Constant Curvature Graph Convolutional Networks

Gregor Bachmann
ETH Zürich

Gary Bécigneul
MIT

Octavian Ganea
MIT
Overview

• Embeddings of graphs into hyperbolic and spherical space and their products
• Extend gyrovector framework to spherical geometry and provide a unifying formalism
• Introduce graph neural networks producing embeddings in product spaces
• Differentiable transitions in geometry during training in each component
Overview

- Embeddings of graphs into hyperbolic and spherical space and their products
Overview

- Embeddings of graphs into hyperbolic and spherical space and their products
- Extend gyrovector framework to spherical geometry and provide a unifying formalism
Overview

- Embeddings of graphs into **hyperbolic** and **spherical** space and their products
- Extend gyrovector framework to spherical geometry and provide a **unifying formalism**
- Introduce graph neural networks producing embeddings in product spaces
Overview

• Embeddings of graphs into hyperbolic and spherical space and their products

• Extend gyrovector framework to spherical geometry and provide a unifying formalism

• Introduce graph neural networks producing embeddings in product spaces

• Differentiable transitions in geometry during training in each component
Graphs

• Lots of data available in the form of graphs (social networks, railway tracks, phylogenetic trees etc.)

• Node set $V = \{1, \ldots, n\}$ and adjacency matrix $A \in \mathbb{R}^{n \times n}$
Graphs

• Lots of data available in the form of graphs (social networks, railway tracks, phylogenetic trees etc.)
Graphs

• Lots of data available in the form of graphs (social networks, railway tracks, phylogenetic trees etc.)
Graphs

- Lots of data available in the form of **graphs** (social networks, railway tracks, phylogenetic trees etc.)

- Node set \( V = \{1, \ldots, n\} \) and adjacency matrix \( A \in \mathbb{R}^{n \times n} \)
Where to Embed Graphs?

• Euclidean geometry not suitable for many graphs
• Graph distance $d_G(i,j) = \text{"Shortest path from } i \text{ to } j\"$ not respected in Euclidean embedding
• Arbitrary low distortion in spherical and hyperbolic space
Where to Embed Graphs?

- Euclidean geometry not suitable for many graphs
Where to Embed Graphs?

- Euclidean geometry **not** suitable for many graphs
Where to Embed Graphs?

- Euclidean geometry not suitable for many graphs

- Graph distance $d_G(i, j) = ”$Shortest path from $i$ to $j”$ not respected in Euclidean embedding

\[ C_4 \]

\[ K_{1,3} \]
Where to Embed Graphs?

- Euclidean geometry **not** suitable for many graphs

  ![Graphs](image)

  - **Graph distance** $d_G(i, j) = \text{"Shortest path from } i \text{ to } j\text{"}$ not respected in Euclidean embedding

  - Arbitrary low distortion in **spherical** and **hyperbolic** space
Non-Euclidean Geometry
Non-Euclidean Geometry

• Focus on constant sectional curvature manifolds
Non-Euclidean Geometry

• Focus on constant sectional curvature manifolds

• Well-studied in the field of Differential Geometry
Non-Euclidean Geometry

• Focus on constant sectional curvature manifolds

• Well-studied in the field of Differential Geometry

• Computationally attractive expressions for distance, exponential map etc.
Hyperbolic Space as Poincaré Ball

- $H^n = \{ x : ||x||^2 \leq 1/\sqrt{c} \}$ with curvature $-c$
  - equipped with Riemannian tensor $g^c_x = 4(1 - c ||x||^2)^2$.

