutional Neural Network (HGCN)?

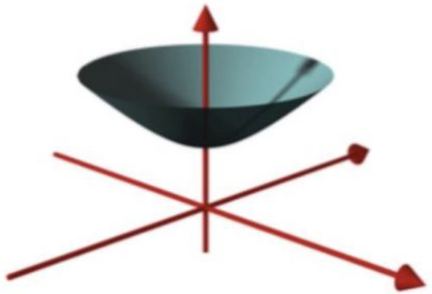


Figure 1: Hyperboloid Model
s

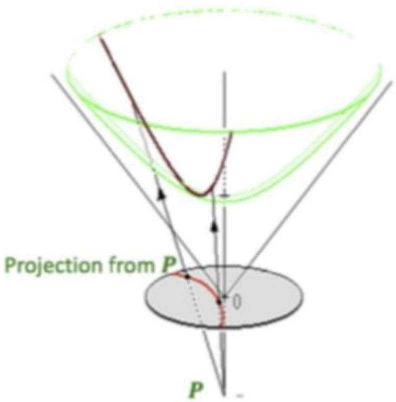


Figure 2: Hyperboloid Projection

HGCNs model hierarchical and complex relational structures in graph data

Negative curvature of hyperbolic geometry ➔ lower-distortion representations of tree-like graphs

HGCNs have shown an error reduction of up to 63.1% in ROC AUC for link prediction, in comparison to GCNs

Foundations and Advances in Hyperbolic Geometry for GCNs

Hyperbolic Embeddings

Poincaré embeddings (Nickel & Kiela) laid the foundation for hierarchical representations.

Hyperbolic GCNs

Chami et al. introduced curvature modeling at the layer level. Demonstrated that hyperbolic geometry at the layer level improves representation of hierarchical data.

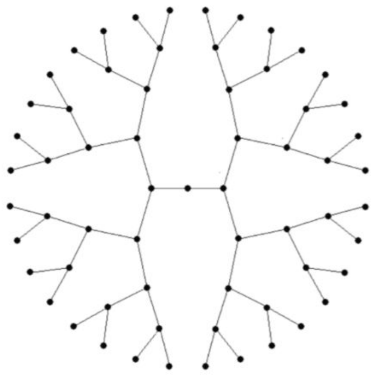
Advances in Attention

Zhang et al. integrated attention mechanisms into hyperbolic spaces (HGATs), capturing complex relational patterns more efficiently.

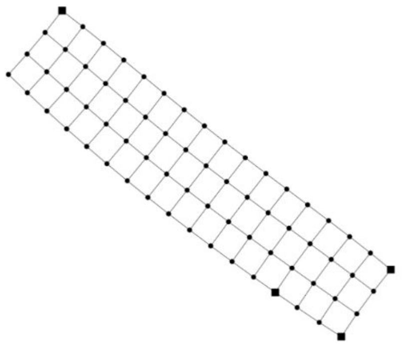
Transformers in Hyperbolic Space

Hype-GT applied hyperbolic encodings to graph to transformers. capturing long range dependencies in hierarchical relationships

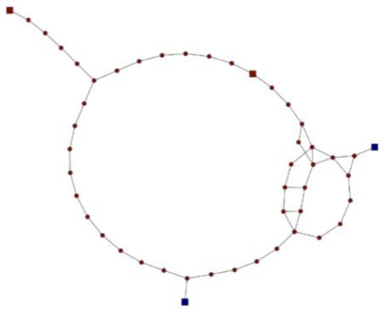
Delta Hyperbolicity: Intuition



$\delta = 0$



$\delta = 3.0$



$\delta = 4.5$

Measures how hierarchical or “tree like” a graph is
Note, if δ=0, the space is perfectly tree-like

Delta Hyperbolicity: Formal Definition

Let (X, d) be a metric space. For any three points $x, y, z \in X$, consider their pairwise distances:

$$d(x, y),\ d(y, z),\ d(z, x).$$

Define the Gromov product of x and z relative to y as:

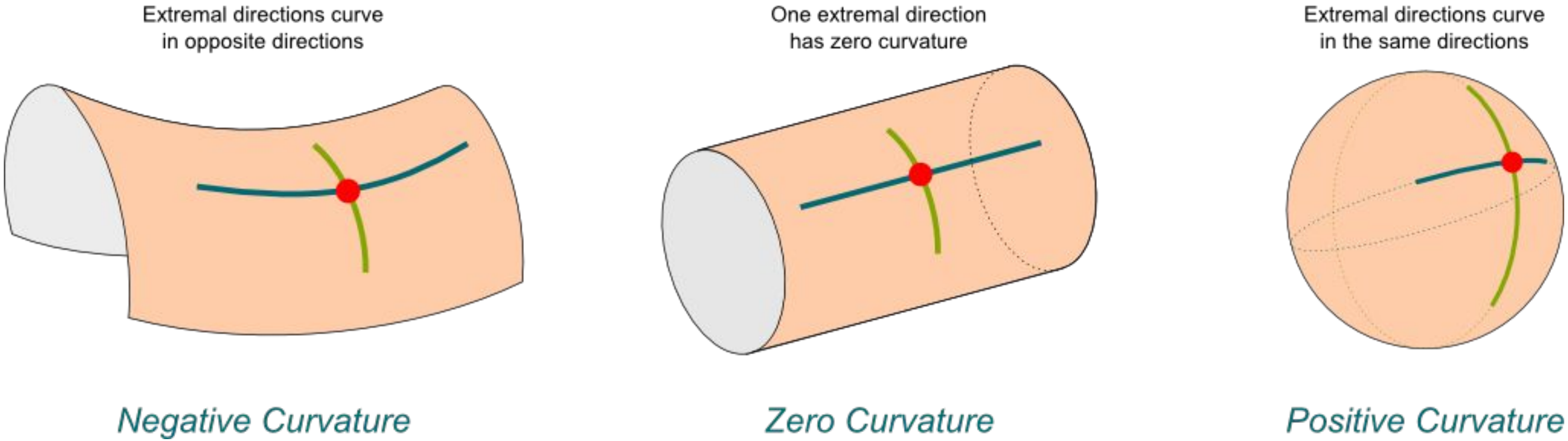
$$(x \cdot z)_y = \frac{1}{2}\left(d(x, y) + d(z, y) - d(x, z)\right).$$

This measures how close y is to being on the geodesic between x and z.

A metric space X is called δ-hyperbolic if there exists a constant $\delta \geq 0$ such that, for all points $x, y, z, w \in X$, the following inequality holds:

$$(x \cdot z)_w \geq \min\{(x \cdot y)_w, (y \cdot z)_w\} - \delta.$$

NOVELTY: Dynamic Curvature



Literature: Fixed Global Curvature (Untrained) or uniquely optimized curvature for each HGCN layer

Our Work: Optimized Global Curvature, trained as a parameter in the HGCN model

Adapts to Graph Structure and Reduces Model Complexity!

