Harnessing Neighborhood Modeling and Asymmetry Preservation for Digraph Representation Learning

Honglu Zhou\textsuperscript{1} , Advith Chegu\textsuperscript{1} , Samuel S. Sohn\textsuperscript{1} , Zuohui Fu\textsuperscript{1} , Gerard de Melo\textsuperscript{2} and Mubbasir Kapadia\textsuperscript{1}

\textsuperscript{1}Rutgers University, Department of Computer Science, Piscataway, NJ, US
\textsuperscript{2}HPI / University of Potsdam, Potsdam, Germany
\{honglu.zhou, ac1771, zuohui.fu\}@rutgers.edu, gerard.demelo@hpi.de, \{sss286, mk1353\}@cs.rutgers.edu

Abstract

Digraph Representation Learning aims to learn representations for directed homogeneous graphs (digraphs). Prior work is largely constrained or has poor generalizability across tasks. Most Graph Neural Networks exhibit poor performance on digraphs due to the neglect of modeling neighborhoods and preserving asymmetry. In this paper, we address these notable challenges by leveraging hyperbolic collaborative learning from multi-ordered partitioned neighborhoods and asymmetry-preserving regularizers. Our resulting formalism, Digraph Hyperbolic Networks (D-HYPR), is versatile for multiple tasks including node classification, link presence prediction, and link property prediction. The efficacy of D-HYPR was meticulously examined against 21 previous techniques, using 8 real-world digraph datasets. D-HYPR statistically significantly outperforms the current state of the art. We release our code at https://github.com/hongluzhou/dhypr.

1 Introduction

Directionality is intrinsic to numerous real-world graphs [Ou et al., 2016]. Digraph Representation Learning (DRL) aims to learn representations for directed homogeneous graphs (digraphs) [Tong et al., 2020a; Zhang et al., 2021a]. Early DRL techniques include factorization and random walk-based approaches [Ou et al., 2016; Sun et al., 2019; Zhou et al., 2017; Khosla et al., 2019]. Yet, these methods face scalability issues or sensitivity to noise. Graph Neural Networks (GNNs) have seen recent success [Zhou et al., 2020], but they mainly focus on undirected graphs. There are two notable challenges that hinder their effectiveness on digraphs.

\textbf{Challenge 1: Neighborhood Modeling.} Node neighborhoods in a graph can carry distinct semantic meanings. Existing GNN techniques often simplify digraphs to undirected graphs or consider only direct out-neighbors [Kipf and Welling, 2016b; Veličković et al., 2017; Chami et al., 2019; Zhu et al., 2020], which can lead to a loss of original structure and ultimately subpar results on digraph-specific tasks.

\textbf{Challenge 2: Asymmetry Preservation.} Learning objectives based on symmetrical measures used by popular GNNs fail to capture asymmetric connections in digraphs [Salha et al., 2019]. Applications based on link prediction or graph topology learning are particularly affected when models fail to preserve digraph structural asymmetry.

Spectral-based DRL GNNs have sought to address the first challenge but struggle when applying models to graphs with different structures [Zhang et al., 2021a]. Solutions for the second challenge, such as viewing edge directions as edge features [Gong and Cheng, 2019] or parametrizing the node pair likelihood function by a neural network [Shi et al., 2019], neglect the first challenge. Moreover, prior DRL techniques are often constrained to directed acyclic graphs [Thost and Chen, 2021], are transductive [Sim et al., 2021], or lack broad applicability and generalizability across tasks [Sim et al., 2021; Tong et al., 2020b; Ma et al., 2019].

We propose \textbf{Digraph HYPERbolic Networks (D-HYPR)}, which use hyperbolic collaborative learning and asymmetry-preserving regularizers to tackle these challenges. Our approach comprises: (1) Modeling node neighborhoods using collaborative learning from multi-ordered and partitioned neighborhoods with larger receptive fields, (2) Using hyperbolic space to avoid distortion in neighborhood modeling, (3) Preserving asymmetry with socio-psychology-inspired regularizers, and (4) Ensuring flexibility through a message-passing-based GNN formalism for general digraphs.

Our contributions are three-fold: (1) D-HYPR that considers unique node neighborhoods and asymmetric relationships in digraphs, (2) Benchmarking across 8 real-world digraphs and 21 prior methods, revealing D-HYPR’s superiority, and (3) Capability of generating meaningful low-dimensional embeddings, an efficiency boon for large-scale applications.

