

Hyperbolic Neural Networks

刘闯

chuangliu@whu.edu.cn

2019.03.08



Reference

1. **Hyperbolic Neural Networks**, ETH Zürich , 2018 NIPS, **Octavian-Eugen Ganea , Thomas Hofmann**
2. **Hyperbolic Graph Neural Networks**, **Facebook AI**, 2019 NIPS , **Maximilian Nickel and Douwe Kiela**
3. **Hyperbolic Graph Convolutional Neural Networks**, **Stanford snap** , NIPS 2019 , **Jure Leskovec**
4. **Hyperbolic Attention Networks**, **DeepMind**, 2019 ICLR , **Caglar Gulcehre**
5. **Hyperbolic Graph Attention Network**, BUPT, 2020 AAAI, **Chuan Shi**



HNN

Hyperbolic Neural Networks

Generalize deep neural models to non-Euclidean domain

1. 双曲空间的表征能力在很多方面不如欧氏空间， 主要是缺少相应的神经网络层数。
2. 本文通过将莫比乌斯陀螺向量空间的形式性与双曲空间的庞加莱模型的黎曼几何相结合，将深度学习推广到双曲空间。



Gyrovector spaces

Möbius addition. The *Möbius addition* of x and y in \mathbb{D}_c^n is defined as

$$x \oplus_c y := \frac{(1 + 2c\langle x, y \rangle + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c\langle x, y \rangle + c^2\|x\|^2\|y\|^2}. \quad (4)$$

In particular, when $c = 0$, one recovers the Euclidean addition of two vectors in \mathbb{R}^n . Note that without loss of generality, the case $c > 0$ can be reduced to $c = 1$. Unless stated otherwise, we will use \oplus as \oplus_1 to simplify notations. For general $c > 0$, this operation is not commutative nor associative. However, it satisfies $x \oplus_c \mathbf{0} = \mathbf{0} \oplus_c x = x$. Moreover, for any $x, y \in \mathbb{D}_c^n$, we have $(-x) \oplus_c x = x \oplus_c (-x) = \mathbf{0}$ and $(-x) \oplus_c (x \oplus_c y) = y$ (left-cancellation law). The *Möbius subtraction* is then defined by the use of the following notation: $x \ominus_c y := x \oplus_c (-y)$. See [29, section 2.1] for a geometric interpretation of the Möbius addition.

Möbius scalar multiplication. For $c > 0$, the *Möbius scalar multiplication* of $x \in \mathbb{D}_c^n \setminus \{\mathbf{0}\}$ by $r \in \mathbb{R}$ is defined as

$$r \otimes_c x := (1/\sqrt{c}) \tanh(r \tanh^{-1}(\sqrt{c}\|x\|)) \frac{x}{\|x\|}, \quad (5)$$

and $r \otimes_c \mathbf{0} := \mathbf{0}$. Note that similarly as for the Möbius addition, one recovers the Euclidean scalar multiplication when c goes to zero: $\lim_{c \rightarrow 0} r \otimes_c x = rx$. This operation satisfies desirable properties such as $n \otimes_c x = x \oplus_c \dots \oplus_c x$ (n additions), $(r + r') \otimes_c x = r \otimes_c x \oplus_c r' \otimes_c x$ (scalar distributivity³), $(rr') \otimes_c x = r \otimes_c (r' \otimes_c x)$ (scalar associativity) and $|r| \otimes_c x / \|r \otimes_c x\| = x / \|x\|$ (scaling property).

Distance. If one defines the generalized hyperbolic metric tensor g^c as the metric conformal to the Euclidean one, with conformal factor $\lambda_x^c := 2/(1 - c\|x\|^2)$, then the induced distance function on (\mathbb{D}_c^n, g^c) is given by⁴

$$d_c(x, y) = (2/\sqrt{c}) \tanh^{-1}(\sqrt{c}\| -x \oplus_c y \|). \quad (6)$$

Again, observe that $\lim_{c \rightarrow 0} d_c(x, y) = 2\|x - y\|$, i.e. we recover Euclidean geometry in the limit⁵. Moreover, for $c = 1$ we recover $d_{\mathbb{D}}$ of Eq. (2).

C = 0 退化到欧氏空间

欧氏空间中存在着很多的向量运算操作：
向量加减，数乘，

Gyrovector spaces 类似于欧氏中的向量空间为双曲空间提供一种代数形式规范



Hyperbolic Neural Networks

神经网络是由一系列基本操作组合而成

Lemma 2. For any point $x \in \mathbb{D}_c^n$, the exponential map $\exp_x^c : T_x \mathbb{D}_c^n \rightarrow \mathbb{D}_c^n$ and the logarithmic map $\log_x^c : \mathbb{D}_c^n \rightarrow T_x \mathbb{D}_c^n$ are given for $v \neq \mathbf{0}$ and $y \neq x$ by:

$$\exp_x^c(v) = x \oplus_c \left(\tanh \left(\sqrt{c} \frac{\lambda_x^c \|v\|}{2} \right) \frac{v}{\sqrt{c} \|v\|} \right), \quad \log_x^c(y) = \frac{2}{\sqrt{c} \lambda_x^c} \tanh^{-1}(\sqrt{c} \| -x \oplus_c y \|) \frac{-x \oplus_c y}{\| -x \oplus_c y \|}. \quad (10)$$