Objectives:

(1) Demonstrate success of HGCNs with globally optimized curvature over GCNs

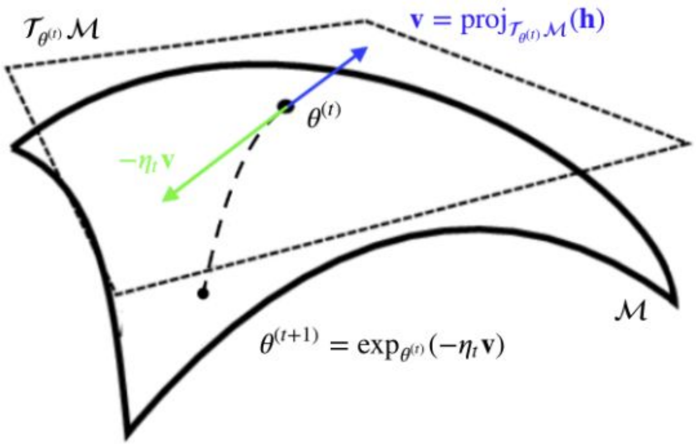
(2) Investigate relationship between hyperbolicity and globally optimal curvature in HGCNs

(3) Identify best optimizer for curvature in HGCNs

- SGD Riemannian Optimizer
- Adam Riemannian Optimizer
- Mixed Precision Riemannian Optimizer

Optimization Techniques

Riemannian Optimization



- The tangent space projection uses the Riemannian metric $g_x = \frac{2}{1-\|x\|^2}$.
- The exponential map is defined as:

$$\mathrm{Exp}_x(v) = \tanh\left(\frac{\|v\|}{1-\|x\|^2}\right)\frac{v}{\|v\|} + x.$$

Riemannian SGD Optimization

Let $x_t \in \mathcal{M}$ denote the current parameters on the manifold. At iteration t, Riemannian SGD proceeds as follows:

Initialization: Set $t = 0$.

Update Steps:

1. Compute the **Riemannian gradient**:
$$g_t = \nabla_{\mathcal{M}} f(x_t).$$
2. Compute the **tangent space update vector**:
$$u_t = -\alpha g_t,$$
where α is the learning rate.
3. Update the parameters using the **exponential map**:
$$x_{t+1} = \mathrm{Exp}_{x_t}(u_t),$$

Riemannian Adam Optimization

Let $x_t \in \mathcal{M}$ denote the current parameters on the manifold. At iteration t, the Riemannian Adam optimizer proceeds as follows:

Initialization: Set $m_0 = 0$, $v_0 = 0$, and $t = 0$.

Algorithm:

1. Compute the **Riemannian gradient**:
$$g_t = \nabla_{\mathcal{M}} f(x_t).$$
2. Update the **biased first and second moment estimates**:
$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t \odot g_t, \end{aligned}$$
where $\odot$ denotes element-wise multiplication.
3. Apply **bias correction**:
$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1-\beta_1^t}, \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t}. \end{aligned}$$
4. Compute the **tangent space update**:
$$u_t = -\alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon},$$
where α is the learning rate and ϵ is a small constant.
5. Update the parameters using the **exponential map**:
$$x_{t+1} = \mathrm{Exp}_{x_t}(u_t),$$

Mixed Precision Riemannian Optimization

Ensures numerical stability by temporarily scaling gradients before applying optimization steps

Let $x_t \in \mathcal{M}$ denote the parameters on the manifold at iteration t. The mixed precision algorithm proceeds as follows:

Initialization: Set $m_0 = 0$, $v_0 = 0$, $t = 0$, and initialize the gradient scale factor scale.

Algorithm:

1. Compute the **scaled Riemannian gradient**:

$$g_t = \text{scale} \cdot \nabla_{\mathcal{M}} f(x_t).$$

2. Update the **biased moment estimates**:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t,$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(g_t \odot g_t).$$

3. Apply **bias correction**:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t},$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}.$$

4. Compute the **scaled tangent space update**:

$$u_t = -\alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}.$$

5. Unscale the update by dividing by scale:

$$u_t' = \frac{u_t}{\text{scale}}.$$

6. Update the parameters using the **exponential map**:

$$x_{t+1} = \text{Exp}_{x_t}(u_t').$$

Datasets

Dataset Statistics

Lower Gromov-δ-hyperbolicity values imply more hyperbolic/hierarchical data.
Entities are Vertices and Relations are Edges.

WordNet
δ-hyperbolicity: 5.5, Entities: 40,945, Relation Classes: 19.

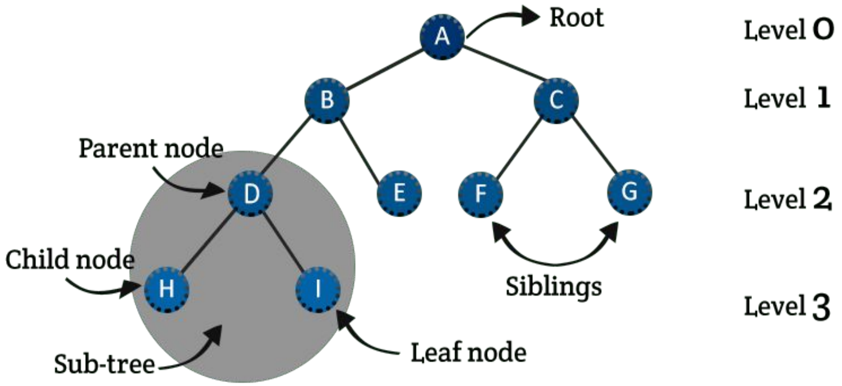
PubMed
δ-hyperbolicity: 3.5, Entities: 19,717, Relation Classes: 3.

FB15K
δ-hyperbolicity: 1.5, Entities: 14,541, Relation Classes: 237.

Diseases
δ-hyperbolicity: 0, Entities: 2,665, Relation Classes: 2.

Disease

δ-hyperbolicity: 0, Entities: 2,665, Relation Classes: 2.



Synthetic Dataset

Nodes Represent people

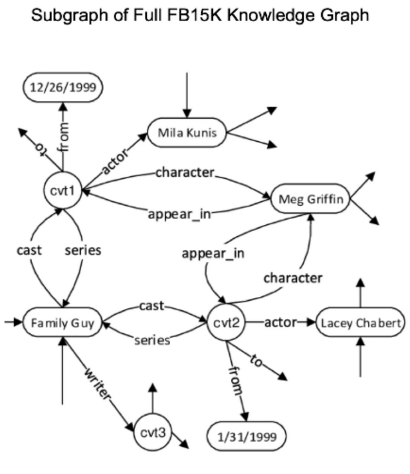
Edges represent Disease Transmission

Perfectly Tree-like, Hierarchical, and Hyperbolic Structure

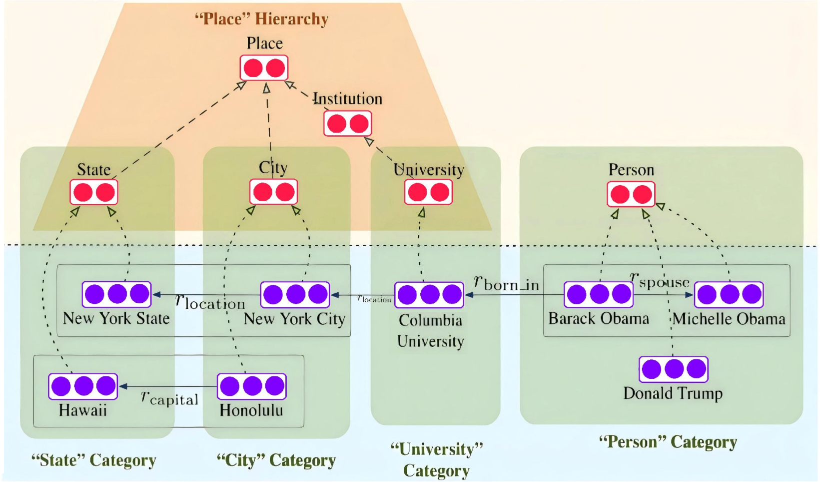
FB15K

δ-hyperbolicity: 1.5, Entities: 14,541, Relation Classes: 237.