- Projection of hyperboloid $d^c_H(x, y) = 1/\sqrt{c} \cosh^{-1}(1 + 2c ||x - y||^2 / (1 - ||x||^2)^2 (1 - ||y||^2)^2)$.
Hyperbolic Space as Poincaré Ball

\[ \mathbb{H}^n = \{ x : \| x \|_2 \leq \frac{1}{\sqrt{c}} \} \text{ with curvature } -c \text{ equipped with the Riemannian tensor } \]

\[ g_{x}^{c} = \frac{4}{(1-c\|x\|^{2})^{2}} 1 \]
Hyperbolic Space as Poincaré Ball

• $\mathbb{H}^n = \{ x : \| x \|_2 \leq \frac{1}{\sqrt{c}} \}$ with curvature $-c$ equipped with Riemannian tensor $g^c_x = \frac{4}{(1-c\|x\|^2)^2} 1$

• Projection of hyperboloid
Hyperbolic Space as Poincaré Ball

- $\mathbb{H}^n = \{ x : \|x\|_2 \leq \frac{1}{\sqrt{c}} \}$ with curvature $-c$ equipped with Riemannian tensor $g^c_x = \frac{4}{(1-c\|x\|^2)^2} \mathbb{1}$

- **Projection** of hyperboloid

- $d^c_{\mathbb{H}}(x, y) = \frac{1}{\sqrt{c}} \cosh^{-1} \left( 1 + \frac{2\|x-y\|_2^2}{(\frac{1}{c}-\|x\|_2^2)(\frac{1}{c}-\|y\|_2^2)} \right)$

Heatmap of $d^c_{\mathbb{H}}$

Projection of hyperboloid [4]
Gyroscope Structure

- Next best thing to a vector space
  - Vector addition \( x + y \mapsto \rightarrow x \oplus c y \)
  - Scalar multiplication \( r x \mapsto \rightarrow r \otimes c x \)
- Geodesic \( \gamma \) \( x \rightarrow y \) \((t) = x \oplus c \left( t \otimes c ( -x \oplus c y ) \right) \)
Gyrospace Structure

- Next best thing to a vector space
Gyrospace Structure

- Next best thing to a vector space

- Vector addition $x + y \mapsto x \oplus_c y$
Gyrospace Structure

- Next best thing to a vector space
- Vector addition $x + y \mapsto x \oplus_c y$
- Scalar multiplication $r x \mapsto r \otimes_c x$
Gyrospace Structure

• Next best thing to a vector space

• Vector addition $\mathbf{x} + \mathbf{y} \mapsto \mathbf{x} \oplus_c \mathbf{y}$

• Scalar multiplication $r\mathbf{x} \mapsto r \otimes_c \mathbf{x}$

• Geodesic $\gamma_{\mathbf{x} \rightarrow \mathbf{y}}(t) = \mathbf{x} \oplus_c (t \otimes_c (-\mathbf{x} \oplus_c \mathbf{y}))$
Spherical Space as Stereographic Projection

• $S^{d+1} \sim R^d \times \mathbb{C}$ where $g(x) = 4(1 + ||x||^2)^{1/2}$

• $d(x, y) = \frac{1}{\sqrt{c}} \cos^{-1} \left( \frac{1 + 2c||x - y||^2}{2c(1 + ||x||^2)^{1/2}(1 + ||y||^2)^{1/2}} \right)$
Spherical Space as Stereographic Projection

- Stereographic projection of $\mathbb{S}^{d+1} \cong \mathbb{R}^d + g_x^c$ where
  
  $$g_x^c = \frac{4}{(1 + c ||x||^2)^2} 1$$
Spherical Space as Stereographic Projection

- Stereographic projection of $\mathbb{S}^{d+1} \cong \mathbb{R}^d + g^c_x$ where
  \[ g^c_x = \frac{4}{(1+c\|x\|^2)^2} \]

- $d^c_{\mathbb{S}}(x, y) = \frac{1}{\sqrt{c}} \cos^{-1} \left( 1 + \frac{\frac{2}{c} \|x-y\|^2}{\left(\frac{1}{c} + \|x\|^2\right)\left(\frac{1}{c} + \|y\|^2\right)} \right)$
Spherical Space as Stereographic Projection

• Stereographic projection of $\mathbb{S}^{d+1} \cong \mathbb{R}^d + g^c_x$ where
  
  \[ g^c_x = \frac{4}{(1+c||x||^2)^2} \mathbb{1} \]

• $d^c_{\mathbb{S}}(x, y) = \frac{1}{\sqrt{c}} \cos^{-1}\left( 1 + \frac{\frac{2}{c}||x-y||^2}{\left(\frac{1}{c}+||x||^2\right)\left(\frac{1}{c}+||y||^2\right)} \right)$
Our Contributions: **1) Unified Formalism**
Our Contributions: 1) Unified Formalism

