A. GCN - A Brief Survey

A.1. Convolutional Neural Networks on Graphs

One of the pioneering works on neural networks in non-Euclidean domains was done by Defferrard et al. (2016). Their idea was to extend convolutional neural networks for graphs using tools from **graph signal processing**.

Given a graph $G = (V, \mathbf{A})$, where $\mathbf{A}$ is the adjacency matrix and V is a set of nodes, we define a signal on the nodes of a graph to be a vector $\mathbf{x} \in \mathbb{R}^n$ where x_i is the value of the signal at node i. Consider the diagonalization of the symmetrized graph Laplacian $\tilde{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$. The eigenbasis $\mathbf{U}$ allows to define the graph Fourier transform $\hat{\mathbf{x}} = \mathbf{U}^T\mathbf{x} \in \mathbb{R}^n$.
In order to define a convolution for graphs, we shift from the vertex domain to the **Fourier domain**:

$$\mathbf{x} \star_G \mathbf{y} = \mathbf{U}\left(\left(\mathbf{U}^T\mathbf{x}\right) \odot \left(\mathbf{U}^T\mathbf{y}\right)\right)$$

Note that $\hat{\mathbf{x}} = \mathbf{U}^T\mathbf{x}$ and $\hat{\mathbf{y}} = \mathbf{U}^T\mathbf{y}$ are the graph Fourier representations and we use the element-wise product $\odot$ since convolutions become products in the Fourier domain. The left multiplication with $\mathbf{U}$ maps the Fourier representation back to a vertex representation.
As a consequence, a signal $\mathbf{x}$ filtered by g_θ becomes $\mathbf{y} = \mathbf{U}g_{\boldsymbol{\theta}}(\mathbf{\Lambda})\mathbf{U}^T\mathbf{x}$ where $g_{\boldsymbol{\theta}} = \text{diag}(\boldsymbol{\theta})$ with $\boldsymbol{\theta} \in \mathbb{R}^n$ constitutes a filter with all parameters free to vary. In order to avoid the resulting complexity $\mathcal{O}(n)$, Defferrard et al. (2016) replace the non-parametric filter by a polynomial filter:

$$g_{\boldsymbol{\theta}}(\mathbf{\Lambda}) = \sum_{k=0}^{K-1} \theta_k \mathbf{\Lambda}^k$$

where $\boldsymbol{\theta} \in \mathbb{R}^K$ resulting in a complexity $\mathcal{O}(K)$. Filtering a signal is unfortunately still expensive since $\mathbf{y} = \mathbf{U}g_{\boldsymbol{\theta}}(\mathbf{\Lambda})\mathbf{U}^T\mathbf{x}$ requires the multiplication with the Fourier basis $\mathbf{U}$, thus resulting in complexity $\mathcal{O}(n^2)$. As a consequence, Defferrard et al. (2016) circumvent this problem by choosing the **Chebyshev polynomials** T_k as a polynomial basis, $g_{\boldsymbol{\theta}}(\mathbf{\Lambda}) = \sum_{k=0}^{K} \theta_k T_k(\tilde{\mathbf{\Lambda}})$ where $\tilde{\mathbf{\Lambda}} = \frac{2\mathbf{\Lambda}}{\lambda_{max}} - \mathbf{I}$. As a consequence, the filter operation becomes $\mathbf{y} = \sum_{k=0}^{K} \theta_k T_k(\hat{\mathbf{L}})\mathbf{x}$ where $\hat{\mathbf{L}} = \frac{2\mathbf{L}}{\lambda_{max}} - \mathbf{I}$. This led to a K-**localized** filter since it depended on the K-th power of the Laplacian. The **recursive** nature of these polynomials allows for an efficient filtering of complexity $\mathcal{O}(K|E|)$, thus leading to an computationally appealing definition of convolution for graphs. The model can also be built in an analogous way to CNNs, by stacking multiple convolutional layers, each layer followed by a non-linearity.

A.2. Graph Convolutional Networks

Kipf & Welling (2017) extended the work of Defferrard et al. (2016) and inspired many follow-up architectures (Chen et al., 2018; Hamilton et al., 2017; Abu-El-Haija et al., 2018; Wu et al., 2019). The core idea of Kipf & Welling (2017) is to limit each filter to 1-hop neighbours by setting $K = 1$, leading to a convolution that is linear in the Laplacian $\hat{\mathbf{L}}$:

$$g_{\boldsymbol{\theta}} \star \mathbf{x} = \theta_0\mathbf{x} + \theta_1\hat{\mathbf{L}}\mathbf{x}$$

They further assume $\lambda_{max} \approx 2$, resulting in the expression

$$g_{\boldsymbol{\theta}} \star \mathbf{x} = \theta_0\mathbf{x} - \theta_1\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{x}$$

To additionally alleviate overfitting, Kipf & Welling (2017) constrain the parameters as $\theta_0 = -\theta_1 = \theta$, leading to the convolution formula

$$g_\theta \star \mathbf{x} = \theta(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{x}$$

Since $\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ has its eigenvalues in the range $[0, 2]$, they further employ a reparametrization trick to stop their model from suffering from numerical instabilities:

$$g_\theta \star \mathbf{x} = \theta\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{x}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{D}_{ii} = \sum_{j=1}^{n} \tilde{A}_{ij}$.

Rewriting the architecture for multiple features $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ and parameters $\mathbf{\Theta} \in \mathbb{R}^{d_1 \times d_2}$ instead of $\mathbf{x} \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$, gives

$$\mathbf{Z} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{\Theta} \in \mathbb{R}^{n \times d_2}$$

The final model consists of multiple stacks of convolutions, interleaved by a non-linearity σ:

$$\mathbf{H}^{(k+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(k)}\mathbf{\Theta}^{(k)}\right)$$

where $\mathbf{H}^{(0)} = \mathbf{X}$ and $\mathbf{\Theta} \in \mathbb{R}^{n \times d_k}$.

The final output $\mathbf{H}^{(K)} \in \mathbb{R}^{n \times d_K}$ represents the embedding of each node i as $\mathbf{h}_i = \mathbf{H}_{i\bullet} \in \mathbb{R}^{d_K}$ and can be used to perform node classification:

$$\hat{\mathbf{Y}} = \text{softmax}\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(K)}\mathbf{W}\right) \in \mathbb{R}^{n \times L}$$

where $\mathbf{W} \in \mathbb{R}^{d_K \times L}$, with L denoting the number of classes.