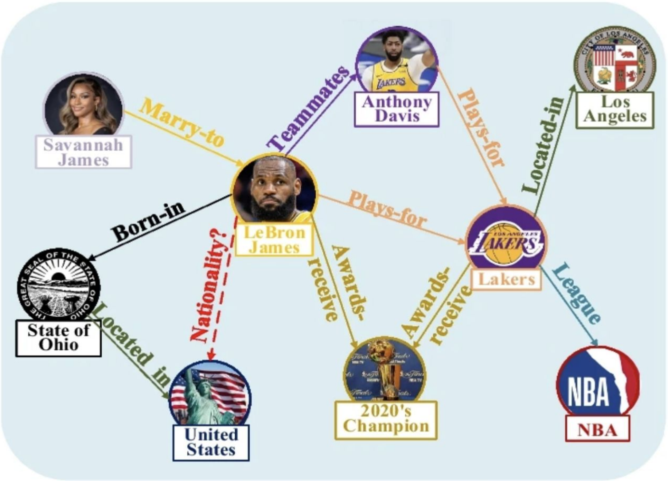
Knowledge Graph



Hierarchical Embeddings

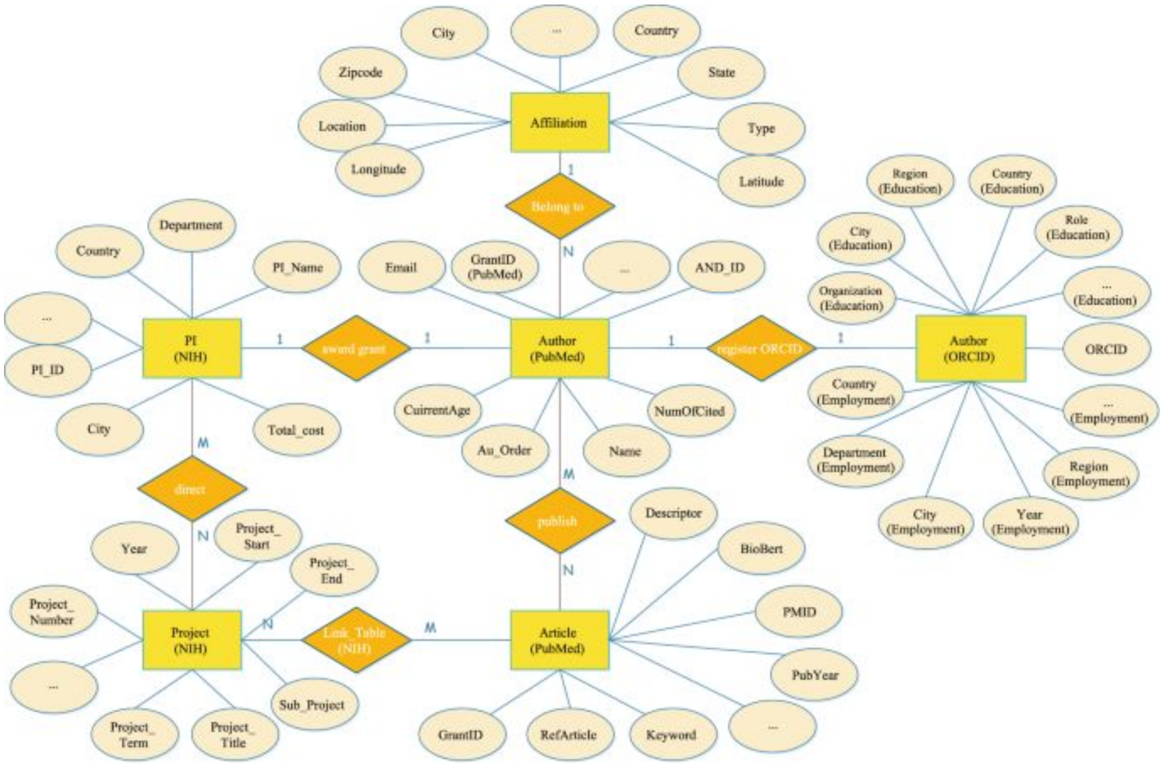


Link Prediction



PubMed

δ-hyperbolicity: 3.5, Entities: 19,717, Relation Classes: 3.



WordNet Random Sampling

δ-hyperbolicity: 5.5, Entities: 40,945, Relation Classes: 19.

Wordnet is a hierarchical knowledge graph of word relationships. Although the full dataset is a tree with hierarchical and hyperbolic structure, we constructed a synthetic dataset from a subset of Wordnet that was selected to have low Hyperbolicity ($\delta = 5.5$).

The most common relation we sampled was synonyms, in order to achieve a bipartite graph structure.

Results

Training Process: Structural Development Over Epochs

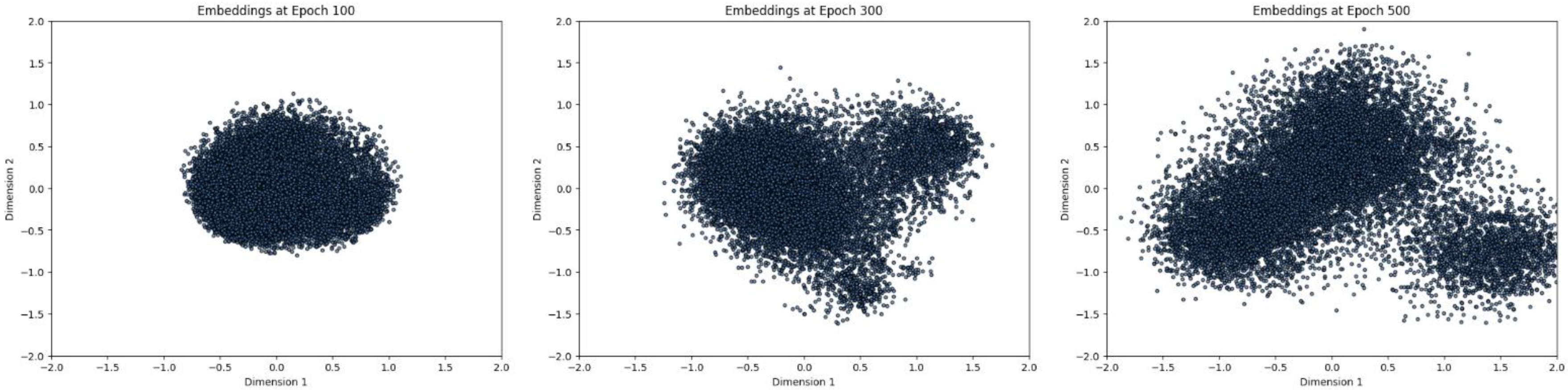


Figure 3: PCA of node embeddings at 100, 300, and 500 epochs. Structure is observed as model converges to a curvature and more node separation is visible in low dimensions.

