**Introduction**

**Goal:** Supervised learning on entire graphs.

**Motivation:**
- Graph neural networks can exploit symmetries in graph-structured data, showing promise for the classification of graphs based on their structural properties.
- Typical properties of complex networks such as heterogeneous degree distributions and strong clustering can often be explained by assuming an underlying hierarchy, which is well captured in hyperbolic space (Krioukov et al., 2010).

**Main Idea:** We extend graph neural networks to operate on Riemannian manifolds with differentiable exponential and logarithmic maps.

**Hyperbolic Graph Neural Networks**

Graph neural networks can be interpreted as performing message passing between nodes (Kipf and Welling, 2017, Gilmer et al., 2017).

**Vanilla GCN**

\[
\tilde{h}_{t+1} = \sigma \left( \sum_{j \in \mathcal{N}(i)} A_{ij} W_h \tilde{h}_j \right)
\]

where \( A = D^{-1/2} (A + I) D^{-1/2} \) corresponds to the normalized adjacency matrix.

We generalize this notion such that the network operates on Riemannian manifolds and becomes manifold preserving, i.e., that \( \sigma: \mathcal{M} \rightarrow \mathcal{M} \).

**HGNN**

\[
\tilde{h}_{t+1} = \sigma \left( \exp_{\mathcal{M}} \left( \sum_{j \in \mathcal{N}(i)} A_{ij} W_h \log_{\mathcal{M}}(\tilde{h}_j) \right) \right)
\]

When applying the non-linearity directly on a manifold, \( M \), we need to ensure that its application is manifold preserving, i.e., that \( \sigma: \mathcal{M} \rightarrow \mathcal{M} \).

**Riemannian Manifolds**

A Riemannian manifold \((\mathcal{M}, g)\) is a real and smooth manifold equipped with an inner product \( g: T_p\mathcal{M} \times T_p\mathcal{M} \rightarrow \mathbb{R} \) at each point \( p \in \mathcal{M} \).

We study the Euclidean and two hyperbolic (Poincaré and Lorentz) manifolds, applying their corresponding log/exp maps.

**Euclidean Space:** The Euclidean manifold is a manifold with zero curvature.

\[
\exp(v) = x + v \quad \text{and} \quad \log(v) = y - x.
\]

**Poincaré Ball Model:** The Poincaré ball model with constant negative curvature corresponds to the Riemannian manifold \((B, g^B)\), where \( B = \{ x \in \mathbb{R}^d : ||x|| < 1 \} \) is an open ball.

\[
\exp_B(v) = \text{cos}(||v||) v + \text{sin}(||v||) v / ||v|| \quad \text{and} \quad \log_B(v) = \sqrt{1 - ||v||^2} 2 v / ||v||.
\]

**Lorentz Model:** The Lorentz model avoids numerical instabilities that may arise with the Poincaré distance (Nicol and Kula, 2018).

\[
\exp_L(v) = \text{cosh}(||v||) v + \text{sinh}(||v||) v / ||v|| \quad \text{and} \quad \log_L(v) = \sqrt{1 + ||v||^2} - 1 / 2 ||v|| v / ||v||.
\]

**Molecular Structures**

We use ZINC (Irwin et al., 2012) for molecular property prediction, which has received attention as a good benchmark for supervised learning on graphs.

**Blockchain Transaction Graphs**

We study the problem of predicting price fluctuations for the underlying asset of the Ethereum blockchain (Wood et al., 2014).

**HGNN Multirelational Extension**

\[
\mathcal{H}^{t+1} = \sigma \left( \exp_{\mathcal{M}} \left( \sum_{j \in \mathcal{N}(i)} A_{ij} W_h \log_{\mathcal{M}}(\mathcal{H}_j) \right) + \sum_{k \in \mathcal{N}(i)} A_{ik} W_h \log_{\mathcal{M}}(\mathcal{H}_k) \right)
\]

**Table 1:** F1 (means) score and standard deviation of classifying synthetically generated graphs according to the underlying graph generation algorithm, we learn to distinguish three distinct graph generation algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Barabási-Albert</th>
<th>Watts-Strogatz</th>
<th>Erdős-Rényi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev</td>
<td>Test</td>
<td>Dev</td>
<td>Test</td>
</tr>
<tr>
<td>Euclidian</td>
<td>17.2 ± 0.12</td>
<td>96.0 ± 0.21</td>
<td>98.8 ± 0.17</td>
</tr>
<tr>
<td>Poincare</td>
<td>93.9 ± 0.15</td>
<td>95.9 ± 0.24</td>
<td>95.2 ± 0.06</td>
</tr>
<tr>
<td>Lorentz</td>
<td>94.1 ± 0.03</td>
<td>95.2 ± 0.25</td>
<td>94.6 ± 0.23</td>
</tr>
</tbody>
</table>

**Table 2:** Mean absolute error of predicting molecular properties logP, QED and SAS, as compared to current state-of-the-art deep learning methods. Scaled by 102 for table formatting (low is good).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>logP 3 5 10 20 256</th>
<th>QED 3 5 10 20 256</th>
<th>SAS 3 5 10 20 256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidian</td>
<td>6.7 ± 0.07 4.7 ± 0.07 4.7 ± 0.02 3.6 ± 0.00 3.6 ± 0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poincare</td>
<td>5.7 ± 0.00 4.6 ± 0.03 3.6 ± 0.02 3.2 ± 0.01 3.1 ± 0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lorentz</td>
<td>5.5 ± 0.02 4.5 ± 0.03 3.3 ± 0.03 2.9 ± 0.01 2.4 ± 0.02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 3:** Mean absolute error of predicting molecular properties logP, QED and SAS, as compared to current state-of-the-art deep learning methods. Scaled by 102 for table formatting (low is good).

**Blockchain Transaction Graphs**

We study the problem of predicting price fluctuations for the underlying asset of the Ethereum blockchain (Wood et al., 2014).