In order to illustrate how embeddings of neighbouring nodes interact, it is easier to view the architecture on the **node level**. Denote by $\mathcal{N}(i)$ the neighbours of node i. One can write the embedding of node i at layer $k+1$ as follows:

$$\boldsymbol{h}_i^{(k+1)} = \sigma\left(\mathbf{\Theta}^{(l)} \sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{\boldsymbol{h}_j^{(k)}}{\sqrt{|\mathcal{N}(j)||\mathcal{N}(i)|}}\right)$$

Notice that there is no dependence of the weight matrices $\boldsymbol{\Theta}^{(l)}$ on the node i, in fact the same **parameters are shared** across all nodes.
In order to obtain the new embedding $\mathbf{h}_i^{(k+1)}$ of node i, we average over all embeddings of the neighbouring nodes. This **Message Passing** mechanism gives rise to a very broad class of graph neural networks (Kipf & Welling, 2017; Veličković et al., 2018; Hamilton et al., 2017; Gilmer et al., 2017; Chen et al., 2018; Klicpera et al., 2019; Abu-El-Haija et al., 2018).

To be more precise, GCN falls into the more general category of models of the form

$$\boldsymbol{z}_i^{(k+1)} = \text{AGGREGATE}^{(k)}(\{\boldsymbol{h}_j^{(k)} : j \in \mathcal{N}(i)\}; \boldsymbol{W}^{(k)})$$
$$\boldsymbol{h}_i^{(k+1)} = \text{COMBINE}^{(k)}(\boldsymbol{h}_i^{(k)}, \boldsymbol{z}_i^{(k+1)}; \boldsymbol{V}^{(k)})$$

Models of the above form are deemed **Message Passing Graph Neural Networks** and many choices for AGGREGATE and COMBINE have been suggested in the literature (Kipf & Welling, 2017; Hamilton et al., 2017; Chen et al., 2018).

B. Graph Embeddings in Non-Euclidean Geometries

In this section we will motivate non-Euclidean embeddings of graphs and show why the underlying geometry of the embedding space can be very beneficial for its representation. We first introduce a measure of how well a graph is represented by some embedding $f : V \to \mathcal{X},\ i \mapsto f(i)$:

Definition 4. *Given an embedding $f : V \to \mathcal{X},\ i \mapsto f(i)$ of a graph $G = (V, \boldsymbol{A})$ in some metric space $\mathcal{X}$, we call f a **D-embedding** for $D \geq 1$ if there exists $r > 0$ such that*

$$r \cdot d_G(i,j) \leq d_{\mathcal{X}}(f(i), f(j)) \leq D \cdot r \cdot d_G(i,j)$$

*The infimum over all such D is called the **distortion** of f.*

The r in the definition of distortion allows for scaling of all distances. Note further that a perfect embedding is achieved when $D = 1$.

B.1. Trees and Hyperbolic Space

Trees are graphs that do not allow for a cycle, in other words there is no node $i \in V$ for which there exists a path starting from i and returning back to i without passing through any node twice. The number of nodes increases **exponentially** with the depth of the tree. This is a property that prohibits Euclidean space from representing a tree accurately. What intuitively happens is that "we run out of space". Consider the trees depicted in fig. 1. Here the yellow nodes represent the roots of each tree. Notice how rapidly we struggle to find appropriate places for nodes in the embedding space because their number increases just too fast.

Moreover, graph distances get **extremely distorted** towards the leaves of the tree. Take for instance the green and the pink node. In graph distance they are very far apart as one has to travel up all the way to the root node and back to the border. In Euclidean space however, they are very closely embedded in a L_2-sense, hence introducing a big error in the embedding.

This problem can be very nicely illustrated by the following theorem:

Theorem 7. *Consider the tree $K_{1,3}$ (also called 3-star) consisting of a root node with three children. Then every embedding $\{\mathbf{x}_1, \dots, \mathbf{x}_4\}$ with $\mathbf{x}_i \in \mathbb{R}^k$ achieves at least distortion $\frac{2}{\sqrt{3}}$ for any $k \in \mathbb{N}$.*

Proof. We will prove this statement by using a special case of the so called **Poincaré-type inequalities** (Deza & Laurent, 1996):

For any $b_1, \dots, b_k \in \mathbb{R}$ with $\sum_{i=1}^k b_i = 0$ and points $\mathbf{x}_1, \dots, \mathbf{x}_k \in \mathbb{R}^n$ it holds that

$$\sum_{i,j=1}^{k} b_i b_j ||\mathbf{x}_i - \mathbf{x}_j||^2 \leq 0$$

Consider now an embedding of the tree $\mathbf{x}_1, \dots, \mathbf{x}_4$ where $\mathbf{x}_1$ represents the root node. Choosing $b_1 = -3$ and $b_i = 1$ for $i \neq 1$ leads to the inequality

$$||\mathbf{x}_2 - \mathbf{x}_3||^2 + ||\mathbf{x}_2 - \mathbf{x}_4||^2 + ||\mathbf{x}_3 - \mathbf{x}_4||^2$$
$$\leq 3||\mathbf{x}_1 - \mathbf{x}_2||^2 + 3||\mathbf{x}_1 - \mathbf{x}_3||^2 + 3||\mathbf{x}_1 - \mathbf{x}_4||^2$$

The left-hand side of this inequality in terms of the graph distance is

$$d_G(2,3)^2 + d_G(2,4)^2 + d_G(3,4)^2 = 2^2 + 2^2 + 2^2 = 12$$

and the right-hand side is

$$3 \cdot d_G(1,2)^2 + 3 \cdot d_G(1,3)^2 + 3 \cdot d_G(1,4)^2 = 3 + 3 + 3 = 9$$

As a result, we always have that the distortion is lower-bounded by $\sqrt{\frac{12}{9}} = \frac{2}{\sqrt{3}}$ □

Euclidean space thus already fails to capture the geometric structure of a very simple tree. This problem can be remedied by replacing the underlying Euclidean space by hyperbolic space.

Consider again the distance function in the Poincaré model, for simplicity with $c = 1$:

$$d_{\mathbb{P}}(\mathbf{x}, \mathbf{y}) = \cosh^{-1}\left(1 + 2\frac{||\mathbf{x} - \mathbf{y}||^2}{(1 - ||\mathbf{x}||^2)(1 - ||\mathbf{y}||^2)}\right)$$

Assume that the tree is embedded in the same way as in fig.1, just restricted to lie in the disk of radius $\frac{1}{\sqrt{c}} = 1$. Notice that as soon as points move closer to the boundary ($||\mathbf{x}|| \to 1$), the fraction explodes and the resulting distance goes to infinity. As a result, the further you move points to the border, the more their distance increases, exactly as nodes on different branches are more distant to each other the further down they are in the tree. We can express this advantage in geometry in terms of distortion:

Theorem 8. *There exists an embedding $\mathbf{x}_1, \dots, \mathbf{x}_4 \in \mathbb{P}^2$ for $K_{1,3}$ achieving distortion $1+\epsilon$ for $\epsilon > 0$ arbitrary small.*

Proof. Since the Poincaré distance is invariant under Möbius translations we can again assume that $x_1 = 0$. Let us place the other nodes on a circle of radius r. Their distance to the root is now given as

$$\begin{aligned} d_{\mathbb{P}}(\mathbf{x}_i, 0) &= \cosh^{-1}\left(1 + 2\frac{||\mathbf{x}_i||^2}{1 - ||\mathbf{x}_i||^2}\right) \\ &= \cosh^{-1}\left(1 + 2\frac{r^2}{1 - r^2}\right) \end{aligned} \tag{20}$$

By invariance of the distance under centered rotations we can assume w.l.o.g. $\mathbf{x}_2 = (r, 0)$. We further embed

- $\mathbf{x}_3 = \left(r\cos(\frac{2}{3}\pi), r\sin(\frac{2}{3}\pi)\right) = \left(-\frac{r}{2}, \frac{\sqrt{3}}{2}r\right)$
- $x_4 = \left(r\cos(\frac{4}{3}\pi), r\sin(\frac{4}{3}\pi)\right) = \left(-\frac{r}{2}, \frac{\sqrt{3}}{2}r\right)$.