Exponential map : $T_x \mathbb{D}_c^n \rightarrow \mathbb{D}_c^n$ 切平面 \rightarrow 双曲面

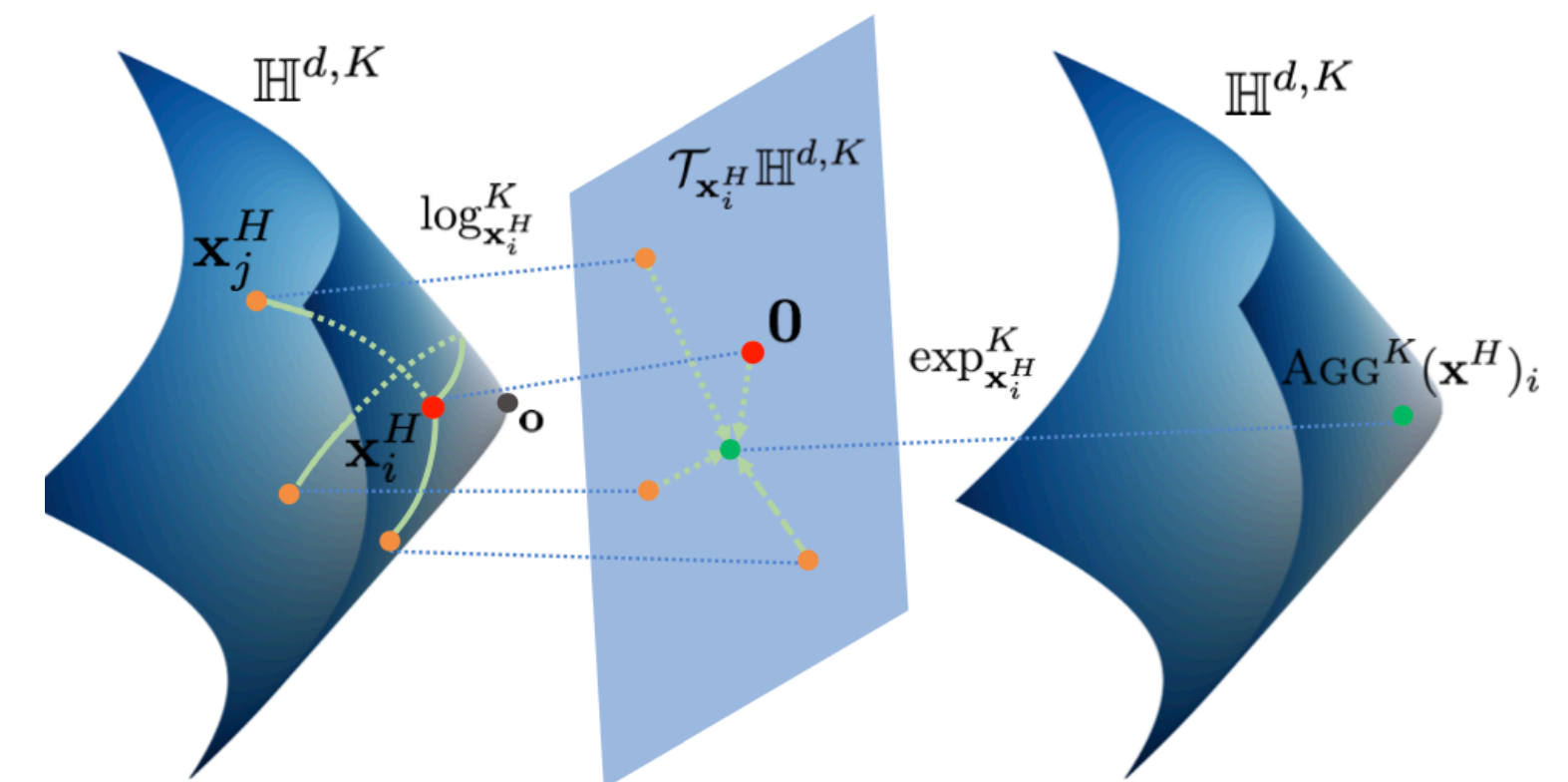
Logarithmic map : $\mathbb{D}_c^n \rightarrow T_x \mathbb{D}_c^n$

前馈神经网络

Definition 3.2 (Möbius version). For $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, we define the Möbius version of f as the map from \mathbb{D}_c^n to \mathbb{D}_c^m by:

$$f^{\otimes_c}(x) := \exp_{\mathbf{0}}^c(f(\log_{\mathbf{0}}^c(x))), \quad (24)$$

where $\exp_{\mathbf{0}}^c : T_{\mathbf{0}_m} \mathbb{D}_c^m \rightarrow \mathbb{D}_c^m$ and $\log_{\mathbf{0}}^c : \mathbb{D}_c^n \rightarrow T_{\mathbf{0}_n} \mathbb{D}_c^n$.