2 Preliminaries

Let $G = (V, E)$ be a homogeneous graph with vertex set $V$ and edge set $E$: $e \in E$ is an ordered pair $e = (i, j)$ between vertices $i$ and $j$. The adjacency matrix of $G$ can be denoted as $A = \{0, 1\}^{V \times V}$. $G$ is a digraph when $e(i, j)$, $A_{i,j} \neq A_{j,i}$. Nodes are described by a feature matrix $X^{0,E} \in \mathbb{R}^{V \times d}$, i.e., each node $i \in V$ has a $d$-dimensional Euclidean feature $x_i^{0,E}$.
The superscript $E$ indicates that the vector lies in a Euclidean space, while $H$ denotes a hyperbolic vector. 0 denotes the input layer. The goal of DRL is to learn a mapping

$$f : \left(\mathbb{V}, \mathcal{E}, \{x_i^0, E\}_{i \in \mathbb{V}}\right) \rightarrow Z \in \mathbb{R}^{|\mathbb{V}| \times d'}$$

(1)

that maps nodes to low-dimensional ($d' \ll |\mathbb{V}|$) vectors.

The Poincaré ball model [Ganea et al., 2018] is defined by the $n$-dimensional manifold $\mathbb{H}^n = \{x \in \mathbb{R}^n : c\|x\| < 1\}$ equipped with the Riemannian metric: $g^0 = \lambda_x g^n$, where $\lambda_x := 2^{1-c\|x\|^2}$, $g^n = I_n$ is the Euclidean metric tensor, and $c > 0$ (we refer to $-c$ as the curvature). $\mathbb{H}^n$ is the open ball of radius $1/\sqrt{c}$. The connections between hyperbolic space and tangent space are established by the exponential map $\exp^n_x : T_x \mathbb{H}^n \rightarrow \mathbb{H}^n$ and logarithmic map $\log^n_x : \mathbb{H}^n \rightarrow T_x \mathbb{H}^n$:

$$\exp^n_x(v) = x \oplus c \left( \tanh \left( \frac{\sqrt{c} \|v\|}{2} \right) \frac{v}{\sqrt{c} \|v\|} \right)$$

(2)

$$\log^n_x(y) = \frac{2}{\sqrt{c}} \tanh^{-1} \left( \sqrt{c} \|y \oplus c x\| \right) \frac{y - x \oplus c x}{\|y - x \oplus c x\|}$$

(3)

where $x, y \in \mathbb{H}^n$, $v \in T_x \mathbb{H}^n$, and $\oplus c$ denotes Möbius addition, and

$$x \oplus c y := \frac{(1 + 2c(x, y) + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c(x, y) + c^2\|x\|^2\|y\|^2}$$

(4)

The Möbius scalar multiplication (Eq. 5) and Möbius matrix multiplication of $x \in \mathbb{H}^n \setminus \{0\}$ (Eq. 6) are

$$r \otimes_c x := \frac{1}{\sqrt{c}} \tanh \left( r \tanh^{-1} \left( \sqrt{c} \|x\| \right) \right) \frac{x}{\|x\|}$$

(5)

$$M \otimes_c x := \left( \frac{Mx}{\|x\|} \right) \tanh^{-1} \left( \sqrt{c} \|x\| \right) \frac{Mx}{\|Mx\|}$$

(6)

where $r \in \mathbb{R}$ and $M \in \mathbb{R}^{m \times n}$. The induced distance function on $(\mathbb{H}^n, g^n)$ is given by

$$d_{\mathbb{H}}(x, y) = (2/\sqrt{c}) \tanh^{-1} \left( \sqrt{c} \|x - y \oplus c x\| \right)$$

(7)

3 Methodology

Hyperbolic Embedding Learning. D-HYPR utilizes hyperbolic GNNs over Euclidean counterparts as the backbone for DRL. Given $G$ and $x_i^0, E$, we obtain $x_i^{\ell, H}$ by applying $\exp_0^\ell(\cdot)$, where $\exp_0$ is learned in training. Hyperbolic message passing (Eqs. 8 to 10) is then performed by multiple layers, forming the Hyperbolic Graph Embedding Layers. The layer is indexed by $\ell$, ranging from 1 to a pre-defined integer $l$.

(1) Hyperbolic Feature Transformation is performed by

$$m_i^{\ell, H} = W^{\ell} \otimes_{c^{\ell-1}} x_i^{\ell-1, H} \oplus_{c^{\ell-1}} b$$

(8)

where $W^{\ell} \in \mathbb{R}^{d^\ell \times d^{\ell-1}}$ is the weight matrix, and $b \in \mathbb{H}^{d^\ell}$ denotes the bias (both are learned).

