Harnessing Neighborhood Modeling and Asymmetry Preservation for Digraph Representation Learning*

Honglu Zhou[1], **Advith Chegu**[1], **Samuel S. Sohn**[1], **Zuohui Fu**[1],
Gerard de Melo[2] and **Mubbasir Kapadia**[1]
[1]Rutgers University, Department of Computer Science, Piscataway, NJ, US
[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.

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.

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 **D**igraph **HYPER**bolic 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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a homogeneous graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$; $e \in \mathcal{E}$ is an ordered pair $e = (i, j)$ between vertices i and j. The adjacency matrix of $\mathcal{G}$ can be denoted as $A = \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$. $\mathcal{G}$ is a digraph when $\exists (i, j), A_{i,j} \neq A_{j,i}$. Nodes are described by a feature matrix $X^{0,E} \in \mathbb{R}^{|\mathcal{V}| \times d}$, i.e., each node $i \in \mathcal{V}$ has a d-dimensional Euclidean feature $\mathbf{x}_i^{0,E}$.

*Published at the 31st ACM International Conference on Information & Knowledge Management - Received the Best Paper Award

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

$$f : \left(\mathcal{V}, \mathcal{E}, \left(\mathbf{x}_i^{0,E}\right)_{i\in\mathcal{V}}\right) \rightarrow Z \in \mathbb{R}^{|\mathcal{V}|\times d'} \tag{1}$$

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

The Poincaré ball model [Ganea *et al.*, 2018] is defined by the n-dimensional manifold $\mathbb{D}_c^n = \{x \in \mathbb{R}^n : c\,\|\mathbf{x}\| < 1\}$ equipped with the Riemannian metric: $g_{\mathbf{x}}^c = \lambda_{\mathbf{x}}^2 g^E$, where $\lambda_{\mathbf{x}} := \frac{2}{1-c\|\mathbf{x}\|^2}$, $g^E = \mathbf{I}_n$ is the Euclidean metric tensor, and $c > 0$ (we refer to $-c$ as the curvature). $\mathbb{D}_c^n$ is the open ball of radius $1/\sqrt{c}$. The connections between hyperbolic space and tangent space are established by the *exponential map* $\exp_{\mathbf{x}}^c : \mathcal{T}_{\mathbf{x}}\mathbb{D}_c^n \rightarrow \mathbb{D}_c^n$ and *logarithmic map* $\log_{\mathbf{x}}^c : \mathbb{D}_c^n \rightarrow \mathcal{T}_{\mathbf{x}}\mathbb{D}_c^n$:

$$\exp_{\mathbf{x}}^c(\mathbf{v}) = \mathbf{x} \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_{\mathbf{x}}^c\|\mathbf{v}\|}{2}\right)\frac{\mathbf{v}}{\sqrt{c}\,\|\mathbf{v}\|}\right) \tag{2}$$

$$\log_{\mathbf{x}}^c(\mathbf{y}) = \frac{2}{\sqrt{c}\lambda_{\mathbf{x}}^c}\tanh^{-1}\left(\sqrt{c}\,\|-\mathbf{x}\oplus_c\mathbf{y}\|\right)\frac{-\mathbf{x}\oplus_c\mathbf{y}}{\|-\mathbf{x}\oplus_c\mathbf{y}\|} \tag{3}$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{D}_c^n$, $\mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathbb{D}_c^n$, and $\oplus_c$ denotes *Möbius addition*, and

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

The *Möbius scalar multiplication* (Eq. 5) and *Möbius matrix multiplication* of $\mathbf{x} \in \mathbb{D}_c^n\backslash\{\mathbf{0}\}$ (Eq. 6) are

$$r \otimes_c \mathbf{x} := \frac{1}{\sqrt{c}}\tanh\left(r\tanh^{-1}(\sqrt{c}\,\|\mathbf{x}\|)\right)\frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{5}$$

$$M \otimes_c \mathbf{x} := (1/\sqrt{c})\tanh\left(\frac{\|M\mathbf{x}\|}{\|\mathbf{x}\|}\tanh^{-1}(\sqrt{c}\,\|\mathbf{x}\|)\right)\frac{M\mathbf{x}}{\|M\mathbf{x}\|} \tag{6}$$

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

$$d_{\mathbb{D}_c^n}(\mathbf{x},\mathbf{y}) = (2/\sqrt{c})\tanh^{-1}\left(\sqrt{c}\,\|-\mathbf{x}\oplus_c\mathbf{y}\|\right) \tag{7}$$