切平面是一个线性空间

先映射到切平面，执行 FNN，再映射回双

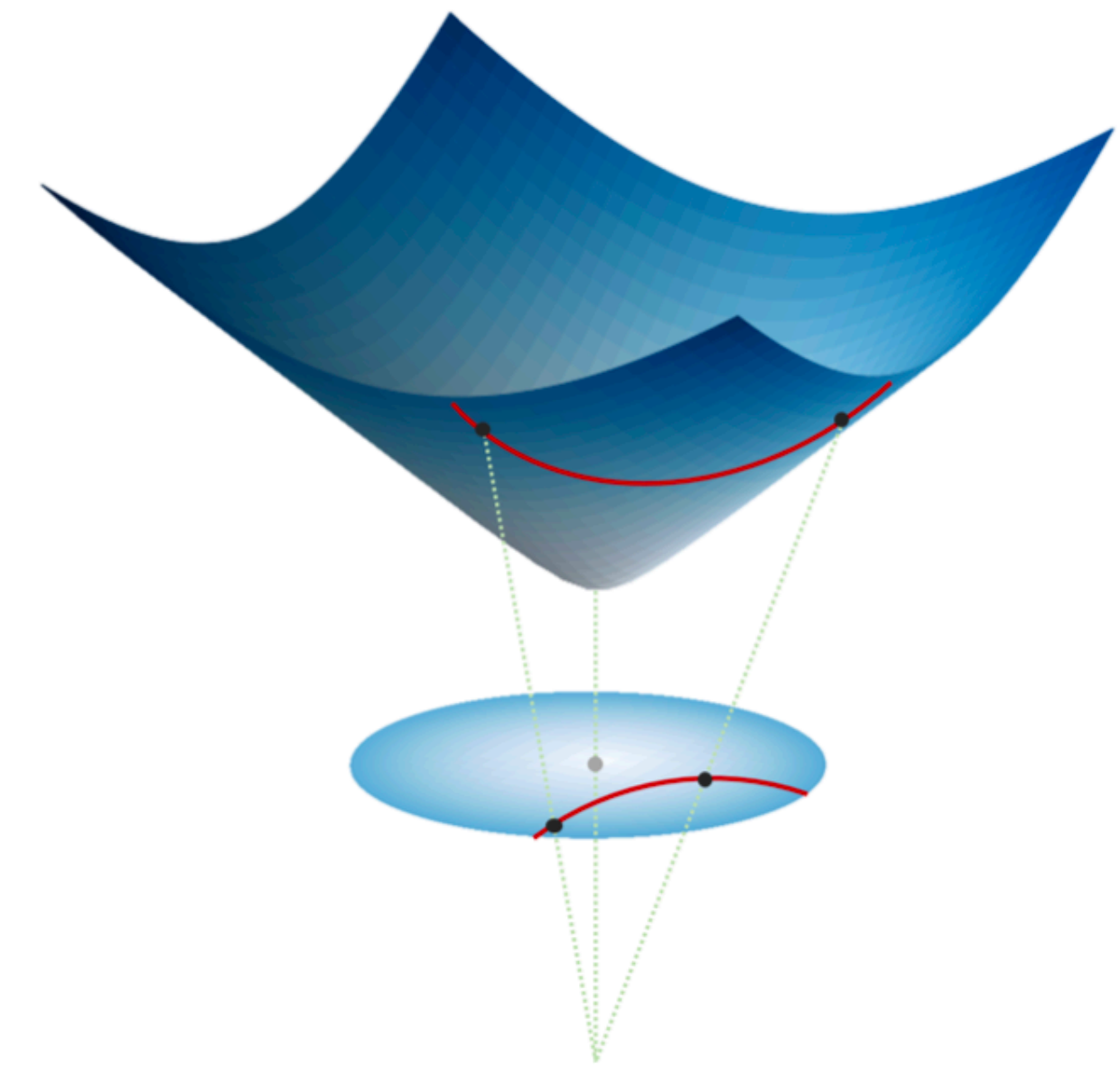


HGCN

Hyperbolic Graph Convolutional Neural Networks

图卷积和双曲神经网络结合

1. 应用双曲面模型
2. 在双曲面上进行 GCN(GAT)
3. Trainable curvature





Hyperboloid manifold

双曲面是双曲线的推广

simplicity and numerical stability

$$y_1^2 + y_2^2 + y_3^2 - y_4^2 = -1$$

$$\mathbb{H}^{d,K} := \{ \mathbf{x} \in \mathbb{R}^{d+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -K, x_0 > 0 \} \quad \mathcal{T}_{\mathbf{x}} \mathbb{H}^{d,K} := \{ \mathbf{v} \in \mathbb{R}^{d+1} : \langle \mathbf{v}, \mathbf{x} \rangle_{\mathcal{L}} = 0 \}$$

双曲空间内积: Minkowski inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} := -x_0 y_0 + x_1 y_1 + \dots + x_d y_d$$