This procedure gives:

$$\begin{aligned} d_{\mathbb{P}}(\mathbf{x}_2, \mathbf{x}_3) &= \cosh^{-1}\left(1 + 2\frac{||\left(\frac{3r}{2}, \frac{-\sqrt{3}}{2}r\right)||^2}{(1 - r^2)^2}\right) \\ &= \cosh^{-1}\left(1 + 2\frac{3r^2}{(1 - r^2)^2}\right) \end{aligned} \tag{21}$$

If we let the points now move to the border of the disk we observe that

$$\frac{\cosh^{-1}\left(1 + 2\frac{3r^2}{(1-r^2)^2}\right)}{\cosh^{-1}\left(1 + 2\frac{r^2}{1-r^2}\right)} \xrightarrow{r \to 1} 2$$

But this means in turn that we can achieve distortion $1 + \epsilon$ for $\epsilon > 0$ arbitrary small. □

The tree-likeliness of hyperbolic space has been investigated on a deeper mathematical level. Sarkar (2011) show that a similar statement as in theorem 8 holds for all weighted or unweighted trees. The interested reader is referred to Hamann (2017); Sarkar (2011) for a more in-depth treatment of the subject.

Cycles are the subclasses of graphs that are not allowed in a tree. They consist of one path that reconnects the first and the last node: $(v_1, \dots, v_n, v_1)$. Again there is a very simple example of a cycle, hinting at the limits Euclidean space incurs when trying to preserve the geometry of these objects (Matousek, 2013).

Theorem 9. *Consider the cycle $G = (V, \boldsymbol{A})$ of length four. Then any embedding $(\mathbf{x}_1, \dots, \mathbf{x}_4)$ where $\mathbf{x}_i \in \mathbb{R}^k$ achieves at least distortion $\sqrt{2}$.*

Proof. Denote by $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ the embeddings in Euclidean space where $\mathbf{x}_1, \mathbf{x}_3$ and $\mathbf{x}_2, \mathbf{x}_4$ are the pairs without an edge. Again using the Poincaré-type inequality with $b_1 = b_3 = 1$ and $b_2 = b_4 = -1$ leads to the **short diagonal theorem** (Matousek, 2013):

$$\begin{aligned} &||\mathbf{x}_1 - \mathbf{x}_3||^2 + ||\mathbf{x}_2 - \mathbf{x}_4||^2 \\ &\leq ||\mathbf{x}_1 - \mathbf{x}_2||^2 + ||\mathbf{x}_2 - \mathbf{x}_3||^2 + ||\mathbf{x}_3 - \mathbf{x}_4||^2 + ||\mathbf{x}_4 - \mathbf{x}_1||^2 \end{aligned} \tag{22}$$

The left hand side of this inequality in terms of the graph distance is $d_G(1,3)^2 + d_G(2,4)^2 = 2^2 + 2^2 = 8$ and the right hand side is $1^2 + 1^2 + 1^2 + 1^2 = 4$.
Therefore any embedding has to shorten one diagonal by at least a factor $\sqrt{2}$. □

It turns out that in spherical space, this problem can be solved perfectly in one dimension for any cycle.

Theorem 10. *Given a cycle $G = (V, \boldsymbol{A})$ of length n, there exists an embedding $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ achieving distortion 1.*

Proof. We model the one dimension spherical space as the circle $\mathbb{S}^1$. Placing the points at angles $\frac{2\pi i}{n}$ and using the arclength on the circle as the distance measure leads to an embedding of distortion 1 as all pairwise distances are perfectly preserved. □

Notice that we could also use the exact same embedding in the two dimensional stereographic projection model with $c = 1$ and we would also obtain distortion 1. The difference to the Poincaré disk is that spherical space is finite and the border does not correspond to infinitely distant points. We therefore have no ϵ since we do not have to pass to a limit.

C. Spherical Space and its Gyrostructure

Contrarily to hyperbolic geometry, **spherical geometry** is not only in violation with the fifth postulate of Euclid but also with the first. Notice that, shortest paths are not unique as for antipodal (oppositely situated) points, we have infinitely many geodesics connecting the two. Hence the first axiom does not hold. Notice that the third postulate holds as we stated it but it is sometimes also phrased as: "A circle

of any center and radius can be constructed". Due to the finiteness of space we cannot have arbitrary large circles and hence phrased that way, the third postulate would not hold.
Finally, we replace the fifth postulate by:

- Given any straight line l and a point p not on l, there exists no shortest line g passing through p but never intersecting l.

The standard model of spherical geometry suffers from the fact that its underlying space depends directly on the curvature κ through a hard constraint $-\kappa\langle \mathbf{x}, \mathbf{x}\rangle = 1$ (similarly to the Lorentz model of hyperbolic geometry). Indeed, when $\kappa \to 0$, the domain diverges to a sphere of infinite radius which is not well defined.
For hyperbolic geometry, we could circumvent the problem by moving to the Poincaré model, which is the stereographic projection of the Lorentz model, relaxing the hard constraint to an inequality. A similar solution is also possible for the spherical model.