• $\kappa$-stereographic model for any $\kappa \in \mathbb{R}$:

$$\mathcal{S}_d^{\kappa} = \{ x \in \mathbb{R}^d | -\kappa \| x \|^2_2 < 1 \}$$
Our Contributions: 1) Unified Formalism

- $\kappa$-stereographic model for any $\kappa \in \mathbb{R}$:

$$\mathcal{S}_{\kappa}^d = \{ \mathbf{x} \in \mathbb{R}^d \mid -\kappa \|\mathbf{x}\|_2^2 < 1 \}$$

<table>
<thead>
<tr>
<th>$\mathbf{x} \oplus_{\kappa} \mathbf{y}$</th>
<th>$\mathbf{x} + \mathbf{y}$</th>
<th>$\mathcal{S}_{\kappa}^d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{x} \oplus_{\kappa} \mathbf{y}$</td>
<td>$\mathbf{x} + \mathbf{y}$</td>
<td>$(1 - 2\kappa \mathbf{x}^T \mathbf{y} - \kappa |\mathbf{y}|^2)\mathbf{x} + (1 + \kappa |\mathbf{x}|^2)\mathbf{y}$</td>
</tr>
<tr>
<td>$r \otimes_{\kappa} \mathbf{x}$</td>
<td>$r\mathbf{x}$</td>
<td>$\tan_{\kappa} (r \cdot \tan_{\kappa}^{-1} |\mathbf{x}|) \frac{\mathbf{x}}{|\mathbf{x}|}$</td>
</tr>
<tr>
<td>$\gamma_{\mathbf{x} \rightarrow \mathbf{y}}(t)$</td>
<td>$\mathbf{x} + t(\mathbf{y} - \mathbf{x})$</td>
<td>$\mathbf{x} \oplus_{\kappa} (t \otimes_{\kappa} (-\mathbf{x} \oplus_{\kappa} \mathbf{y}))$</td>
</tr>
</tbody>
</table>
Our Contributions: 1) Unified Formalism

- \( \kappa \)-stereographic model for any \( \kappa \in \mathbb{R} \):

\[
\mathcal{S}_\kappa = \{ x \in \mathbb{R}^d \mid -\kappa \| x \|_2^2 < 1 \}
\]

<table>
<thead>
<tr>
<th>( x \oplus_\kappa y )</th>
<th>( x + y )</th>
<th>( \mathcal{S}_\kappa )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \oplus_\kappa y )</td>
<td>( x + y )</td>
<td>( \frac{(1-2\kappa x^T y - \kappa | y |_2^2)x + (1+\kappa | x |_2^2)y}{1-2\kappa x^T y + \kappa^2 | x |_2^2 | y |_2^2} )</td>
</tr>
<tr>
<td>( r \otimes_\kappa x )</td>
<td>( rx )</td>
<td>( \tan_\kappa (r \cdot \tan_\kappa^{-1} | x |) \frac{x}{| x |} )</td>
</tr>
<tr>
<td>( \gamma_{x \rightarrow y}(t) )</td>
<td>( x + t(y - x) )</td>
<td>( x \oplus_\kappa (t \otimes_\kappa (-x \oplus_\kappa y)) )</td>
</tr>
</tbody>
</table>

- More unifying expressions for distance, exponential map etc. in our paper!
Our Contributions: 2) Matrix Multiplications

• Embeddings $X$ where $X_i \cdot \in \text{std}_\kappa, W \in \mathbb{R}^d \times k$ and $A \in \mathbb{R}^{n \times n}$

• Right matrix multiplication $XW$ acts on columns $X_i \cdot$

• Thus lift to tangent space at zero: $(X \otimes \kappa W)_i \cdot = \exp \kappa_0 ((\log \kappa_0 (X) W)_i \cdot)$

• Introduced in [2], we extended it to spherical spaces
Our Contributions: **2) Matrix Multiplications**