较为接近欧氏内积, 较为稳定

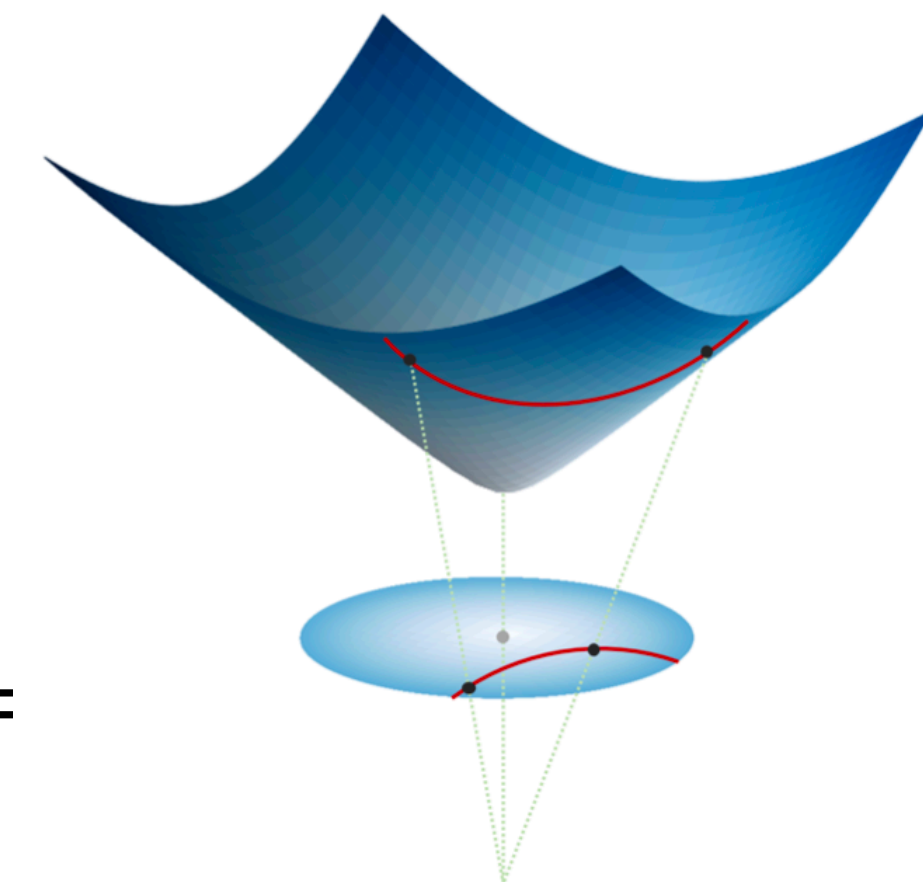
距离和指对映射

Proposition 3.1. Let $\mathbf{x} \in \mathbb{H}^{d,K}$, $\mathbf{u} \in \mathcal{T}_{\mathbf{x}} \mathbb{H}^{d,K}$ be unit-speed. The unique unit-speed geodesic $\gamma_{\mathbf{x} \rightarrow \mathbf{u}}(\cdot)$ such that $\gamma_{\mathbf{x} \rightarrow \mathbf{u}}(0) = \mathbf{x}$, $\dot{\gamma}_{\mathbf{x} \rightarrow \mathbf{u}}(0) = \mathbf{u}$ is $\gamma_{\mathbf{x} \rightarrow \mathbf{u}}^K(t) = \cosh\left(\frac{t}{\sqrt{K}}\right) \mathbf{x} + \sqrt{K} \sinh\left(\frac{t}{\sqrt{K}}\right) \mathbf{u}$, and the intrinsic distance function between two points \mathbf{x}, \mathbf{y} in $\mathbb{H}^{d,K}$ is then:

$$d_{\mathcal{L}}^K(\mathbf{x}, \mathbf{y}) = \sqrt{K} \operatorname{arcosh}(-\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} / K). \quad (4)$$

Proposition 3.2. For $\mathbf{x} \in \mathbb{H}^{d,K}$, $\mathbf{v} \in \mathcal{T}_{\mathbf{x}} \mathbb{H}^{d,K}$ and $\mathbf{y} \in \mathbb{H}^{d,K}$ such that $\mathbf{v} \neq \mathbf{0}$ and $\mathbf{y} \neq \mathbf{x}$, the exponential and logarithmic maps of the hyperboloid model are given by:

$$\exp_{\mathbf{x}}^K(\mathbf{v}) = \cosh\left(\frac{\|\mathbf{v}\|_{\mathcal{L}}}{\sqrt{K}}\right) \mathbf{x} + \sqrt{K} \sinh\left(\frac{\|\mathbf{v}\|_{\mathcal{L}}}{\sqrt{K}}\right) \frac{\mathbf{v}}{\|\mathbf{v}\|_{\mathcal{L}}}, \quad \log_{\mathbf{x}}^K(\mathbf{y}) = d_{\mathcal{L}}^K(\mathbf{x}, \mathbf{y}) \frac{\mathbf{y} + \frac{1}{K} \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} \mathbf{x}}{\|\mathbf{y} + \frac{1}{K} \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} \mathbf{x}\|_{\mathcal{L}}}.$$





HGCN

1. 将欧氏空间特征 map 双曲空间

$$\mathbf{x}^{0,H} = \exp_{\mathbf{o}}^K \left((0, \mathbf{x}^{0,E}) \right) = \left(\sqrt{K} \cosh \left(\frac{\| \mathbf{x}^{0,E} \|_2}{\sqrt{K}} \right), \sqrt{K} \sinh \left(\frac{\| \mathbf{x}^{0,E} \|_2}{\sqrt{K}} \right) \frac{\mathbf{x}^{0,E}}{\| \mathbf{x}^{0,E} \|_2} \right)$$

将 $(0, \mathbf{x}^{0,E})$ 看作切平面的一个点

2. 在双曲空间进行特征变换

leverage the exp and log maps

$$\mathbf{h}_i^{\ell,E} = W^{\ell} \mathbf{x}_i^{\ell-1,E} + \mathbf{b}^{\ell}$$

平移

$$W \otimes^K \mathbf{x}^H := \exp_{\mathbf{o}}^K \left(W \log_{\mathbf{o}}^K (\mathbf{x}^H) \right)$$

$$\mathbf{x}^H \oplus^K \mathbf{b} := \exp_{\mathbf{x}^H}^K \left(P_{\mathbf{o} \rightarrow \mathbf{x}^H}^K (\mathbf{b}) \right)$$