3 Methodology

Hyperbolic Embedding Learning. D-HYPR utilizes hyperbolic GNNs over Euclidean counterparts as the backbone for DRL. Given $\mathcal{G}$ and $\mathbf{x}_i^{0,E}$, we obtain $\mathbf{x}_i^{0,H}$ by applying $\exp_{\mathbf{0}}^{c^0}(\cdot)$, where c^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 ℓ, ranging from 1 to a pre-defined integer l.
(1) *Hyperbolic Feature Transformation* is performed by

$$\mathbf{m}_i^{\ell,H} = W^\ell \otimes_{c^{\ell-1}} \mathbf{x}_i^{\ell-1,H} \oplus_{c^{\ell-1}} \mathbf{b} \tag{8}$$

where $W^\ell \in \mathbb{R}^{F^\ell\times F^{\ell-1}}$ is the weight matrix, and $\mathbf{b} \in \mathbb{D}_{c^{\ell-1}}^{F^\ell}$ denotes the bias (both are learned).
(2) *Hyperbolic Neighbor Aggregation* results in $\mathbf{h}_i^{\ell,H}$,

$$\mathbf{h}_i^{\ell,H} = \exp_{\mathbf{0}}^{c^{\ell-1}}\left(\sum_{j\in\{i\}\cup\mathcal{N}(i)} e_{ij}\log_{\mathbf{0}}^{c^{\ell-1}}\left(\mathbf{m}_j^{\ell,H}\right)\right) \tag{9}$$

where $\mathcal{N}(i) = \{j : (i,j) \in \mathcal{E}\}$ denotes the set of neighbors of $i \in \mathcal{V}$. We apply out-degree normalization of the adjacency matrix to obtain the aggregation weights e_{ij}.
(3) *Non-Linear Activation with Trainable Curvatures.* The output hyperbolic representation of node i in layer ℓ is set as

$$\mathbf{x}_i^{\ell,H} = \exp_{\mathbf{0}}^{c^\ell}\left(\sigma\left(\log_{\mathbf{0}}^{c^{\ell-1}}\left(\mathbf{h}_i^{\ell,H}\right)\right)\right) \tag{10}$$

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 $A_{d_{in}}^k$,

$$A_{d_{in}}^k(i,j) = \mathbb{1}\left(\sum_{p\in\mathcal{V}} A_{d_{in}}^{k-1}(i,p)\cdot A_{d_{in}}^1(p,j)\right) \tag{11}$$

where $A_{d_{in}}^1 = A^\intercal$, $\cdot$ is the inner product and $\mathbb{1}$ 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 $A_{d_{out}}^k$,

$$A_{d_{out}}^k(i,j) = \mathbb{1}\left(\sum_{p\in\mathcal{V}} A_{d_{out}}^{k-1}(i,p)\cdot A_{d_{out}}^1(p,j)\right) \tag{12}$$

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 $A_{c_{in}}^k$,

$$A_{c_{in}}^k(i,j) = \mathbb{1}\left(\sum_{p\in\mathcal{V}} A_{d_{in}}^k(i,p)\cdot A_{d_{out}}^k(p,j)\right) \tag{13}$$

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 $A_{c_{out}}^k$,

$$A_{c_{out}}^k(i,j) = \mathbb{1}\left(\sum_{p\in\mathcal{V}} A_{d_{out}}^k(i,p)\cdot A_{d_{in}}^k(p,j)\right) \tag{14}$$

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 $\mathbf{z}_i^{\text{fuse}}$. We then apply Eq. (9) with the learned curvature $-c^l$ and equal aggregation weights. The resulting output, $\mathbf{z}_i^{l,H}$, 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

$$p(i,j)_f = \frac{1}{e^{\left(d_{\mathbb{D}_{c^l}^{d'}}\left(\mathbf{z}_i^{l,H},\mathbf{z}_j^{l,H}\right)^2 - r\right)/t} + 1}, \tag{15}$$

where r=2 and t=1 (default), and $d_{\mathbb{D}_{c^l}^{d'}}(\cdot,\cdot)$ is defined in Eq. 7.