- Embeddings $X$ where $X_i \in \mathfrak{so}_d$, $W \in \mathbb{R}^{d \times k}$ and $A \in \mathbb{R}^{n \times n}$
Our Contributions: 2) Matrix Multiplications

- Embeddings $X$ where $X_{i\cdot} \in St^d_{\kappa}$, $W \in \mathbb{R}^{d \times k}$ and $A \in \mathbb{R}^{n \times n}$

- Right matrix multiplication $XW$ acts on columns $X_{\cdot i}$

Thus lift to tangent space at zero:

$$(X \otimes_{\kappa} W)_{i\cdot} = \exp_{0}^{\kappa}((\log_{0}^{\kappa}(X)W)_{i\cdot})$$
Our Contributions: 2) Matrix Multiplications

- Embeddings $X$ where $X_{i \bullet} \in \mathfrak{st}^d_{\kappa}$, $W \in \mathbb{R}^{d \times k}$ and $A \in \mathbb{R}^{n \times n}$

- **Right** matrix multiplication $XW$ acts on columns $X_{\bullet i}$
  
  Thus lift to tangent space at zero:
  
  $$(X \otimes_{\kappa} W)_{i \bullet} = \exp_{0}^{\kappa} ((\log_{0}^{\kappa}(X) W)_{i \bullet})$$

- Introduced in [2], we extended it to spherical spaces
Our Contributions: 2) Matrix Multiplications
Our Contributions: 2) Matrix Multiplications

- **Left** matrix multiplication $AX$ acts on rows $X_i$:

  $$(AX)_{i'} = A_{i1}X_{1'} + \cdots + A_{in}X_{n'}$$
Our Contributions: 2) Matrix Multiplications

• **Left** matrix multiplication $AX$ acts on rows $X_i$:  
  
  $$(AX)_i = A_{i1}X_1 + \cdots + A_{in}X_n$$

• **Idea:** Reduce problem of linear combination to definition of a non-euclidean midpoint
Our Contributions: 2) Matrix Multiplications

- **Left** matrix multiplication $AX$ acts on rows $X_i$: 

  $$(AX)_i = A_{i1}X_1 + \cdots + A_{in}X_n.$$ 

- **Idea:** Reduce problem of linear combination to definition of a non-euclidean *midpoint*.
Our Contributions: 2) Matrix Multiplications

• Leverage gyromidpoint for hyperbolic space and extend it to standard $\kappa$:

$$m_{\kappa}(x_1, \ldots, x_n; \alpha) = \frac{1}{2} \otimes m_{\kappa}(\sum_{i=1}^{n} \alpha_i \lambda_{\kappa} x_i \sum_{j=1}^{n} \alpha_j (\lambda_{\kappa} x_j - 1) x_i)$$

• Define left matrix multiplication row-wise:

$$(A \otimes_{\kappa} X)_{i•} := \left( \sum_{j} A_{ij} \right) \otimes m_{\kappa}(X_{1•}, \ldots, X_{n•}; A_{i•})$$

• Same scaling behaviour:

$$d_{\kappa}(0, r \otimes_{\kappa} x) = r \cdot d_{\kappa}(0, x)$$
Our Contributions: 2) Matrix Multiplications

- Leverage gyromidpoint for hyperbolic space and extend it to $\mathbb{H}^d$:

$$m_\kappa(x_1, \cdots, x_n; \alpha) = \frac{1}{2} \otimes_\kappa \left( \sum_{i=1}^{n} \alpha_i \lambda_{x_i}^\kappa \frac{1}{\sum_{j=1}^{n} \alpha_j (\lambda_{x_j}^\kappa - 1)} x_i \right)$$
Our Contributions: 2) Matrix Multiplications

- Leverage \textbf{gyromidpoint} for hyperbolic space and extend it to \( \mathcal{S} t^d \):

\[
m_\kappa(x_1, \cdots, x_n; \alpha) = \frac{1}{2} \otimes_\kappa \left( \sum_{i=1}^{n} \frac{\alpha_i \lambda_{x_i}^\kappa}{\sum_{j=1}^{n} \alpha_j (\lambda_{x_j}^\kappa - 1)} x_i \right)
\]