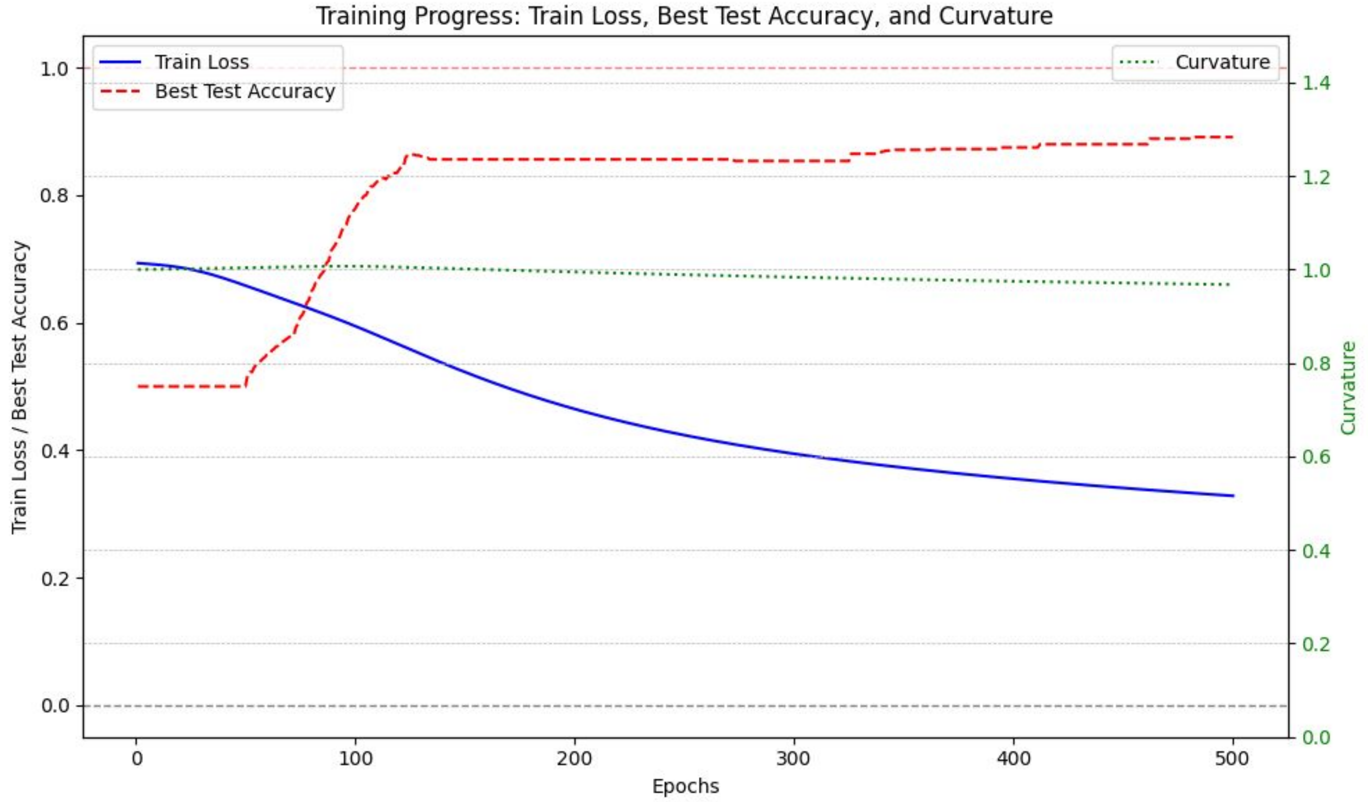
HGCN vs GCN - Disease Dataset

Model	Optimizer	Learning Rate	Mean Accuracy (%)	Standard Deviation (%)
Euclidean GCN	Euclidean Adam	$1e^{-4}$	85.62	0.95
HGCN with learnable curvature	Riemannian Adam	$1e^{-4}$	91.13	1.23

Learned Curvature vs Accuracy over Time (Riemannian ADAM)