3. Attention based aggregation

$$w_{ij} = \text{SOFTMAX}_{j \in \mathcal{N}(i)} \left(\text{MLP} \left(\log_{\mathbf{o}}^K (\mathbf{x}_i^H) \parallel \log_{\mathbf{o}}^K (\mathbf{x}_j^H) \right) \right)$$

切平面计算权值

$$\text{AGG}^K (\mathbf{x}^H)_i = \exp_{\mathbf{x}_i^H}^K \left(\sum_{j \in \mathcal{N}(i)} w_{ij} \log_{\mathbf{x}_i^H}^K (\mathbf{x}_j^H) \right)$$

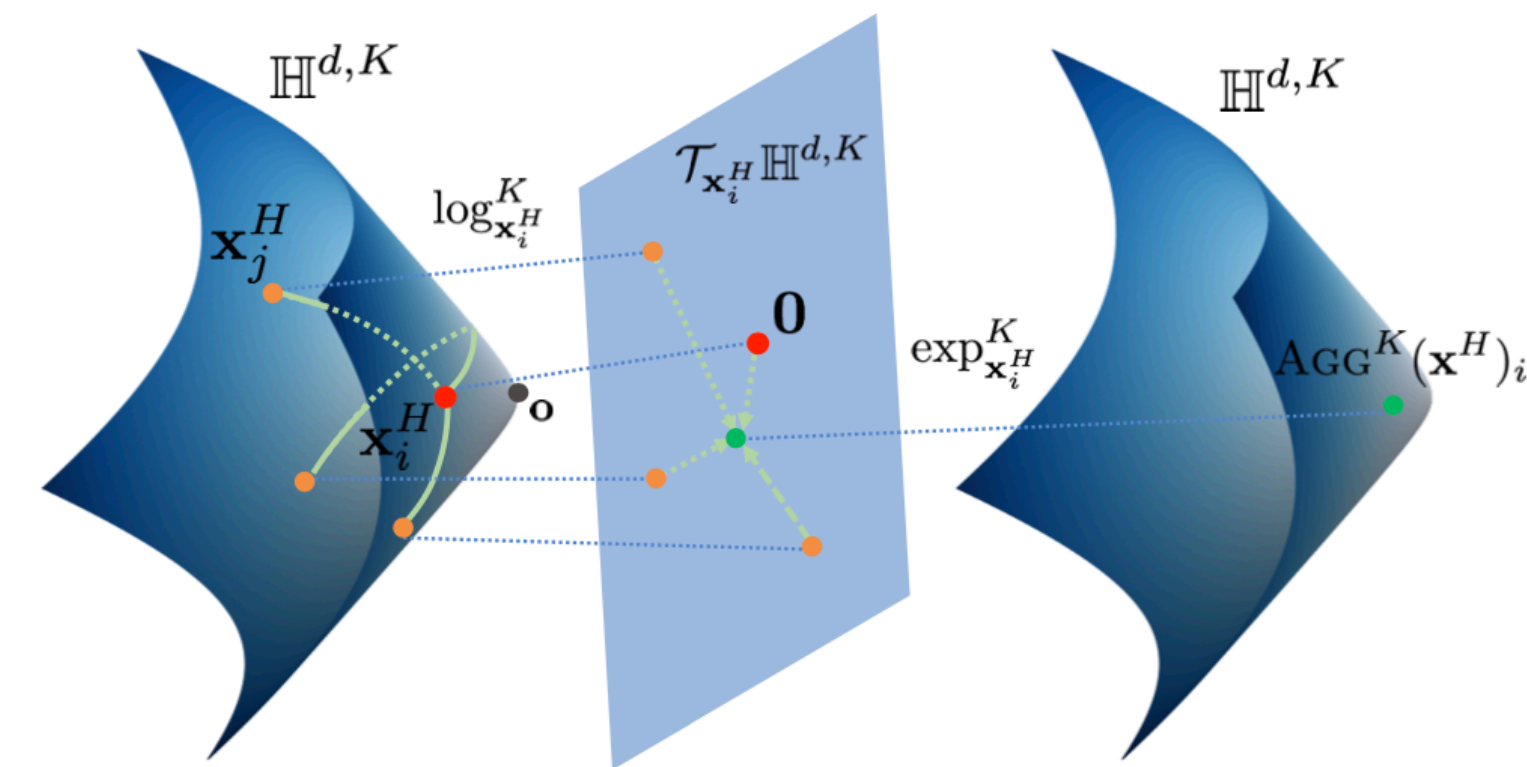
$$\sigma^{\otimes K_{\ell-1}, K_{\ell}} (\mathbf{x}^H) = \exp_{\mathbf{o}}^{K_{\ell}} \left(\sigma \left(\log_{\mathbf{o}}^{K_{\ell-1}} (\mathbf{x}^H) \right) \right)$$

不同曲率非线性激活



HGCN Architecture

给定一个网络 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, 以及相应的欧氏空间特征 $(\mathbf{x}^{0,E})_{i \in \mathcal{V}}$