C.1. Stereographic Projection Model of the Sphere

In the following we construct a model in perfect duality to the construction of the Poincaré model.
Fix the south pole $\mathbf{z} = (\mathbf{0}, -\frac{1}{\sqrt{\kappa}})$ of the sphere of curvature $\kappa > 0$, *i.e.* of radius $R := \kappa^{-\frac{1}{2}}$. The **stereographic projection** is the map:

$$\Phi : \mathbb{S}^n_R \to \mathbb{R}^n, \mathbf{x}' \mapsto \mathbf{x} = \frac{1}{1 + \sqrt{\kappa}\mathbf{x}'_{n+1}}\mathbf{x}'_{1:n}$$

with the inverse given by

$$\Phi^{-1} : \mathbb{R}^n \to \mathbb{S}^n_R, \mathbf{x} \mapsto \mathbf{x}' = \left(\lambda^\kappa_{\mathbf{x}}\mathbf{x}, \frac{1}{\sqrt{\kappa}}(\lambda^\kappa_{\mathbf{x}} - 1)\right)$$

where we define $\lambda^\kappa_{\mathbf{x}} = \frac{2}{1+\kappa||\mathbf{x}||^2}$.
Again we take the image of the sphere $\mathbb{S}^n_R$ under the extended projection $\Phi((0, \dots, 0, -\frac{1}{\kappa})) = \mathbf{0}$, leading to the stereographic model of the sphere. The metric tensor transforms as:

$$g^\kappa_{ij} = (\lambda^\kappa_{\mathbf{x}})^2\delta_{ij}$$

C.2. Gyrovector Space in the Stereographic Model

C.2.1. Proof of Theorem 1

Using Cauchy-Schwarz's inequality, we have $A := 1 - 2\kappa\mathbf{x}^T\mathbf{y} + \kappa^2||\mathbf{x}||^2||\mathbf{y}||^2 \geq 1 - 2|\kappa|||\mathbf{x}||||\mathbf{y}|| + \kappa^2||\mathbf{x}||^2||\mathbf{y}||^2 = (1 - |\kappa|||\mathbf{x}||||\mathbf{y}||)^2 \geq 0$. Since equality in the Cauchy-Schwarz inequality is only reached for colinear vectors, we have that $A = 0$ is equivalent to $\kappa > 0$ and $\mathbf{x} = \mathbf{y}/(\kappa\|\mathbf{y}\|^2)$.

C.2.2. Proof of Theorem 2

Let us start by proving that for $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{v} \in T_{\mathbf{x}}\mathbb{R}^n$ the **exponential map** is given by

$$\exp^\kappa_{\mathbf{x}}(\mathbf{v}) = \frac{\lambda^\kappa_{\mathbf{x}}\left(\alpha - \sqrt{\kappa}\mathbf{x}^T\frac{\mathbf{v}}{||\mathbf{v}||}\beta\right)\mathbf{x} + \frac{1}{\sqrt{\kappa}}\beta\frac{\mathbf{v}}{||\mathbf{v}||}}{1 + (\lambda^\kappa_{\mathbf{x}} - 1)\alpha - \sqrt{\kappa}\lambda^\kappa_{\mathbf{x}}\mathbf{x}^T\frac{\mathbf{v}}{||\mathbf{v}||}\beta} \quad (23)$$

where $\alpha = \cos_\kappa(\lambda^\kappa_{\mathbf{x}}||\mathbf{v}||)$ and $\beta = \sin_\kappa(\lambda^\kappa_{\mathbf{x}}||\mathbf{v}||)$

Indeed, take a unit speed geodesic $\gamma_{\mathbf{x},\mathbf{v}}(t)$ starting from $\mathbf{x}$ with direction $\mathbf{v}$. Notice that the unit speed geodesic on the sphere starting from $\mathbf{x}' \in \mathbb{S}^{n-1}$ is given by $\Gamma_{\mathbf{x}',\mathbf{v}'}(t) = \mathbf{x}'\cos_\kappa(t) + \frac{1}{\sqrt{\kappa}}\sin_\kappa(t)\mathbf{v}'$. By the Egregium theorem, we know that $\Phi(\gamma_{\mathbf{x},\mathbf{v}}(t))$ is again a unit speed geodesic in the sphere where $\Phi^{-1} : \mathbf{x} \mapsto \mathbf{x}' = \left(\lambda^\kappa_{\mathbf{x}}\mathbf{x}, \frac{1}{\sqrt{\kappa}}(\lambda^\kappa_{\mathbf{x}} - 1)\right)$. Hence $\Phi(\gamma_{\mathbf{x},\mathbf{v}}(t))$ is of the form of Γ for some $\mathbf{x}'$ and $\mathbf{v}'$. We can determine those by

$$\mathbf{x}' = \Phi^{-1}(\gamma(0)) = \Phi^{-1}(\mathbf{x}) = \left(\lambda^\kappa_{\mathbf{x}}\mathbf{x}, \frac{1}{\sqrt{\kappa}}(\lambda^\kappa_{\mathbf{x}} - 1)\right)$$

$$\mathbf{v}' = \dot{\Gamma}(0) = \frac{\partial\Phi^{-1}(\mathbf{y})}{\partial\mathbf{y}}\gamma(0)\dot{\gamma}(0)$$

Notice that $\nabla_{\mathbf{x}}\lambda^\kappa_{\mathbf{x}} = -\kappa(\lambda^\kappa_{\mathbf{x}})^2\mathbf{x}$ and we thus get

$$\mathbf{v}' = \begin{pmatrix} -2\kappa(\lambda^\kappa_{\mathbf{x}})^2\mathbf{x}^T\mathbf{v}\mathbf{x} + \lambda^\kappa_{\mathbf{x}}\mathbf{v} \\ -\sqrt{\kappa}(\lambda^\kappa_{\mathbf{x}})^2\mathbf{x}^T\mathbf{v} \end{pmatrix}$$

We can obtain $\gamma_{\mathbf{x},\mathbf{v}}$ again by inverting back by calculating $\gamma_{\mathbf{x},\mathbf{v}}(t) = \Phi(\Gamma_{\mathbf{x}',\mathbf{v}'}(t))$, resulting in

$$\gamma_{\mathbf{x},\mathbf{v}}(t) = \frac{(\lambda^\kappa_{\mathbf{x}}\cos_\kappa(t) - \sqrt{\kappa}(\lambda^\kappa_{\mathbf{x}})^2\mathbf{x}^T\mathbf{v}\sin_\kappa(t))\mathbf{x}}{1 + (\lambda^\kappa_{\mathbf{x}} - 1)\cos_\kappa(t) - \sqrt{\kappa}(\lambda^\kappa_{\mathbf{x}})^2\mathbf{x}^T\mathbf{v}\sin_\kappa(t)} + \frac{\frac{1}{\sqrt{\kappa}}\lambda^\kappa_{\mathbf{x}}\sin_\kappa(t)\mathbf{v}}{1 + (\lambda^\kappa_{\mathbf{x}} - 1)\cos_\kappa(t) - \sqrt{\kappa}(\lambda^\kappa_{\mathbf{x}})^2\mathbf{x}^T\mathbf{v}\sin_\kappa(t)}$$

Denoting $g^\kappa_{\mathbf{x}}(\mathbf{v},\mathbf{v}) = ||\mathbf{v}||^2\lambda^\kappa_{\mathbf{x}}$ we have that $\exp^\kappa_{\mathbf{x}}(\mathbf{v}) = \gamma_{\mathbf{x},\frac{1}{\sqrt{g^\kappa_{\mathbf{x}}(\mathbf{v},\mathbf{v})}}\mathbf{v}}\left(\sqrt{g^\kappa_{\mathbf{x}}(\mathbf{v},\mathbf{v})}\right)$ which concludes the proof of the above formula of the exponential map. One then notices that it can be re-written in terms of the κ-addition. The formula for the logarithmic map is easily checked by verifying that it is indeed the inverse of the exponential map. Finally, the distance formula is obtained via the well-known identity $d_\kappa(\mathbf{x},\mathbf{y}) = \|\log^\kappa_{\mathbf{x}}(\mathbf{y})\|_{\mathbf{x}}$ where $\|\mathbf{v}\|_{\mathbf{x}} = \sqrt{g^\kappa_{\mathbf{x}}(\mathbf{v},\mathbf{v})}$.