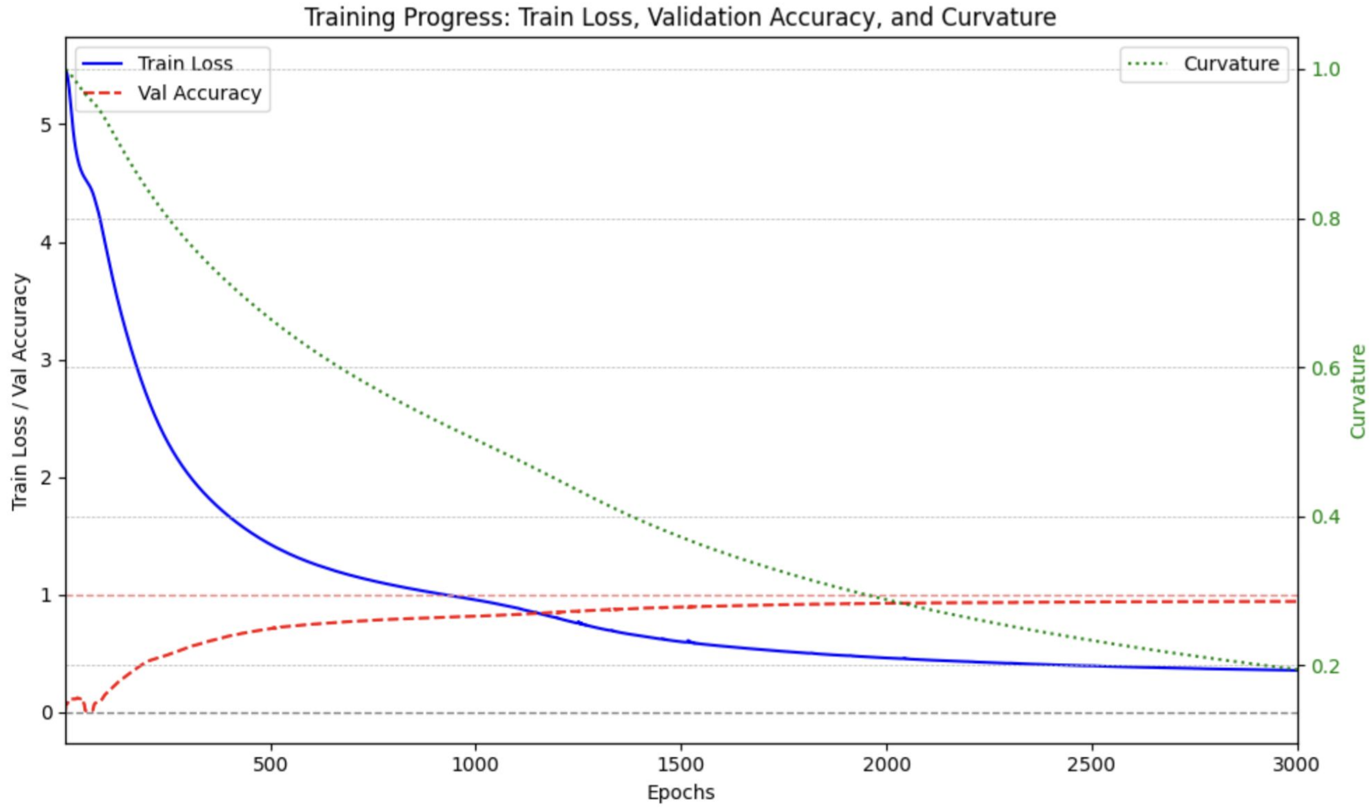
δ-hyperbolicity: 0

Disease



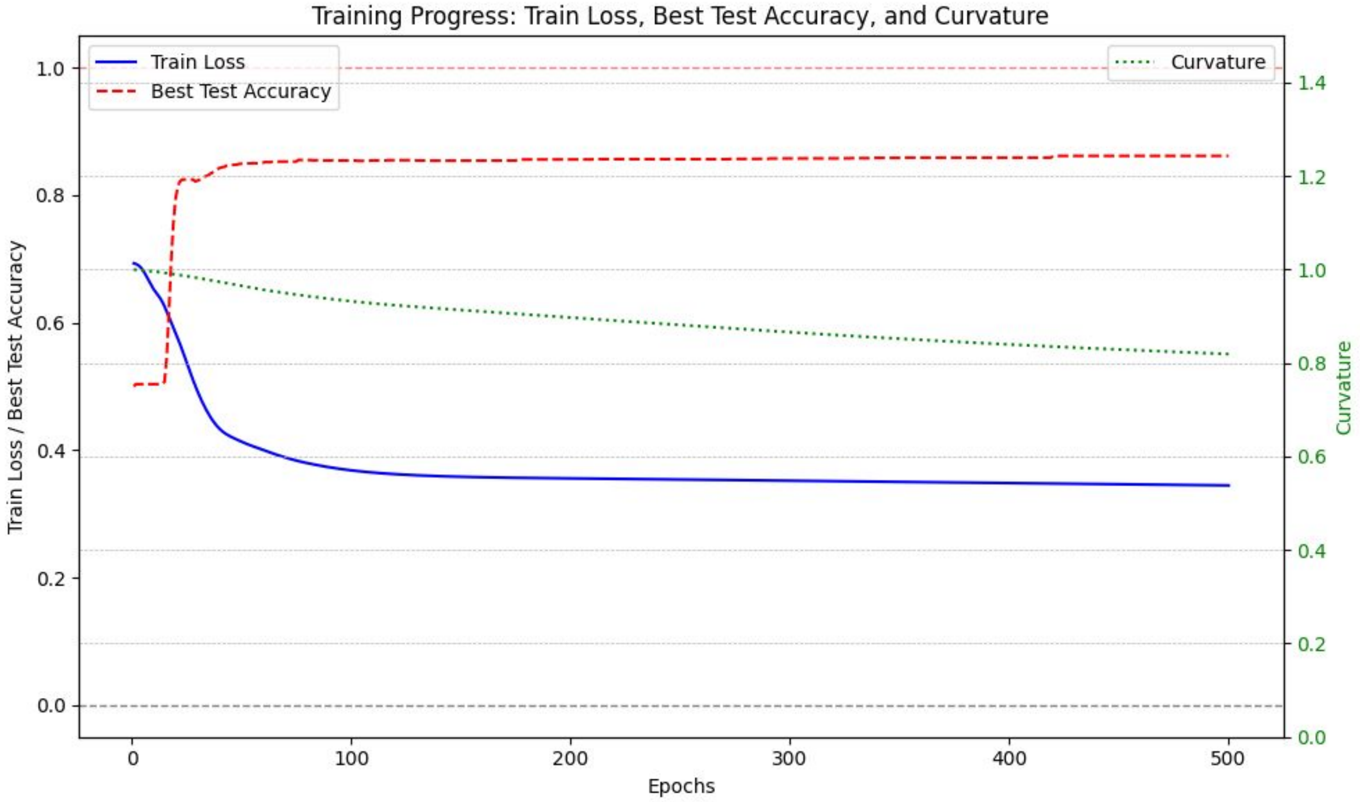
δ-hyperbolicity: 1.5

FB15k



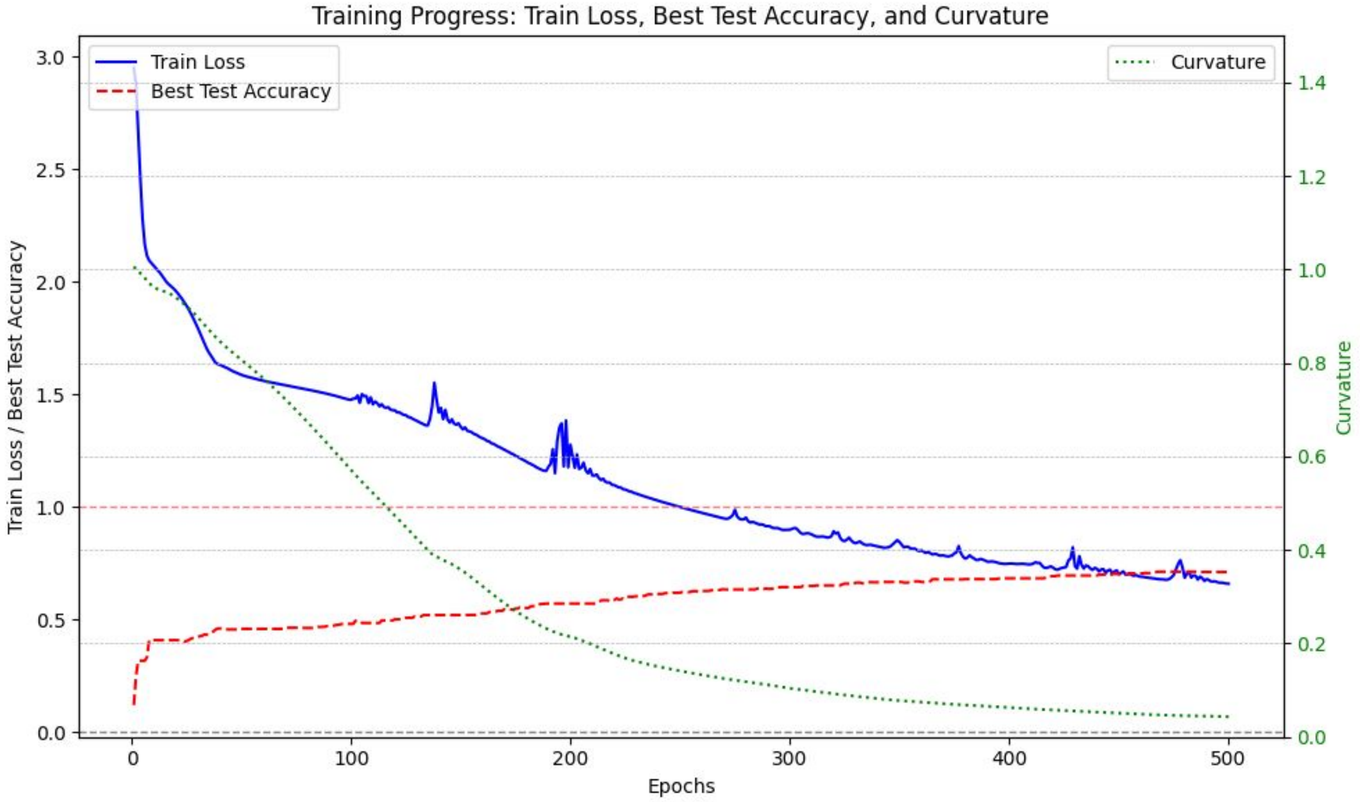
δ-hyperbolicity: 3.5

Pubmed



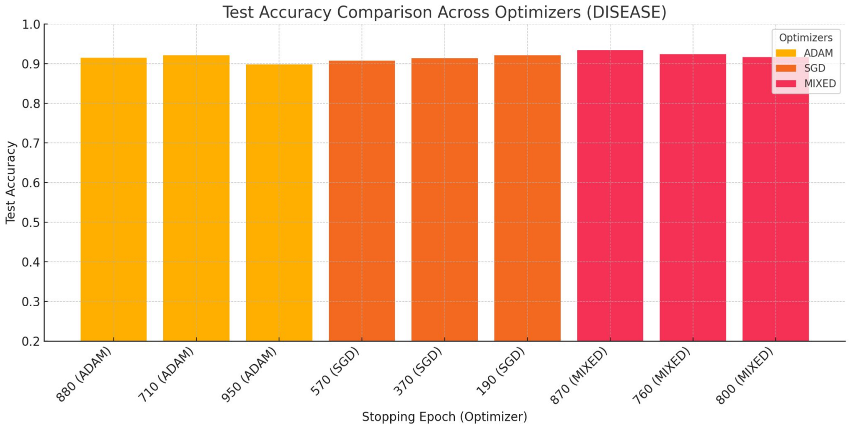