1. HGCN 第一层将特征映射到双曲空间
2. HGCN 之后堆叠多个图卷积层:

$$\mathbf{h}_i^{\ell,H} = (W^\ell \otimes^{K_{\ell-1}} \mathbf{x}_i^{\ell-1,H}) \oplus^{K_{\ell-1}} \mathbf{b}^\ell \quad (\text{hyperbolic feature transform})$$

$$\mathbf{y}_i^{\ell,H} = \text{AGG}^{K_{\ell-1}}(\mathbf{h}^{\ell,H})_i \quad (\text{attention-based neighborhood aggregation})$$

$$\mathbf{x}_i^{\ell,H} = \sigma^{\otimes^{K_{\ell-1}, K_\ell}}(\mathbf{y}_i^{\ell,H}) \quad (\text{non-linear activation with different curvatures})$$

1. link prediction

使用 费米-狄拉克分布计算

$$p\left((i,j) \in \mathcal{E} \mid \mathbf{x}_i^{L,H}, \mathbf{x}_j^{L,H}\right) = \left[e^{\left(d_{\mathcal{L}}^{K_L}(\mathbf{x}_i^{L,H}, \mathbf{x}_j^{L,H})^2 - r \right) / t} + 1 \right]^{-1}$$

2. node classification

将最后一层的输出使用 log 映射映射到切平面空间, 用欧氏逻辑回归



HGCN: Trainable curvature

不同曲率会影响实验结果

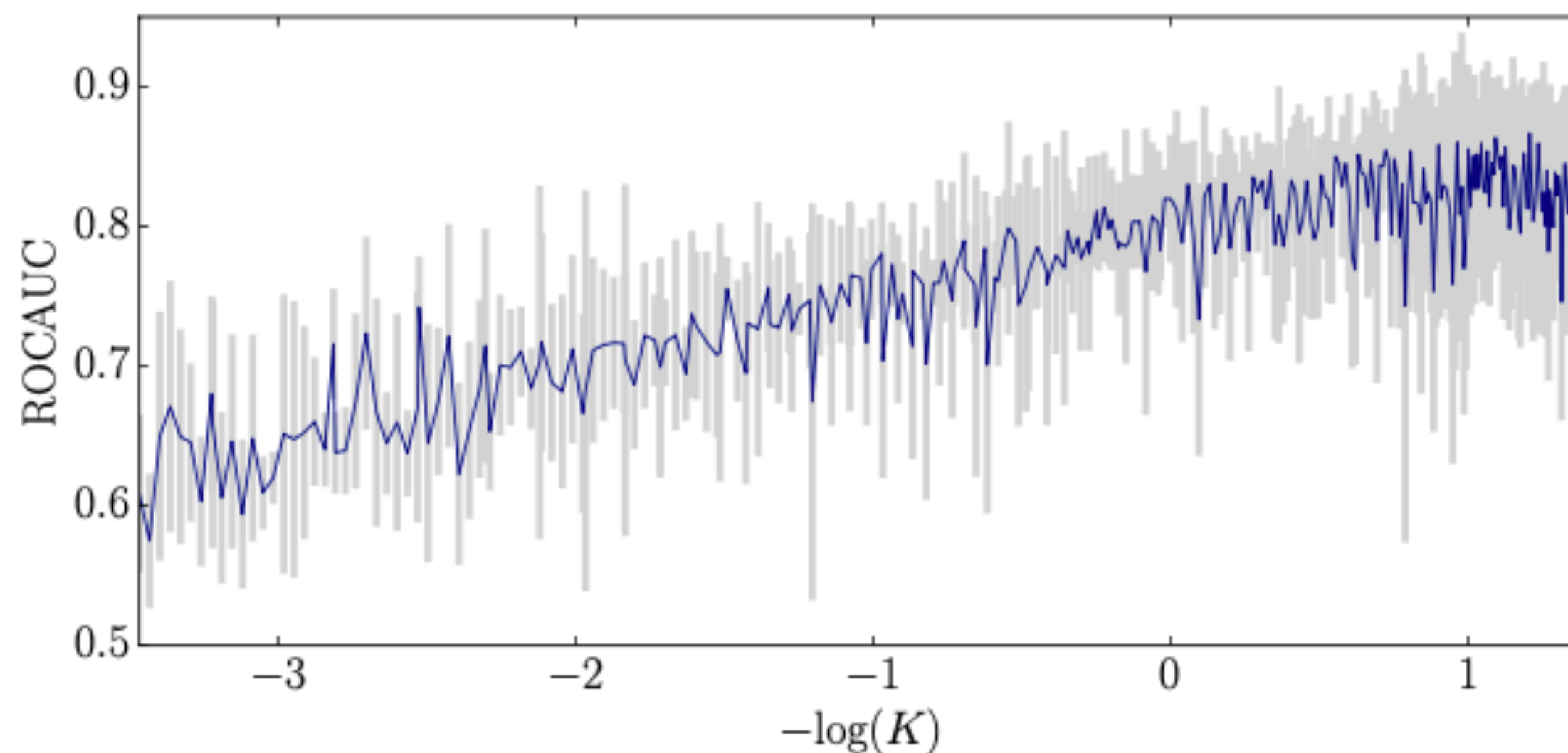
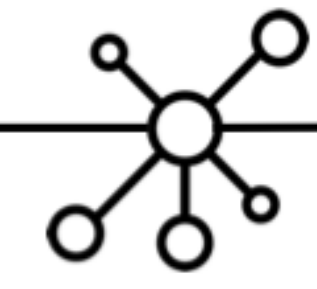


Figure 4: Decreasing curvature ($-1/K$) improves link prediction performance on DISEASE.