- Define left matrix multiplication row-wise:

\[
(A \boxdot_\kappa X)_{i\bullet} := (\sum_j A_{ij}) \otimes_\kappa m_\kappa(X_{1\bullet}, \cdots, X_{n\bullet}; A_{i\bullet})
\]
Our Contributions: 2) Matrix Multiplications

- Leverage gyromidpoint for hyperbolic space and extend it to $s_t^d$: 

$$m_{\kappa}(x_1, \cdots, x_n; \alpha) = \frac{1}{2} \otimes_{\kappa} \left( \sum_{i=1}^{n} \frac{\alpha_i \lambda^{\kappa}_{x_i}}{\sum_{j=1}^{n} \alpha_j (\lambda^{\kappa}_{x_j} - 1)} x_i \right)$$

- Define left matrix multiplication row-wise:

$$(A \boxtimes_{\kappa} X)_{i.} := (\sum_j A_{ij}) \otimes_{\kappa} m_{\kappa}(X_{1.}, \cdots, X_{n.}; A_{i.})$$

- Same scaling behaviour: $d_{\kappa}(0, r \otimes_{\kappa} x) = r \cdot d_{\kappa}(0, x)$
Gyromidpoint for Varying Curvature

\[ \kappa = -1.909 \]
Our Contributions: 3) Differentiable Interpolation
Our Contributions: 3) Differentiable Interpolation

• All quantities recover their Euclidean counterpart for $\kappa \to 0^\pm$
Our Contributions: **3) Differentiable Interpolation**

- All quantities *recover* their Euclidean counterpart for $\kappa \to 0^\pm$
- We proved an even *stronger* result:
Our Contributions: 3) Differentiable Interpolation

- All quantities recover their Euclidean counterpart for $\kappa \rightarrow 0^{\pm}$
- We proved an even stronger result:

**Differentiability of $s^d_{t,\kappa}$ w.r.t. $\kappa$ around 0**

The first order derivatives at $0^-$ and $0^+$ w.r.t. to $\kappa$ of all the mentioned quantities exist and are equal.
Our Contributions: 3) Differentiable Interpolation

- All quantities **recover** their Euclidean counterpart for $\kappa \to 0^{\pm}$

- We proved an even **stronger** result:

  **Differentiability of $\mathcal{S}_{\kappa}^d$ w.r.t. $\kappa$ around 0**

  The first order derivatives at $0^-$ and $0^+$ w.r.t. to $\kappa$ of all the mentioned quantities **exist** and are **equal**.

- Enables **learning the curvature** $\kappa$ with gradient descent with a differentiable change of sign
Our Contributions: 4) Constant Curvature GCN
Our Contributions: 4) Constant Curvature GCN

• Given graph $G = (V, A, X)$ where $V = \{1, \ldots, n\}$, adjacency $A \in \mathbb{R}^{n \times n}$ and node-level features $X \in \mathbb{R}^{n \times d}$
Our Contributions: 4) Constant Curvature GCN

• Given graph $G = (V, A, X)$ where $V = \{1, \ldots, n\}$, adjacency $A \in \mathbb{R}^{n \times n}$ and node-level features $X \in \mathbb{R}^{n \times d}$

• Graph neural networks are a very popular class of models for inference on graphs
Our Contributions: 4) Constant Curvature GCN

• Given graph $G = (V, A, X)$ where $V = \{1, \ldots, n\}$, adjacency $A \in \mathbb{R}^{n \times n}$ and node-level features $X \in \mathbb{R}^{n \times d}$

• Graph neural networks are a very popular class of models for inference on graphs

• We extend the vanilla GCN [3]:

$$H^{(t+1)} = \sigma \left( \hat{A}H^{(t)}W^{(t)} \right)$$

for some non-linearity $\sigma$, $\hat{A} = \tilde{D}^{-\frac{1}{2}} (A + 1) \tilde{D}^{-\frac{1}{2}}$, $\tilde{D}_{ii} = \sum_k \tilde{A}_{ik}$ and trainable parameters $W^{(l)}$
Our Contributions: 4) Constant Curvature GCN
Our Contributions: 4) Constant Curvature GCN