Note that as expected, $\exp^\kappa_{\mathbf{x}}(\mathbf{v}) \to_{\kappa\to 0} \mathbf{x} + \mathbf{v}$, converging to the Euclidean exponential map.

□

C.2.3. Proof of Theorem 3

We first compute a Taylor development of the κ-addition w.r.t κ around zero:

$$\begin{aligned}\mathbf{x} \oplus_\kappa \mathbf{y} &= \frac{(1-2\kappa\mathbf{x}^T\mathbf{y} - \kappa||\mathbf{y}||^2)\mathbf{x} + (1+\kappa||\mathbf{x}||^2)\mathbf{y}}{1-2\kappa\mathbf{x}^T\mathbf{y}+\kappa^2||\mathbf{x}||^2||\mathbf{y}||^2} \\ &= [(1-2\kappa\mathbf{x}^T\mathbf{y} - \kappa||\mathbf{y}||^2)\mathbf{x} \\ &\quad + (1+\kappa||\mathbf{x}||^2)\mathbf{y}][1+2\kappa\mathbf{x}^T\mathbf{y}+\mathcal{O}(\kappa^2)] \\ &= (1-2\kappa\mathbf{x}^T\mathbf{y} - \kappa||\mathbf{y}||^2)\mathbf{x} + (1+\kappa||\mathbf{x}||^2)\mathbf{y} \\ &\quad + 2\kappa\mathbf{x}^T\mathbf{y}[\mathbf{x}+\mathbf{y}] + \mathcal{O}(\kappa^2) \\ &= (1-\kappa||\mathbf{y}||^2)\mathbf{x} + (1+\kappa||\mathbf{x}||^2)\mathbf{y} + 2\kappa(\mathbf{x}^T\mathbf{y})\mathbf{y} \\ &\quad + \mathcal{O}(\kappa^2) \\ &= \mathbf{x}+\mathbf{y}+\kappa[||\mathbf{x}||^2\mathbf{y} - ||\mathbf{y}||^2\mathbf{x} + 2(\mathbf{x}^T\mathbf{y})\mathbf{y}] + \mathcal{O}(\kappa^2).\end{aligned} \tag{24}$$

We then notice that using the Taylor of $\|\cdot\|_2$, given by $\|\mathbf{x}+\mathbf{v}\|_2 = \|\mathbf{x}\|_2 + \langle \mathbf{x}, \mathbf{v}\rangle + \mathcal{O}(\|\mathbf{v}\|_2^2)$ for $\mathbf{v} \to \mathbf{0}$, we get

$$\begin{aligned}\|\mathbf{x} \oplus_\kappa \mathbf{y}\| &= \|\mathbf{x}+\mathbf{y}\| + \kappa\langle ||\mathbf{x}||^2\mathbf{y} - ||\mathbf{y}||^2\mathbf{x} \\ &\quad + 2(\mathbf{x}^T\mathbf{y})\mathbf{y}, \mathbf{x}+\mathbf{y}\rangle + \mathcal{O}(\kappa^2) \\ &= \|\mathbf{x}+\mathbf{y}\| + \kappa(\mathbf{x}^T\mathbf{y})\|\mathbf{x}+\mathbf{y}\|^2 + \mathcal{O}(\kappa^2).\end{aligned} \tag{25}$$

Finally Taylor developments of $\tan_\kappa(|\kappa|^{\frac{1}{2}}u)$ and $|\kappa|^{-\frac{1}{2}}\tan_\kappa^{-1}(u)$ w.r.t κ around 0 for fixed u yield For $\kappa \to 0^+$

$$\begin{aligned}\tan_\kappa(|\kappa|^{\frac{1}{2}}u) &= \kappa^{-\frac{1}{2}}\tan(\kappa^{\frac{1}{2}}u) \\ &= \kappa^{-\frac{1}{2}}(\kappa^{\frac{1}{2}}u + \kappa^{\frac{3}{2}}u^3/3\mathcal{O}(\kappa^{\frac{5}{2}}) \\ &= u + \kappa u^3/3 + \mathcal{O}(\kappa^2).\end{aligned} \tag{26}$$

For $\kappa \to 0^-$,

$$\begin{aligned}\tan_\kappa(|\kappa|^{\frac{1}{2}}u) &= (-\kappa)^{-\frac{1}{2}}\tanh((-\kappa)^{\frac{1}{2}}u) \\ &= (-\kappa)^{-\frac{1}{2}}((-\kappa)^{\frac{1}{2}}u - (-\kappa)^{\frac{3}{2}}u^3/3 + \mathcal{O}(\kappa^{\frac{5}{2}}) \\ &= u + \kappa u^3/3 + \mathcal{O}(\kappa^2).\end{aligned} \tag{27}$$

The left and right derivatives match, hence even though $\kappa \mapsto |\kappa|^{\frac{1}{2}}$ is not differentiable at $\kappa = 0$, the function $\kappa \mapsto \tan_\kappa(|\kappa|^{\frac{1}{2}}u)$ is. A similar analysis yields the same conclusion for $\kappa \mapsto |\kappa|^{-\frac{1}{2}}\tan_\kappa^{-1}(u)$ yielding

$$\text{For } \kappa \to 0, \quad |\kappa|^{-\frac{1}{2}}\tan_\kappa^{-1}(u) = u - \kappa u^3/3 + \mathcal{O}(\kappa^2). \tag{28}$$