(2) Hyperbolic Neighbor Aggregation results in $h_i^{\ell, H}$,

$$h_i^{\ell, H} = \exp_0^{\ell-1} \left( \sum_{j \in (1) \cup N(i)} c_{ij} \log_0^{\ell-1} \left( m_j^{\ell, H} \right) \right)$$

(9)

where $N(i) = \{j : (i, j) \in E\}$ denotes the set of neighbors of $i \in \mathbb{V}$. We apply out-degree normalization of the adjacency matrix to obtain the aggregation weights $c_{ij}$.

(3) Non-Linear Activation with Trainable Curvature. The output hyperbolic representation of node $i$ in layer $\ell$ is set as

$$x_i^{\ell, H} = \exp_0^\ell \left( \sigma \left( \log_0^{\ell-1} \left( h_i^{\ell, H} \right) \right) \right)$$

(10)

where $\sigma(\cdot)$ represents the ReLU non-linearity function.

Neighborhood Collaborative Learning. D-HYPR considers four primary neighborhood types. To achieve this, four types of $k$-order proximity matrix are defined:

(1) diffusion in $A_{d_{in}}^k$,

$$A_{d_{in}}^k(i, j) = \frac{1}{\sum_{p \in \mathbb{V}} A_{d_{in}}^{k-1}(i, p) \cdot A_{d_{in}}^k(p, j)}$$

(11)

where $A_{d_{in}}^1 = A$, is the inner product and $1$ is the indicator function. $A_{d_{in}}^k(i, j) = 1$ if there is a directed path from node $j$ to node $i$ of length exactly $k$.

(2) diffusion out $A_{d_{out}}^k$,

$$A_{d_{out}}^k(i, j) = \frac{1}{\sum_{p \in \mathbb{V}} A_{d_{out}}^{k-1}(i, p) \cdot A_{d_{out}}^k(p, j)}$$

(12)

where $A_{d_{out}}^1 = A$. $A_{d_{out}}^k(i, j) = 1$ if there is a directed path from node $i$ to node $j$ of length exactly $k$.

(3) common in $A_{c_{in}}^k$,

$$A_{c_{in}}^k(i, j) = \frac{1}{\sum_{p \in \mathbb{V}} A_{c_{in}}^{k-1}(i, p) \cdot A_{c_{in}}^k(p, j)}$$

(13)

where $i \neq j \neq p$. $A_{c_{in}}^k(i, j) = 1$ if node $i$ and node $j$ have a common in-neighbor $k$ hops away.

(4) common out $A_{c_{out}}^k$,

$$A_{c_{out}}^k(i, j) = \frac{1}{\sum_{p \in \mathbb{V}} A_{c_{out}}^{k-1}(i, p) \cdot A_{c_{out}}^k(p, j)}$$

(14)

where $i \neq j \neq p$. $A_{c_{out}}^k(i, j) = 1$ if node $i$ and node $j$ have a common out-neighbor $k$ hops away.

We compute these matrices for $k = 1$ to $K$ ($K$ is a hyperparameter) which replace the adjacency matrix as input to Hyperbolic Graph Embedding Layers that output $4K$ hyperbolic vectors. Subsequently, the hyperbolic average of the $4K$ vectors yields $z_i^{\text{fuse}}$. We then apply Eq. (9) with the learned curvature $-c^\ell$ and equal aggregation weights. The resulting output, $z_i^{H}$, is the final hyperbolic embedding of node $i$.

Asymmetry-Preserving Regularizers. We adopt the hyperbolic Fermi-Dirac decoder [Krioukov et al., 2010] to account for homophily [McPherson et al., 2001]. The decoder defines the likelihood of a node pair $(i, j)$ as

$$p(i, j) = \frac{1}{e^{\frac{\sigma^2(s_{i,j} - s_{j,i})^2}{t+1}} + 1}$$

(15)

where $r=2$ and $t=1$ (default), and $d^{\text{c}}(\cdot, \cdot)$ is defined in Eq. 7.

We then preserve the individual asymmetric node connectivity to account for preferential attachment [Mitzenmacher,
Figure 1: Neighborhood analyses. The common in/out neighborhood consists of more neighbors than diffusion in/out neighborhood that traditional methods typically use. The 8 digraph datasets demonstrate a clear scale-free characteristic for most neighborhoods.