δ-hyperbolicity: 5.5

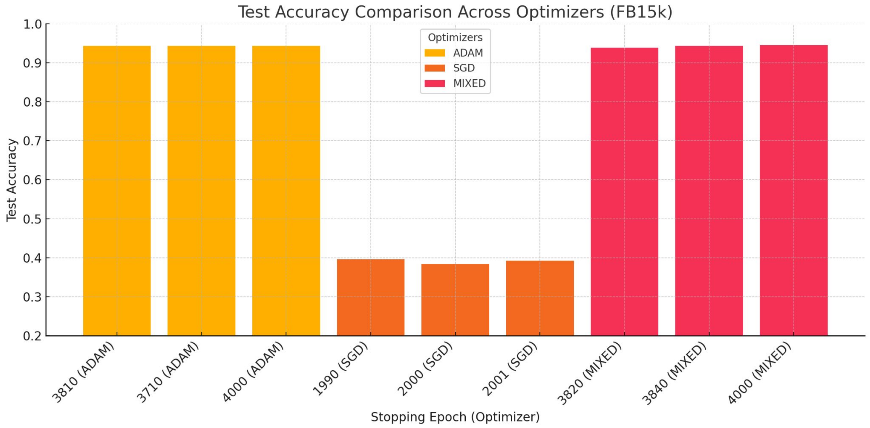
Wordnet Subset

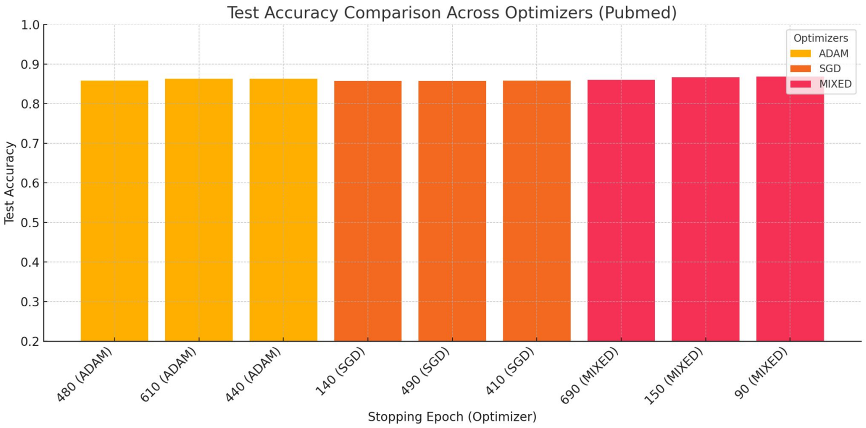


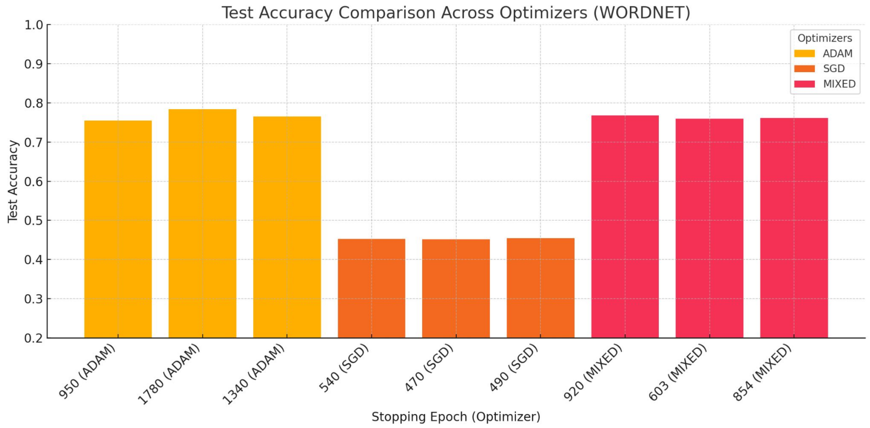
Comparison of Optimizers

Final Test Accuracies by Optimizer (3 Runs per Experiment)