Since a composition of differentiable functions is differentiable, we consequently obtain that $\otimes_\kappa$, $\exp^\kappa$, $\log^\kappa$ and d_κ are differentiable functions of κ, under the assumptions on $\mathbf{x}, \mathbf{y}, \mathbf{v}$ stated in Theorem 3. Finally, the Taylor development of d_κ follows by composition of Taylor developments:

$$\begin{aligned}d_\kappa(\mathbf{x},\mathbf{y}) &= 2\|\kappa\|^{-\frac{1}{2}}\tan_\kappa^{-1}(\|(-\mathbf{x}) \oplus_\kappa \mathbf{y}\|) \\ &= 2(\|\mathbf{x}-\mathbf{y}\| + \kappa((-\mathbf{x})^T\mathbf{y})\|\mathbf{x}-\mathbf{y}\|^2)\Big(1- \\ &\quad (\kappa/3)(\|\mathbf{x}-\mathbf{y}\| + \mathcal{O}(\kappa))^2\Big) + \mathcal{O}(\kappa^2) \\ &= 2(\|\mathbf{x}-\mathbf{y}\| + \kappa((-\mathbf{x})^T\mathbf{y})\|\mathbf{x}-\mathbf{y}\|^2)\Big(1 \\ &\quad - (\kappa/3)\|\mathbf{x}-\mathbf{y}\|^2\Big) + \mathcal{O}(\kappa^2) \\ &= 2\|\mathbf{x}-\mathbf{y}\| - 2\kappa\left((\mathbf{x}^T\mathbf{y})\|\mathbf{x}-\mathbf{y}\|^2 + \|\mathbf{x}-\mathbf{y}\|^3/3\right) \\ &\quad + \mathcal{O}(\kappa^2).\end{aligned}$$

□

C.2.4. Proof of Theorem 4

If $\mathbf{A} = \mathbf{I}_n$ then for all i we have $\sum_j A_{ij} = 1$, hence

$$(\mathbf{I}_n \boxtimes \mathbf{X})_{i\bullet} = \frac{1}{2} \otimes_\kappa \left(\sum_j \frac{\delta_{ij}\lambda^\kappa_{\mathbf{x}_j}}{\sum_k \delta_{ik}(\lambda^\kappa_{\mathbf{x}_k}-1)}\mathbf{x}_j\right) \tag{29}$$

$$= \frac{1}{2} \otimes_\kappa \left(\frac{\lambda^\kappa_{\mathbf{x}_i}}{(\lambda^\kappa_{\mathbf{x}_i}-1)}\mathbf{x}_i\right) \tag{30}$$

$$= \frac{1}{2} \otimes_\kappa (2 \otimes_\kappa \mathbf{x}_i) \tag{31}$$

$$= \mathbf{x}_i \tag{32}$$

$$= (\mathbf{X})_{i\bullet}. \tag{33}$$

For associativity, we first note that the gyromidpoint is unchanged by a scalar rescaling of $\mathbf{A}$. The property then follows by scalar associativity of the κ-scaling.

□

C.2.5. Proof of Theorem 5

It is proved in Ungar (2005) that the gyromidpoint commutes with isometries. The exact same proof holds for positive curvature, with the same algebraic manipulations. Moreover, when the matrix $\mathbf{A}$ is right-stochastic, for each row, the sum over columns gives 1, hence our operation $\boxtimes_\kappa$ reduces to a gyromidpoint. As a consequence, our $\boxtimes_\kappa$ commutes with isometries in this case. Since isometries preserve distance, we have proved the theorem.

□

C.2.6. Proof of Theorem 6

We begin our proof by stating the *left-cancellation law*:

$$\mathbf{x} \oplus_\kappa (-\mathbf{x} \oplus_\kappa \mathbf{y}) = \mathbf{y} \tag{34}$$

and the following simple identity stating that orthogonal maps commute with κ-addition

$$\mathbf{R}\mathbf{x} \oplus_\kappa \mathbf{R}\mathbf{y} = \mathbf{R}(\mathbf{x} \oplus_\kappa \mathbf{y}), \quad \forall \mathbf{R} \in O(d) \tag{35}$$

Next, we generalize the gyro operator from Möbius gyrovector spaces as defined in Ungar (2008):

$$\text{gyr}[\mathbf{u}, \mathbf{v}]\mathbf{w} := -(\mathbf{u} \oplus_\kappa \mathbf{v}) \oplus_\kappa (\mathbf{u} \oplus_\kappa (\mathbf{v} \oplus_\kappa \mathbf{w})) \tag{36}$$

Note that this definition applies only for $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathfrak{st}_\kappa^d$ for which the κ-addition is defined (see theorem 1). Following Ungar (2008), we have an alternative formulation (verifiable via computer algebra):

$$\text{gyr}[\mathbf{u}, \mathbf{v}]\mathbf{w} = \mathbf{w} + 2\frac{A\mathbf{u} + B\mathbf{v}}{D}. \tag{37}$$

where the quantities A, B, D have the following closed-form expressions:

$$A = -\kappa^2\langle \mathbf{u}, \mathbf{w}\rangle\|\mathbf{v}\|^2 - \kappa\langle \mathbf{v}, \mathbf{w}\rangle + 2\kappa^2\langle \mathbf{u}, \mathbf{v}\rangle \cdot \langle \mathbf{v}, \mathbf{w}\rangle, \tag{38}$$

$$B = -\kappa^2\langle \mathbf{v}, \mathbf{w}\rangle\|\mathbf{u}\|^2 + \kappa\langle \mathbf{u}, \mathbf{w}\rangle, \tag{39}$$

$$D = 1 - 2\kappa\langle \mathbf{u}, \mathbf{v}\rangle + \kappa^2\|\mathbf{u}\|^2\|\mathbf{v}\|^2. \tag{40}$$

We then have the following relations:

Lemma 11. *For all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathfrak{st}_\kappa^d$ for which the κ-addition is defined we have the following relations: i) gyration is a linear map, ii) $\mathbf{u} \oplus_\kappa \mathbf{v} = gyr[\mathbf{u}, \mathbf{v}](\mathbf{v} \oplus_\kappa \mathbf{u})$, iii) $-(\mathbf{z} \oplus_\kappa \mathbf{u}) \oplus_\kappa (\mathbf{z} \oplus_\kappa \mathbf{v}) = gyr[\mathbf{z}, \mathbf{u}](-\mathbf{u} \oplus_\kappa \mathbf{v})$, iv) $\|gyr[\mathbf{u}, \mathbf{v}]\mathbf{w}\| = \|\mathbf{w}\|$.*

Proof. The proof is similar with the one for negative curvature given in Ungar (2008). The fact that gyration is a linear map can be easily verified from its definition. For the second part, we have

$$\begin{aligned} -\text{gyr}[\mathbf{u}, \mathbf{v}](\mathbf{v} \oplus_\kappa \mathbf{u}) &= \text{gyr}[\mathbf{u}, \mathbf{v}](-(\mathbf{v} \oplus_\kappa \mathbf{u})) \\ &= -(\mathbf{u} \oplus_\kappa \mathbf{v}) \\ &\quad \oplus_\kappa (\mathbf{u} \oplus_\kappa (\mathbf{v} \oplus_\kappa (-(\mathbf{v} \oplus_\kappa \mathbf{u})))) \\ &= -(\mathbf{u} \oplus_\kappa \mathbf{v}) \end{aligned} \tag{41}$$

where the first equality is a trivial consequence of the fact that gyration is a linear map, while the last equality is the consequence of left-cancellation law.