Figure 2: Accuracy on the Semi-supervised Node Classification task. The embedding dimensionality is 32.

4 Experiments
In all tables, the best score is bolded, the second best is underlined, and the third best is in italic. Relative gains are computed as (BEST – SECOND)/SECOND. * indicates statistically superior performance of the best to the second best at a significance level of 0.001 using a standard paired t-test. Values after ± are standard deviations.

Neighborhood Analyses. We provide neighborhood analyses of datasets in Fig. 1, where pie charts show the ratio of the 4 types of neighborhoods in each dataset (K=1). Unlike the diffusion in/out neighborhood that traditional GNNs typically use, common in/out neighborhood consists of more neighbors, which suggests that neighborhood collaborative learning benefits from encoding additional context. For each neighborhood type, we also plot a histogram showing the distribution of the number of neighbors a node has over neighborhood.

Table 1: Results of Link Sign Prediction on the Wiki dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN [Kipf and Welling, 2016]</td>
<td>78.96 ± 0.4</td>
</tr>
<tr>
<td>GAT [Veličković et al., 2017]</td>
<td>79.38 ± 0.2</td>
</tr>
<tr>
<td>HGCN [Chami et al., 2019]</td>
<td>78.72 ± 0.0</td>
</tr>
<tr>
<td><strong>D-HYPR (ours)</strong></td>
<td>79.83 ± 0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN [Kipf and Welling, 2016]</td>
<td>78.76 ± 0.1</td>
</tr>
<tr>
<td>GAT [Veličković et al., 2017]</td>
<td>79.41 ± 0.2</td>
</tr>
<tr>
<td>HGCN [Chami et al., 2019]</td>
<td>79.23 ± 0.2</td>
</tr>
<tr>
<td><strong>D-HYPR (ours)</strong></td>
<td>79.47 ± 0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN [Kipf and Welling, 2016]</td>
<td>79.39 ± 0.1</td>
</tr>
<tr>
<td>GAT [Veličković et al., 2017]</td>
<td>79.66 ± 0.1</td>
</tr>
<tr>
<td>HGCN [Chami et al., 2019]</td>
<td>79.21 ± 0.2</td>
</tr>
<tr>
<td><strong>D-HYPR (ours)</strong></td>
<td>79.73 ± 0.2</td>
</tr>
</tbody>
</table>
We list the results in Table 2 and Table 3. One advantage of hyperbolic digraph embedding is low data distortion even with a low-dimensional embedding space. The superior performance of D-HYPR is evident—the highest relative gain of D-HYPR is 21.43% on AP over the Cora dataset. As the dimensionality increases, the gap from D-HYPR to the other methods decreases, but D-HYPR remains the best-performing method.

**Link Prediction (LP).** We list the results in Table 2 and Table 3. One advantage of hyperbolic digraph embedding is low data distortion even with a low-dimensional embedding space. The superior performance of D-HYPR is evident—the highest relative gain of D-HYPR is 21.43% on AP over the Cora dataset. As the dimensionality increases, the gap from D-HYPR to the other methods decreases, but D-HYPR remains the best-performing method.

**Semi-supervised Node Classification (NC).** We follow prior work [Grover and Leskovec, 2016] in reporting the results when the number of nodes labeled for training is varied between 1% and 10%. According to Fig. 2, D-HYPR consistently outperforms the baselines, and tends to perform well at fairly low label rates. D-HYPR statistically significantly outperforms the state-of-the-art (SOTA) methods.

**Link Sign Prediction (SP).** Table 1 reports the results which show D-HYPR is the most effective GNN model. Similar to LP and NC tasks, the effectiveness of D-HYPR is the most striking using a 4 dimensional embedding space.

Please refer to [Zhou et al., 2022] for additional details regarding the datasets and experimental setup, as well as additional analyses and comparisons with more SOTA techniques.

## 5 Conclusion

We propose D-HYPR: the Digraph HYPERbolic Network, as a novel GNN-based formalism for Digraph Representation Learning (DRL) by addressing Neighborhood Modeling and Asymmetry Preservation. Through rigorous evaluation, we empirically demonstrate the superiority of D-HYPR. D-HYPR retains effectiveness given a low budget of embedding dimensionality or labeled training samples, which is desirable for real-world applications.
Acknowledgments

The research was supported in part by NSF awards: IIS-1703883, IIS-1955404, IIS-1955365, RETTLE-2119265, and EAGER-2122119. This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 22STESS00001 01 01.

Disclaimer: The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

References


[Zhou et al., 2017] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for