We then preserve the individual asymmetric node connectivity to account for *preferential attachment* [Mitzenmacher,

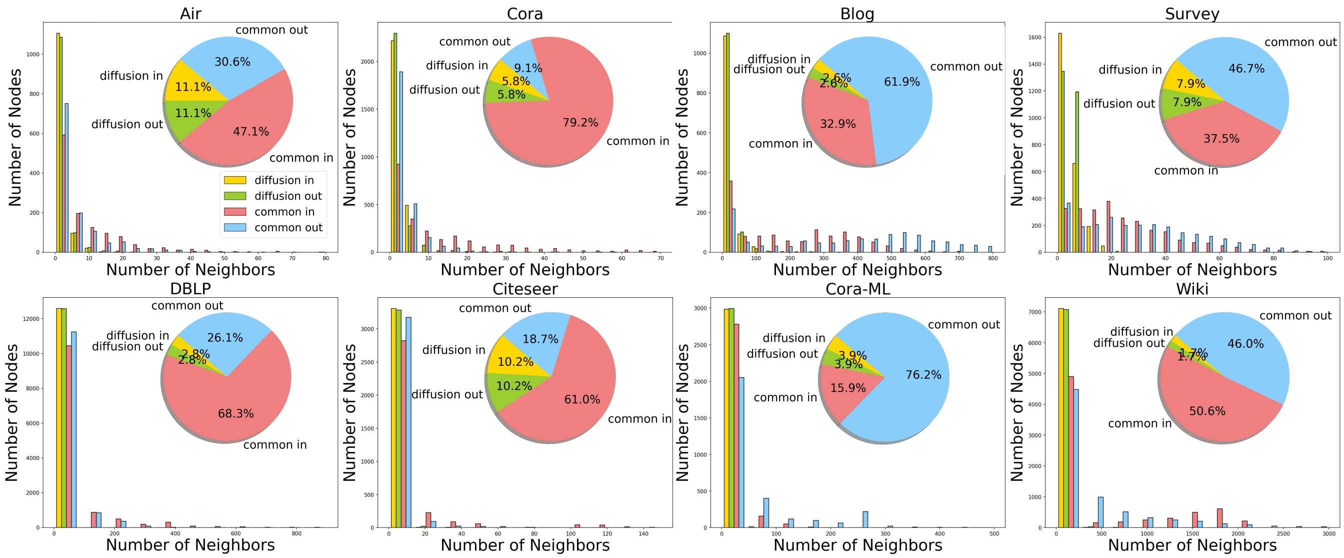

Figure 1: Neighborhood analyses. The `common in/out` neighborhood consists of more neighbors than `diffusion in/out` neighborhood that traditional methods typically use. The 8 digraph datasets demonstrate a clear scale-free characteristic for most neighborhoods.

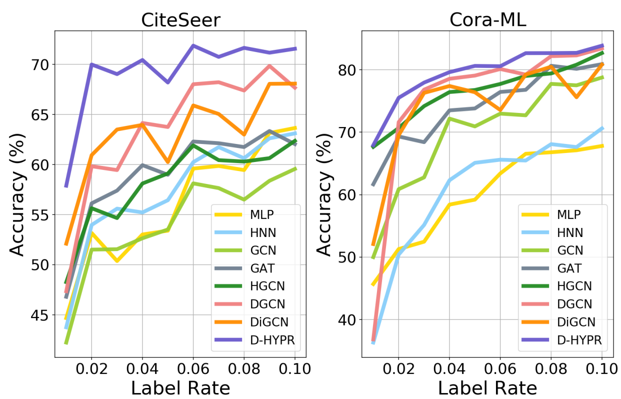

Figure 2: Accuracy on the *Semi-supervised Node Classification* task. The embedding dimensionality is 32.

4-Dim	GCN [Kipf and Welling, 2016a]	*78.96±0.4*
	GAT [Veličković *et al.*, 2017]	79.38±0.2
	HGCN [Chami *et al.*, 2019]	78.72 ± 0.0
	`D-HYPR` (ours)	***79.83±0.0**
8-Dim	GCN [Kipf and Welling, 2016a]	78.76 ± 0.1
	GAT [Veličković *et al.*, 2017]	79.41±0.2
	HGCN [Chami *et al.*, 2019]	*79.23±0.2*
	`D-HYPR` (ours)	***79.47±0.3**
32-Dim	GCN [Kipf and Welling, 2016a]	*79.39± 0.1*
	GAT [Veličković *et al.*, 2017]	79.66± 0.1
	HGCN [Chami *et al.*, 2019]	79.21 ± 0.2
	`D-HYPR` (ours)	***79.73 ± 0.2**