- Turn it non-euclidean:

  \[ H^{(l+1)} = \sigma^{\otimes \kappa} \left( \hat{A} \boxtimes \kappa \left( H^{(l)} \otimes \kappa W^{(l)} \right) \right) \]

  where \( \sigma^{\otimes \kappa} \) is the \( \kappa \)-stereographic version of \( \sigma \) (see paper)
Our Contributions: 4) Constant Curvature GCN

• Turn it non-euclidean:

\[ H^{(l+1)} = \sigma^\otimes_\kappa \left( \hat{A} \boxtimes_\kappa \left( H^{(l)} \otimes_\kappa W^{(l)} \right) \right) \]

where \( \sigma^\otimes_\kappa \) is the \( \kappa \)-stereographic version of \( \sigma \) (see paper)

• Learn the curvature to adapt to the geometry of the data
Our Contributions: 4) Constant Curvature GCN

• Turn it non-euclidean:

\[ H^{(l+1)} = \sigma \otimes \kappa \left( \hat{A} \otimes_{\kappa} \left( H^{(l)} \otimes_{\kappa} W^{(l)} \right) \right) \]

where \( \sigma \otimes \kappa \) is the \( \kappa \)-stereographic version of \( \sigma \) (see paper)

• Learn the curvature to adapt to the geometry of the data

• Allows for differentiable transitions in the geometry during training
Our Contributions: 5) Product GCN
Our Contributions: 5) Product GCN

- We can take it one step further: Embed in product space

\[ st_{\kappa_1}^d \times \cdots \times st_{\kappa_m}^d \]
Our Contributions: **5) Product GCN**

- We can take it one step further: Embed in **product space**

\[ \mathcal{ST}^d_{\kappa_1} \times \cdots \times \mathcal{ST}^d_{\kappa_m} \]
Our Contributions: 5) Product GCN

- We can take it one step further: Embed in product space

\[ \mathbb{S}^d \times \cdots \times \mathbb{S}^d \]

- Again we find a gyrovector space structure
Our Contributions: **5) Product GCN**

- We can take it one step further: Embed in **product space**

\[ \mathbb{S}^d_{k_1} \times \cdots \times \mathbb{S}^d_{k_m} \]

- Again we find a **gyrovector space** structure

- The **operations** extend component-wise while still preserving the desired properties
Experiments: Distortion Task

Minimize the discrepancy between embedding distances and graph distances

\[
L(x_1, \ldots, x_n) = \frac{1}{n^2} \sum_{i,j \neq i} (d_{\kappa}(x_i, x_j) - d_G(i, j))^2 - 1^2
\]

Train \(\kappa\)-GCN on three synthetic datasets, tree (negative curvature), spherical graph (positive curvature) and toroidal graph (product of positive curvature)

<table>
<thead>
<tr>
<th>Model</th>
<th>Tree</th>
<th>Toroidal</th>
<th>Spherical</th>
</tr>
</thead>
<tbody>
<tr>
<td>(E_{10}(\text{GCN}))</td>
<td>0.0502</td>
<td>0.0603</td>
<td>0.0409</td>
</tr>
<tr>
<td>(H_{10}(\kappa\text{-GCN}))</td>
<td>0.0029</td>
<td>0.272</td>
<td>0.267</td>
</tr>
<tr>
<td>(S_{10}(\kappa\text{-GCN}))</td>
<td>0.473</td>
<td>0.0485</td>
<td>0.0337</td>
</tr>
<tr>
<td>(H_{5} \times H_{5}(\kappa\text{-GCN}))</td>
<td>0.0048</td>
<td>0.112</td>
<td>0.152</td>
</tr>
<tr>
<td>(S_{5} \times S_{5}(\kappa\text{-GCN}))</td>
<td>0.51</td>
<td>0.0464</td>
<td>0.0359</td>
</tr>
</tbody>
</table>
Experiments: Distortion Task

- **Minimize** the discrepancy between embedding distances and graph distances

\[
L(x_1, \ldots, x_n) = \frac{1}{n^2} \sum_{i,j} \left( \left( \frac{d_{\kappa}(x_i, x_j)}{d_{G(i,j)}} \right)^2 - 1 \right)^2
\]
Experiments: Distortion Task