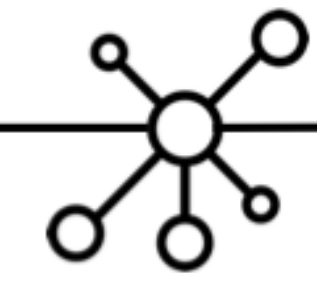


Results Analyze

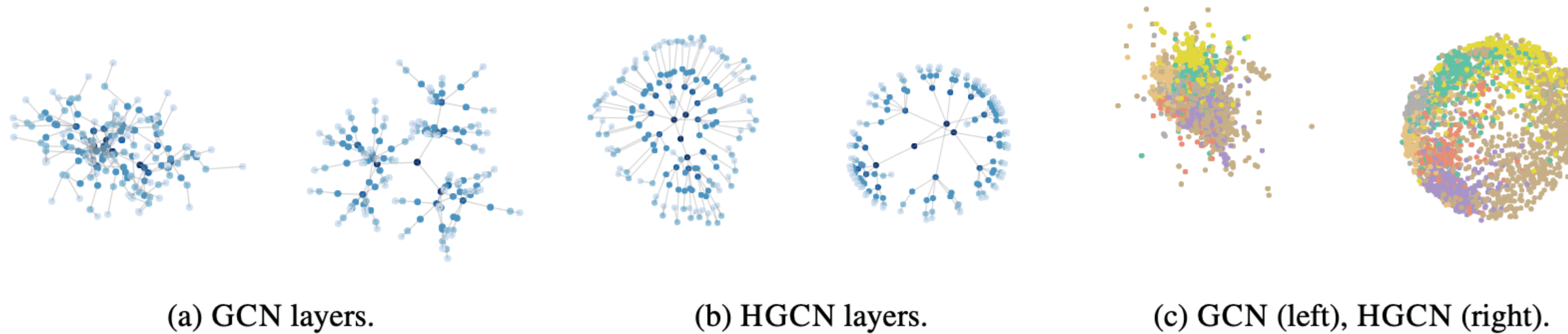
Dataset		DISEASE		DISEASE-M		HUMAN PPI		AIRPORT		PUBMED		CORA	
Hyperbolicity δ		$\delta = 0$		$\delta = 0$		$\delta = 1$		$\delta = 1$		$\delta = 3.5$		$\delta = 11$	
Method		LP	NC	LP	NC	LP	NC	LP	NC	LP	NC	LP	NC
Shallow	EUC	59.8 \pm 2.0	32.5 \pm 1.1	-	-	-	-	92.0 \pm 0.0	60.9 \pm 3.4	83.3 \pm 0.1	48.2 \pm 0.7	82.5 \pm 0.3	23.8 \pm 0.7
	HYP [29]	63.5 \pm 0.6	45.5 \pm 3.3	-	-	-	-	94.5 \pm 0.0	70.2 \pm 0.1	87.5 \pm 0.1	68.5 \pm 0.3	87.6 \pm 0.2	22.0 \pm 1.5
	EUC-MIXED	49.6 \pm 1.1	35.2 \pm 3.4	-	-	-	-	91.5 \pm 0.1	68.3 \pm 2.3	86.0 \pm 1.3	63.0 \pm 0.3	84.4 \pm 0.2	46.1 \pm 0.4
	HYP-MIXED	55.1 \pm 1.3	56.9 \pm 1.5	-	-	-	-	93.3 \pm 0.0	69.6 \pm 0.1	83.8 \pm 0.3	73.9 \pm 0.2	85.6 \pm 0.5	45.9 \pm 0.3
NN	MLP	72.6 \pm 0.6	28.8 \pm 2.5	55.3 \pm 0.5	55.9 \pm 0.3	67.8 \pm 0.2	55.3 \pm 0.4	89.8 \pm 0.5	68.6 \pm 0.6	84.1 \pm 0.9	72.4 \pm 0.2	83.1 \pm 0.5	51.5 \pm 1.0
	HNN [10]	75.1 \pm 0.3	41.0 \pm 1.8	60.9 \pm 0.4	56.2 \pm 0.3	72.9 \pm 0.3	59.3 \pm 0.4	90.8 \pm 0.2	80.5 \pm 0.5	94.9 \pm 0.1	69.8 \pm 0.4	89.0 \pm 0.1	54.6 \pm 0.4
GNN	GCN [21]	64.7 \pm 0.5	69.7 \pm 0.4	66.0 \pm 0.8	59.4 \pm 3.4	77.0 \pm 0.5	69.7 \pm 0.3	89.3 \pm 0.4	81.4 \pm 0.6	91.1 \pm 0.5	78.1 \pm 0.2	90.4 \pm 0.2	81.3 \pm 0.3
	GAT [41]	69.8 \pm 0.3	70.4 \pm 0.4	69.5 \pm 0.4	62.5 \pm 0.7	76.8 \pm 0.4	70.5 \pm 0.4	90.5 \pm 0.3	81.5 \pm 0.3	91.2 \pm 0.1	79.0 \pm 0.3	93.7 \pm 0.1	83.0 \pm 0.7
	SAGE [15]	65.9 \pm 0.3	69.1 \pm 0.6	67.4 \pm 0.5	61.3 \pm 0.4	78.1 \pm 0.6	69.1 \pm 0.3	90.4 \pm 0.5	82.1 \pm 0.5	86.2 \pm 1.0	77.4 \pm 2.2	85.5 \pm 0.6	77.9 \pm 2.4
	SGC [44]	65.1 \pm 0.2	69.5 \pm 0.2	66.2 \pm 0.2	60.5 \pm 0.3	76.1 \pm 0.2	71.3 \pm 0.1	89.8 \pm 0.3	80.6 \pm 0.1	94.1 \pm 0.0	78.9 \pm 0.0	91.5 \pm 0.1	81.0 \pm 0.1
Ours	HGCN	90.8 \pm 0.3	74.5 \pm 0.9	78.1 \pm 0.4	72.2 \pm 0.5	84.5 \pm 0.4	74.6 \pm 0.3	96.4 \pm 0.1	90.6 \pm 0.2	96.3 \pm 0.0	80.3 \pm 0.3	92.9 \pm 0.1	79.9 \pm 0.2
	(%) ERR RED	-63.1%	-13.8%	-28.2%	-25.9%	-29.2%	-11.5%	-60.9%	-47.5%	-27.5%	-6.2%	+12.7%	+18.2%