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

2004] by learning a mass for each node. This design [Salha *et al.*, 2019] is derived from Newton's theory of universal gravitation. To incorporate this design into `D-HYPR` based in hyperbolic space instead of Euclidean, we first perform $\mathbf{z}_i^{l,E} = \log_{\mathbf{0}}^{c^l}\left(\mathbf{z}_i^{l,H}\right)$, and then employ a Euclidean linear layer to learn $m_i \in \mathbb{R}$ (mass of node i). The likelihood of node pair (i, j) is computed by

$$p(i,j)_g = \gamma\left(m_j - \lambda \log\left(d_{\mathbb{D}^{d'}_{c^l}}(\mathbf{z}_i^{l,H}, \mathbf{z}_j^{l,H})^2\right)\right), \tag{16}$$

where $\gamma(\cdot)$ denotes the sigmoid function, and $\lambda \in \mathbb{R}$ is a hyper-parameter. $p(i,j)_g \neq p(j,i)_g$. Eqs. (16) and (15) both serve as self-supervised regularizers by minimizing the binary cross-entropy loss with negative sampling to estimate the likelihood of each node pair. However, Eq. (16) is employed one layer after where Eq. (15) is used. Thus, even though $d_{\mathbb{D}^{d'}_{c^l}}(\cdot,\cdot)$ also appears in Eq. (16), we find that Eq. (15) provides auxiliary guidance for the model to better construct the final hyperbolic embedding space.

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 $(\text{BEST} - \text{SECOND})/\text{SECOND}$. * 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 $\pm$ 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

Model (Dataset: Cora)	**4-Dim** AUC	AP	**8-Dim** AUC	AP
GCN [Kipf and Welling, 2016a]	65.92 (61.00)	65.92 (59.97)	70.89 (65.67)	71.26 (65.28)
VGAE [Kipf and Welling, 2016b]	63.86 (56.90)	63.86 (55.39)	66.60 (60.33)	66.60 (58.75)
GAT [Veličković *et al.*, 2017]	68.18 (64.73)	68.18 64.31)	72.70 (68.70)	73.93 (69.08)
Gravity GCN † [Salha *et al.*, 2019]	70.37 (65.80)	70.37 (64.65)	75.29 (71.85)	77.17 (72.50)
Gravity VGAE † [Salha *et al.*, 2019]	66.74 (61.79)	66.74 (60.61)	71.04 (65.45)	71.04 (64.15)
DGCN † [Tong *et al.*, 2020b]	75.33 (*71.88*)	71.95 (*68.58*)	79.01 (75.30)	79.01 (74.28)
DiGCN † [Tong *et al.*, 2020a]	70.61 (65.81)	67.11 (61.57)	74.63 (70.65)	74.88 (69.86)
MagNet † [Zhang *et al.*, 2021a]	*77.45* (55.93)	*79.32* (56.84)	77.46 (66.82)	76.59 (63.96)
HAT § [Zhang *et al.*, 2021b]	76.25 (72.84)	74.38 (70.27)	*82.58* (*77.82*)	*82.05* (*77.39*)
HGCN § [Chami *et al.*, 2019]	80.02 (67.37)	82.16 (66.66)	85.05 (83.07)	88.04 (84.63)
D−HYPR (ours) †§	**86.08** (***83.99**)	**88.74** (***85.33**)	**88.88** (***86.31**)	**91.13** (***87.76**)
Relative Gains (%)	7.57 (15.31)	8.01 (21.43)	4.5 (3.9)	3.51 (3.7)

Table 2: Results of *Link Prediction on Digraphs* with low-dimensional node embeddings. † denotes the method was designed for homogeneous digraphs (i.e., DRL), and § denotes the use of hyperbolic space. Results on each dataset of each method are from 100 experiments (10 unique train/test splits and 10 runs using different random seeds per split). We list the best and average results (average in brackets).