The third part follows easily from the definition of the gyration and the left-cancellation law. The fourth part can be checked using the alternate form in equation (37). □

We now follow Ungar (2014) and describe all isometries of $\mathfrak{st}_\kappa^d$ spaces:

Theorem 12. *Any isometry ϕ of $\mathfrak{st}_\kappa^d$ can be uniquely written as:*

$$\phi(\mathbf{x}) = \mathbf{z} \oplus_\kappa \mathbf{R}\mathbf{x}, \quad \textit{where } \mathbf{z} \in \mathfrak{st}_\kappa^d, \mathbf{R} \in O(d) \tag{42}$$

The proof is exactly the same as in theorems 3.19 and 3.20 of Ungar (2014), so we will skip it.

We can now prove the main theorem. Let $\phi(\mathbf{x}) = \mathbf{z} \oplus_\kappa \mathbf{R}\mathbf{x}$ be any isometry of $\mathfrak{st}_\kappa^d$, where $\mathbf{R} \in O(d)$ is an orthogonal matrix. Let us denote by $\mathbf{v} := \sum_{i=1}^n \alpha_i \log_{\mathbf{x}}^\kappa(\mathbf{x}_i)$. Then, using lemma 11 and the formula of the log map from theorem 2, one obtains the following identity:

$$\sum_{i=1}^n \alpha_i \log_{\phi(\mathbf{x})}^\kappa(\phi(\mathbf{x}_i)) = \frac{\lambda_{\mathbf{x}}^\kappa}{\lambda_{\phi(\mathbf{x})}^\kappa}\text{gyr}[\mathbf{z}, \mathbf{R}\mathbf{x}]\mathbf{R}\mathbf{v} \tag{43}$$

and, thus, using the formula of the exp map from theorem 2 we obtain:

$$\begin{aligned} \mathfrak{tg}_{\phi(\mathbf{x})}(\{\phi(\mathbf{x}_i)\}; \{\alpha_i\}) = \phi(\mathbf{x}) \oplus_\kappa \text{gyr}[\mathbf{z}, \mathbf{R}\mathbf{x}]\mathbf{R}\Big(-\mathbf{x} \\ \oplus_\kappa \exp_{\mathbf{x}}^\kappa(\mathbf{v})\Big) \end{aligned} \tag{44}$$

Using eq. (36), we get that $\forall \mathbf{w} \in \mathfrak{st}_\kappa^d$:

$$\text{gyr}[\mathbf{z}, \mathbf{R}\mathbf{x}]\mathbf{R}\mathbf{w} = -\phi(\mathbf{x}) \oplus_\kappa (\mathbf{z} \oplus_\kappa (\mathbf{R}\mathbf{x} \oplus_\kappa \mathbf{R}\mathbf{w})) \tag{45}$$

giving the desired

$$\begin{aligned} \mathfrak{tg}_{\phi(\mathbf{x})}(\{\phi(\mathbf{x}_i)\}; \{\alpha_i\}) &= \mathbf{z} \oplus_\kappa \mathbf{R}\exp_{\mathbf{x}}^\kappa(\mathbf{v}) \\ &= \phi\left(\mathfrak{tg}_{\mathbf{x}}(\{\mathbf{x}_i\}; \{\alpha_i\})\right) \end{aligned} \tag{46}$$

□

D. Logits

The final element missing in the κ-GCN is the logit layer, a necessity for any classification task. We here use the formulation of Ganea et al. (2018a). Denote by $\{1, \dots, K\}$ the possible labels and let $\mathbf{a}_k \in \mathbb{R}^d$, $b_k \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^d$. The output of a feed forward neural network for classification tasks is usually of the form

$$p(y = k|\mathbf{x}) = \text{S}(\langle \mathbf{a}_k, \mathbf{x}\rangle - b_k)$$

where $S(\cdot)$ denotes the softmax function. In order to generalize this expression to hyperbolic space, Ganea et al. (2018a) realized that the term in the softmax can be rewritten as

$$\langle \mathbf{a}_k, \mathbf{x}\rangle - b_k = \text{sign}(\langle \mathbf{a}_k, \mathbf{x}\rangle - b_k)||\mathbf{a}_k||d(\mathbf{x}, H_{\mathbf{a}_k, b_k})$$

Table 3. Average curvature obtained for node classification. $\mathbb{H}$ and $\mathbb{S}$ denote hyperbolic and spherical models respectively. Curvature for Pubmed was fixed for the product model.

MODEL	CITESEER	CORA	PUBMED	AIRPORT
$\mathbb{H}^{16}$ (κ-GCN)	-1.306 ± 0.08	-1.51 ± 0.11	-1.42 ± 0.12	-0.93 ± 0.08
$\mathbb{S}^{16}$ (κ-GCN)	0.81 ± 0.05	1.24 ± 0.06	0.71 ± 0.15	1.49 ± 0.08
PROD κ-GCN	$[1.21, -0.64] \pm [0.09, 0.07]$	$[-0.83, -1.61] \pm [0.04, 0.06]$	$[-1, -1]$	$[1.23, -0.89] \pm [0.07, 0.11]$

where $H_{\mathbf{a},b} = \{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{x}, \mathbf{a} \rangle - b = 0\} = \{\mathbf{x} \in \mathbb{R}^d : \langle -\mathbf{p} + \mathbf{x}, \mathbf{a} \rangle = 0\} = \tilde{H}_{\mathbf{a},\mathbf{p}}$ with $\mathbf{p} \in \mathbb{R}^d$.
As a first step, they define the hyperbolic hyperplane as

$$\tilde{H}^{\kappa}_{\mathbf{a},\mathbf{p}} = \{\mathbf{x} \in \mathfrak{st}^d_\kappa \langle -\mathbf{p} \oplus_\kappa \mathbf{x}, \mathbf{a} \rangle = 0\}$$

where now $\mathbf{a} \in \mathcal{T}_{\mathbf{p}} \mathfrak{st}^d_\kappa$ and $\mathbf{p} \in \mathfrak{st}^d_\kappa$. They then proceed proving the following formula:

$$d_\kappa(\mathbf{x}, \tilde{H}_{\mathbf{a},\mathbf{p}}) = \frac{1}{\sqrt{-\kappa}} \sinh^{-1}\left(\frac{2\sqrt{-\kappa}|\langle \mathbf{z}, \mathbf{a} \rangle|}{(1 + \kappa ||\mathbf{z}||^2)||\mathbf{a}||} \right) \quad (47)$$

where $\mathbf{z} = -\mathbf{p} \oplus_\kappa \mathbf{x}$ Using this equation, they were able to obtain the following expression for the logit layer:

$$p(y = k|\mathbf{x}) = \mathrm{S}\left(\frac{||\mathbf{a}_k||_{\mathbf{p}_k}}{\sqrt{-\kappa}} \sinh^{-1}\left(\frac{2\sqrt{-\kappa}\langle \mathbf{z}_k, \mathbf{a}_k \rangle}{(1 + \kappa ||\mathbf{z}_k||^2)||\mathbf{a}_k||} \right) \right) \quad (48)$$

where $\mathbf{a}_k \in \mathcal{T}_0 \mathfrak{st}^d_\kappa \cong \mathbb{R}^d$, $\mathbf{x} \in \mathfrak{st}^d_\kappa$ and $\mathbf{p}_k \in \mathfrak{st}^d_\kappa$. Combining all these operations leads to the definition of a hyperbolic feed forward neural network. Notice that the weight matrices $\mathbf{W}$ and the normal vectors $\mathbf{a}_k$ live in Euclidean space and hence can be optimized by standard methods such as ADAM (Kingma & Ba, 2015).