- **Minimize** the discrepancy between embedding distances and graph distances

\[
L(x_1, \ldots, x_n) = \frac{1}{n^2} \sum_{i,j} \left( \left( \frac{d_\kappa(x_i, x_j)}{d_G(i,j)} \right)^2 - 1 \right)^2
\]

- Train $\kappa$-GCN on three syntethic datasets, **tree** (negative curvature), **spherical** graph (positive curvature) and **toroidal** graph (product of positive curvature)
Experiments: Distortion Task

- **Minimize** the discrepancy between embedding distances and graph distances

\[
L(x_1, \ldots, x_n) = \frac{1}{n^2} \sum_{i,j} \left( \left( \frac{d_\kappa(x_i, x_j)}{d_G(i,j)} \right)^2 - 1 \right)^2
\]

- Train \( \kappa \)-GCN on three syntethic datasets, tree (negative curvature), spherical graph (positive curvature) and toroidal graph (product of positive curvature)

<table>
<thead>
<tr>
<th>Model</th>
<th>Tree</th>
<th>Toroidal</th>
<th>Spherical</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E^{10} ) (GCN)</td>
<td>0.0502</td>
<td>0.0603</td>
<td>0.0409</td>
</tr>
<tr>
<td>( H^{10} ) (( \kappa )-GCN)</td>
<td><strong>0.0029</strong></td>
<td>0.272</td>
<td>0.267</td>
</tr>
<tr>
<td>( S^{10} ) (( \kappa )-GCN)</td>
<td>0.473</td>
<td>0.0485</td>
<td><strong>0.0337</strong></td>
</tr>
<tr>
<td>( H^5 \times H^5 ) (( \kappa )-GCN)</td>
<td>0.0048</td>
<td>0.112</td>
<td>0.152</td>
</tr>
<tr>
<td>( S^5 \times S^5 ) (( \kappa )-GCN)</td>
<td>0.51</td>
<td><strong>0.0464</strong></td>
<td>0.0359</td>
</tr>
</tbody>
</table>
Experiments: Node Classification

- Evaluate on four real-world datasets
Experiments: Node Classification

• Evaluate on four real-world datasets
• Report mean accuracy across 5 splits and 5 runs each
Experiments: Node Classification

- Evaluate on four **real-world** datasets
- Report mean **accuracy** across 5 splits and 5 runs each

<table>
<thead>
<tr>
<th>Model</th>
<th>Citeeseer</th>
<th>Cora</th>
<th>Pubmed</th>
<th>Airport</th>
</tr>
</thead>
<tbody>
<tr>
<td>E$^{16}$ [3]</td>
<td>72.9 ± 0.54</td>
<td>81.4 ± 0.4</td>
<td>79.2 ± 0.39</td>
<td>81.4 ± 0.29</td>
</tr>
<tr>
<td>H$^{16}$ [1]</td>
<td>71 ± 0.49</td>
<td>80.3 ± 0.46</td>
<td>79.8 ± 0.43</td>
<td>84.4 ± 0.41</td>
</tr>
<tr>
<td>H$^{16}$ ($\kappa$-GCN)</td>
<td><strong>73.2 ± 0.51</strong></td>
<td>81.2 ± 0.5</td>
<td>78.5 ± 0.36</td>
<td>81.9 ± 0.33</td>
</tr>
<tr>
<td>S$^{16}$ ($\kappa$-GCN)</td>
<td>72.1 ± 0.45</td>
<td><strong>81.9 ± 0.45</strong></td>
<td>78.8 ± 0.49</td>
<td>80.9 ± 0.58</td>
</tr>
<tr>
<td>Prod-GCN</td>
<td>71.1 ± 0.59</td>
<td>80.8 ± 0.41</td>
<td>78.1 ± 0.6</td>
<td>81.7 ± 0.44</td>
</tr>
</tbody>
</table>
THANK YOU!

Check out our website hyperbolicdeeplearning.com

HYPERBOLIC DEEP LEARNING
References