Table 1: ROC AUC for Link Prediction (LP) and F1 score for Node Classification (NC) tasks. For inductive datasets, we only evaluate inductive methods since shallow methods cannot generalize to unseen nodes/graphs. We report graph hyperbolicity values δ (lower is more hyperbolic).

计算 hyperbolicity values, 越小越像树结构



Results Analyze



HGCN 学习到层级结构

Figure 3: Visualization of embeddings for LP on DISEASE and NC on CORA (visualization on the Poincaré disk for HGCN). (a) GCN embeddings in first and last layers for DISEASE LP hardly capture hierarchy (depth indicated by color). (b) In contrast, HGCN preserves node hierarchies. (c) On CORA NC, HGCN leads to better class separation (indicated by different colors).

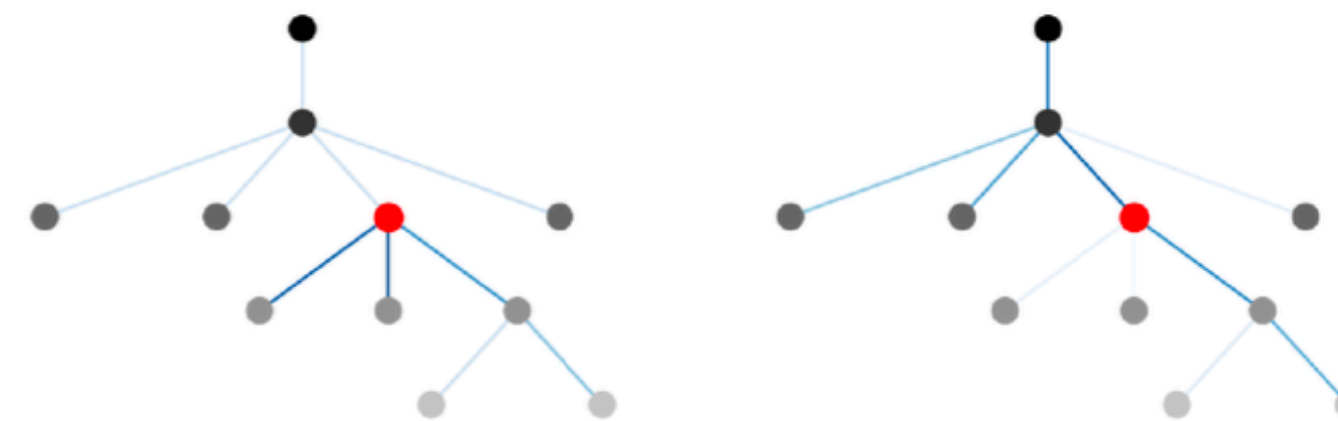


Figure 5: Attention: Euclidean GAT (left), HGCN (right). Each graph represents a 2-hop neighborhood of the DISEASE-M dataset.

更关注父节点, sick parents will propagate the disease to their children.



HGNN

Hyperbolic Graph Neural Networks

$$\mathbf{h}_u^{k+1} = \sigma \left(\sum_{v \in \mathcal{J}(u)} \tilde{\mathbf{A}}_{uv} \mathbf{W}^k \mathbf{h}_v^k \right)$$

$$\mathbf{h}_u^{k+1} = \sigma \left(\exp_{\mathbf{x}'} \left(\sum_{v \in \mathcal{J}(u)} \tilde{\mathbf{A}}_{uv} \mathbf{W}^k \log_{\mathbf{x}'} (\mathbf{h}_v^k) \right) \right)$$

NeurIPS 2019 announced accepted papers, we also became aware of the concurrently developed HGNN model [26] for learning GNNs in hyperbolic space. The main difference with our work is how our HGNN defines the architecture for neighborhood aggregation and uses a learnable curvature. Additionally, while [26] demonstrates strong performance on graph classification tasks and provides an elegant extension to dynamic graph embeddings, we focus on link prediction and node classification.