For positive curvature $\kappa > 0$ we use in our experiments the following formula for the softmax layer:

$$p(y = k|\mathbf{x}) = \mathrm{S}\left(\frac{||\mathbf{a}_k||_{\mathbf{p}_k}}{\sqrt{\kappa}} \sin^{-1}\left(\frac{2\sqrt{\kappa}\langle \mathbf{z}_k, \mathbf{a}_k \rangle}{(1 + \kappa ||\mathbf{z}_k||^2)||\mathbf{a}_k||} \right) \right), \quad (49)$$

which is inspired from the formula $i \sin(x) = \sinh(ix)$ where $i := \sqrt{-1}$. However, we leave for future work the rigorous proof that the distance to geodesic hyperplanes in the positive curvature setting is given by this formula.

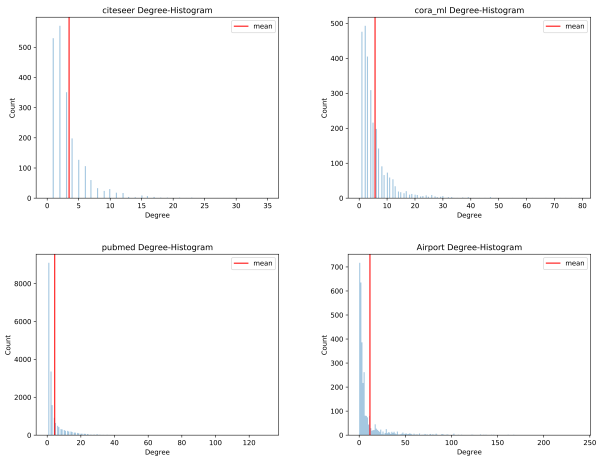

Figure 8. Histogram of node degrees

E. Additional Experiments

To study the differentiability of the derived interpolation empirically, we embedded a small tree into $\mathfrak{st}^d_\kappa$ for $\kappa \in [-5, 5]$. To that end, we used 200 equidistant values κ in $[-5, 5]$ and trained a κ-GCN (with non-trainable curvature) to produce an embedding that minimizes distortion. Moreover we also trained a Euclidean GCN with the same architecture. The results are summarized in Figure 9.
One can observe a smooth transition between the different geometries. As expected, the distortion improves for increased hyperbolicity and worsens when the embedding space becomes more spherical. The small kink for the spherical model is due to numerical issues.

F. More Experimental Details

We here present training details for the node classification experiments.

We split the data into training, early stopping, validation and test set. Namely we first split the dataset into a known subset of size n_{known} and an unknown subset consisting of the rest of the nodes. For all the graphs we use $n_{known} = 1500$ except for the airport dataset, where we follow the setup of Chami et al. (2019) and use $n_{known} = 2700$. For the citation graphs, the known subset is further split into a

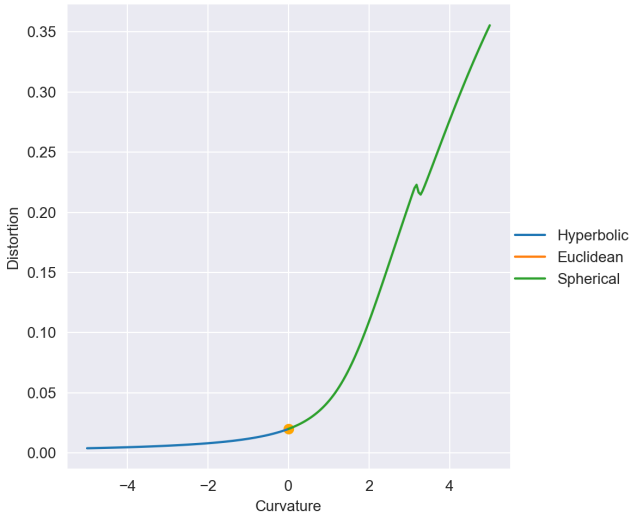

Figure 9. Distortion of κ-GCNs of varying curvature κ trained to embed a tree

Algorithm 1 Curvature Estimation

Input: Graph G, n_{iter}
for $m \in G$ **do**
  for $i = 1$ **to** n_{iter} **do**
    $b, c \sim \mathcal{U}(\mathcal{N}(m))$ and $a \sim \mathcal{U}(G)$ such that $m \neq a$
    $\psi(m; b, c; a) = \frac{d_G(a,m)}{2} + \frac{d_G^2(b,c)}{8d_G(a,m)} - \frac{2d_G^2(a,b)+2d_G^2(a,c)}{4d_G(a,m)}$
  end for
  $\psi(m) = \text{AVERAGE}(\psi(m; a, b; c))$
end for
return: $\kappa = \text{AVERAGE}(\psi(m))$

training set consisting of 20 data points per label, an early stopping set of size 500 and a validation set of the remaining nodes. For airport, we separate the known subset into 2100 training nodes, 300 validation nodes and 300 early stopping nodes. Notice that the whole structure of the graph and all the node features are used in an unsupervised fashion since the embedding of a training node might for instance depend on the embedding of a node from the validation set. But when calculating the loss, we only provide supervision with the training data.
The unknown subset serves as the test data and is only used for the final evaluation of the model. Hyperparameter-tuning is performed on the validation set. We further use early stopping in all the experiments. We stop training as soon as the early stopping cross entropy loss has not decreased in the last $n_{patience} = 200$ epochs or as soon as we have reached $n_{max} = 2000$ epochs. The model chosen is the one with the highest accuracy score on the early stopping set. For the final evaluation we test the model on five different data splits and two runs each and report mean accuracy and bootstrapped confidence intervals. We use the described setup for both the Euclidean and non-Euclidean models to ensure a fair comparison.

Learned curvatures . We report the learnt curvatures for node classification in tab. 3

G. Graph Curvature Estimation Algorithm

We used the procedure outlined in Algorithm 1 to estimate the curvature of a dataset developed by Gu et al. (2019).