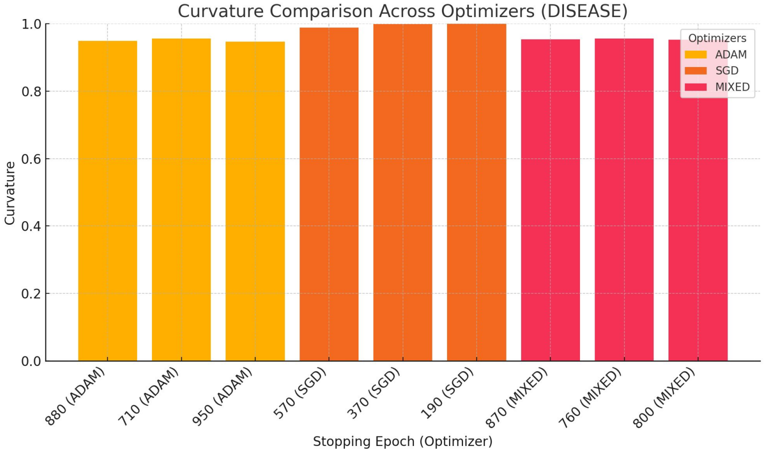


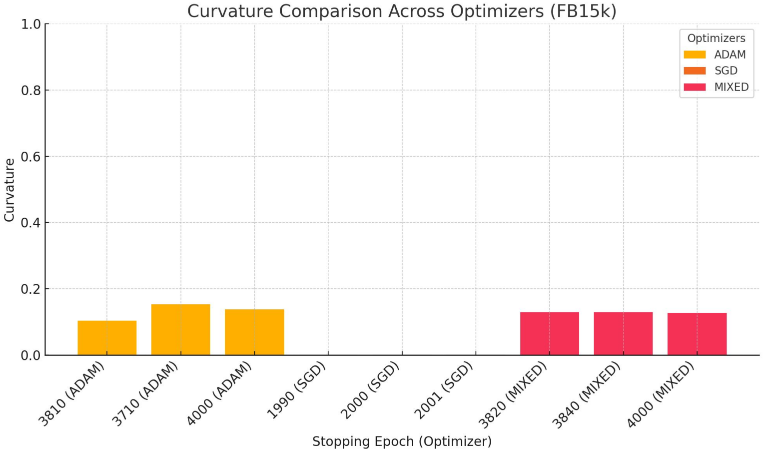


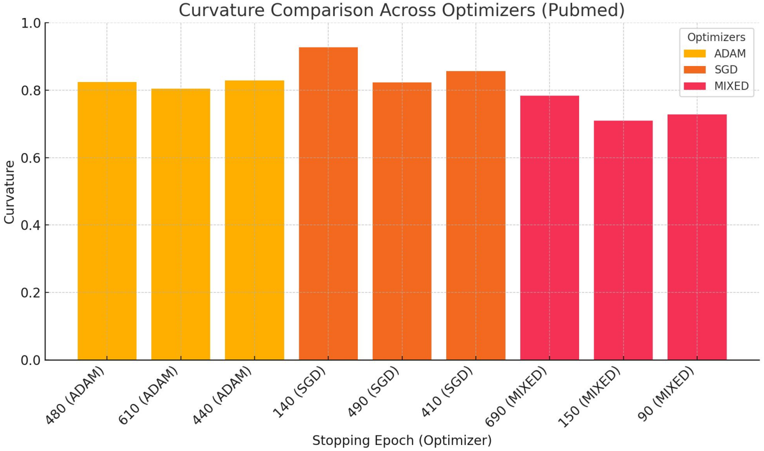


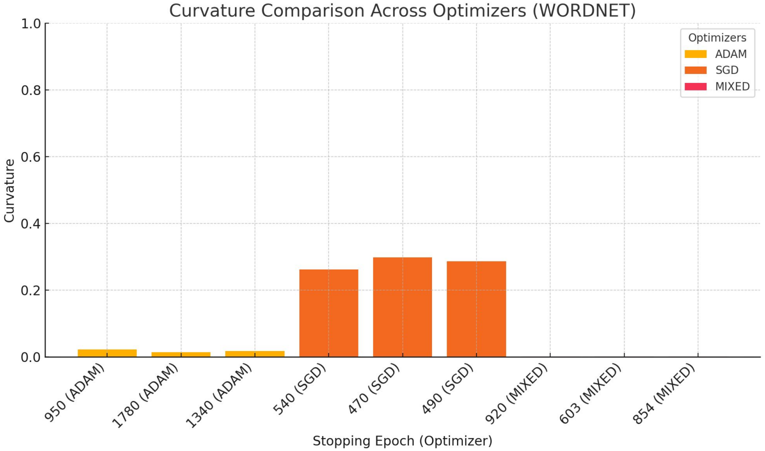


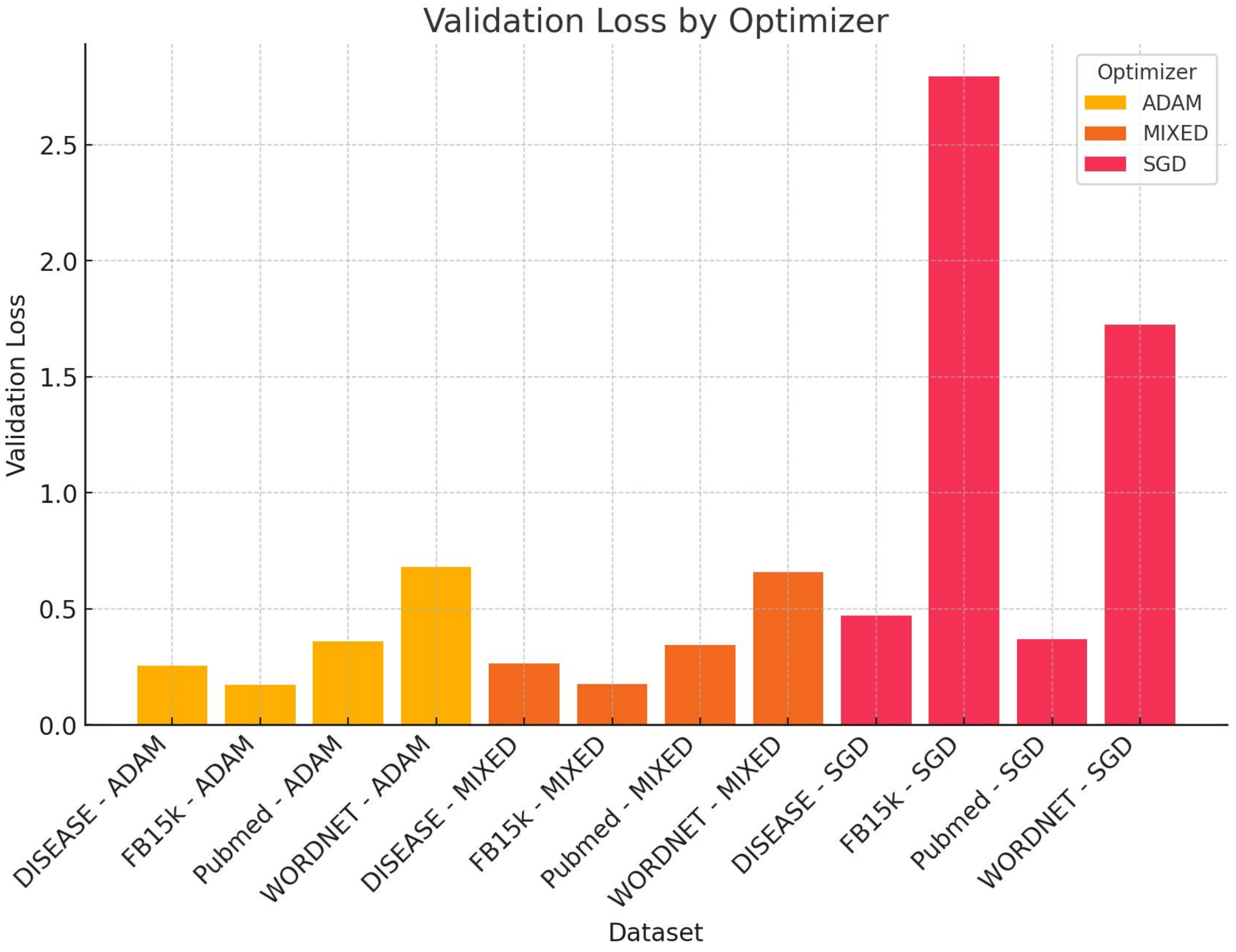
Final Learned Curvatures by Optimizer (3 Runs per Experiment)





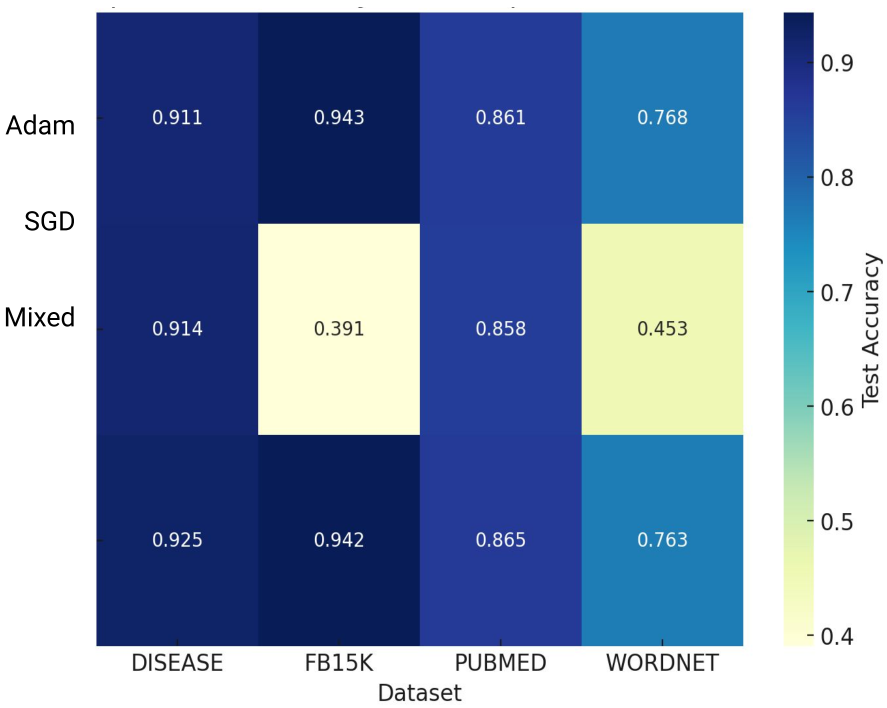




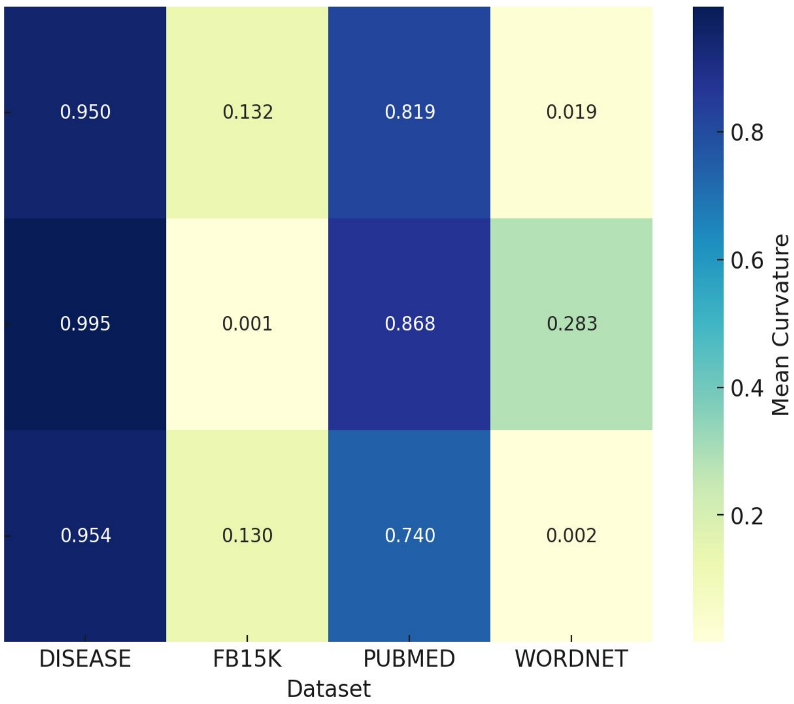

Validation Loss by Optimizer
Optimizer
ADAM
MIXED
SGD
Validation Loss
2.5
2.0
1.5
1.0
0.5
0.0
DISEASE - ADAM
FB15k - ADAM
Pubmed - ADAM
WORDNET - ADAM
DISEASE - MIXED
FB15k - MIXED
Pubmed - MIXED
WORDNET - MIXED
DISEASE - SGD
FB15k - SGD
Pubmed - SGD
WORDNET - SGD
Dataset

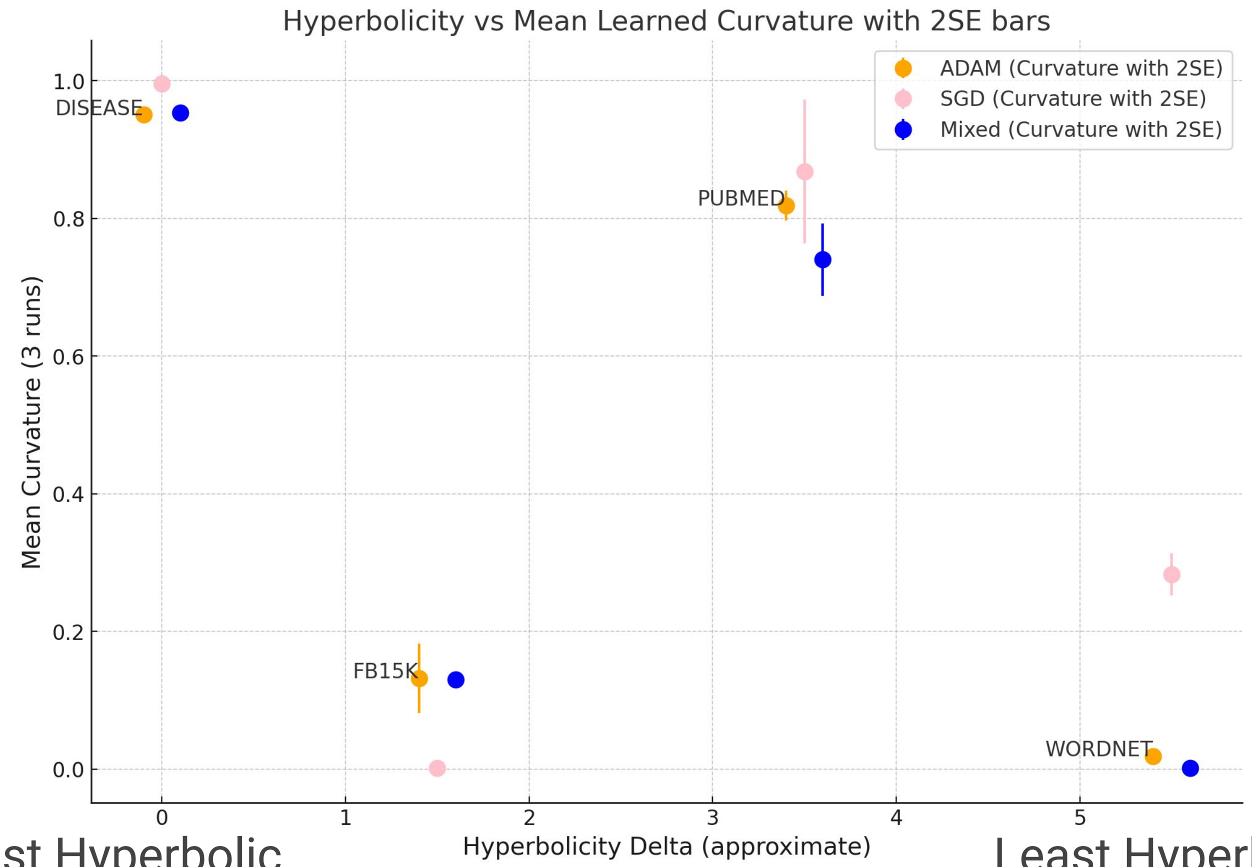
Heatmaps across Datasets and Optimizers

Mean Test Accuracy



Mean Curvature





← Most Hyperbolic Dataset

Least Hyperbolic → Dataset

Discussion

Key Outcomes

Project Results Overview

Enhanced Model Flexibility
Unified global curvature simplifies hyperbolic geometry integration and aligns embedding geometry with graph topology.

Improved Task Performance
HGCN with learnable curvature achieved superior accuracy to baseline GCN in link prediction, for hierarchical datasets.

Optimizer Effectiveness
Riemannian Adam and mixed-precision optimizers delivered the best stability and accuracy compared to SGD.

Curvature and Data Structure

Insights and Case Studies

Insights from δ-Hyperbolicity
Lower δ-hyperbolicity datasets (e.g., Diseases: δ=0) demonstrated high accuracy and curvature stability.

Hierarchical Advantage
Curvature values in hierarchical datasets like Diseases align strongly with graph structure.

Case Study: WordNet Subset vs. Diseases
WordNet Subset (δ=5.5) had minimal curvature (~0.02) with lower accuracy (~76%), compared to Diseases (~0.95 curvature, >91% accuracy).

Optimizer Comparisons

Performance Across Techniques

Riemannian Adam
Best stability and accuracy for hierarchical datasets (e.g., 91.13% for Diseases, 94.34% for FB15K).

Mixed-Precision Optimizer
Balanced computational efficiency with precision, yielding consistently strong performance and consistent learned curvature.

Riemannian SGD
Struggled with noisy datasets; underperformed on FB15K (39.06% accuracy).

Conclusion:

(1) HGCN with curvature optimized globally is superior to a GCN for highly hierarchical data

(2) Hierarchical small δ-hyperbolicity datasets have very negative optimal curvature. "Euclidean-like" datasets are suited to curvatures closer to 0.

(3) The Adam and Mixed Precision Riemannian Optimizers are more effective at determining the optimal curvature than the SGD Riemannian Optimizer, likely due to their adaptive learning rates.