Model (Metric: AUC)	**Air**	**Cora**	**Blog**	**Survey**	**DBLP**
MLP	81.29 (76.52)	84.47 (81.67)	93.31 (92.48)	91.21 (89.98)	51.22 (49.98)
NERD † [Khosla *et al.*, 2019]	60.62 (56.39)	65.62 (62.02)	95.03 (94.00)	77.12 (69.30)	95.78 (95.37)
ATP † [Sun *et al.*, 2019]	68.99 (66.40)	88.47 (*86.44*)	85.05 (83.46)	73.53 (71.47)	60.43 (59.21)
APP † [Zhou *et al.*, 2017]	85.08 (82.72)	86.65 (85.50)	92.33 (91.65)	91.16 (90.34)	95.58 (95.33)
GCN [Kipf and Welling, 2016a]	76.71 (72.27)	80.77 (78.73)	91.87 (90.18)	89.29 (87.98)	92.98 (92.34)
VGAE [Kipf and Welling, 2016b]	77.79 (73.75)	80.80 (79.24)	92.25 (91.39)	90.07 (88.78)	93.36 (92.64)
GAT [Veličković *et al.*, 2017]	84.21 (80.24)	85.40 (82.58)	92.69 (89.95)	92.01 (91.05)	95.94 (95.62)
Gravity GCN † [Salha *et al.*, 2019]	85.16 (82.22)	85.62 (83.87)	95.11 (94.46)	91.63 (90.86)	96.89 (96.78)
Gravity VGAE † [Salha *et al.*, 2019]	83.98 (80.06)	87.17 (84.46)	96.15 (95.59)	91.64 (90.96)	95.98 (95.57)
DGCN † [Tong *et al.*, 2020b]	77.83 (73.68)	83.57 (81.34)	87.74 (86.74)	90.47 (89.49)	92.26 (91.83)
DiGCN† [Tong *et al.*, 2020a]	75.35 (71.27)	81.80 (78.90)	91.98 (90.50)	89.85 (88.17)	89.99 (89.72)
MagNet † [Zhang *et al.*, 2021a]	79.32 (75.58)	82.77 (71.90)	91.83 (90.81)	86.65 (84.81)	81.89 (80.57)
HNN § [Ganea *et al.*, 2018]	88.42 (*85.79*)	*88.75* (86.33)	*95.80* (*95.39*)	*92.07* (*91.39*)	*97.43* (*97.14*)
HGCN § [Chami *et al.*, 2019]	*88.26* (86.12)	89.24 (87.68)	95.64 (95.23)	92.15 (91.50)	97.54 (97.33)
D−HYPR (ours) †§	**89.07** (**86.33**)	**89.50** (***88.22**)	**96.19** (**95.62**)	**92.56** (***91.96**)	**97.66** (***97.38**)
Relative Gains (%)	0.74 (0.24)	0.29 (0.62)	0.04 (0.03)	0.44 (0.50)	0.12 (0.05)

Table 3: Results of *Link Prediction* with 32-dimensional node embeddings on more digraphs. Every result is from 100 experiments.

the entire graph. We observe asymptotical power-law node-degree distributions (i.e., scale-free) for most neighborhoods.

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 statistical 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 **D**igraph **HYPER**bolic 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, RETTL-2119265, and EAGER-2122119. This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 22STESE00001 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

[Chami *et al.*, 2019] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in neural information processing systems*, pages 4868–4879, 2019.

[Ganea *et al.*, 2018] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in neural information processing systems*, pages 5345–5355, 2018.

[Gong and Cheng, 2019] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9211–9219, 2019.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[Khosla *et al.*, 2019] Megha Khosla, Jurek Leonhardt, Wolfgang Nejdl, and Avishek Anand. Node representation learning for directed graphs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 395–411. Springer, 2019.

[Kipf and Welling, 2016a] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Kipf and Welling, 2016b] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[Krioukov *et al.*, 2010] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[Ma *et al.*, 2019] Yi Ma, Jianye Hao, Yaodong Yang, Han Li, Junqi Jin, and Guangyong Chen. Spectral-based graph convolutional network for directed graphs. *arXiv preprint arXiv:1907.08990*, 2019.

[McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[Mitzenmacher, 2004] Michael Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251, 2004.

[Ou *et al.*, 2016] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, 2016.

[Salha *et al.*, 2019] Guillaume Salha, Stratis Limnios, Romain Hennequin, Viet-Anh Tran, and Michalis Vazirgiannis. Gravity-inspired graph autoencoders for directed link prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 589–598, 2019.

[Shi *et al.*, 2019] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019.

[Sim *et al.*, 2021] Aaron Sim, Maciej L Wiatrak, Angus Brayne, Páidí Creed, and Saee Paliwal. Directed graph embeddings in pseudo-riemannian manifolds. In *International Conference on Machine Learning*, pages 9681–9690. PMLR, 2021.

[Sun *et al.*, 2019] Jiankai Sun, Bortik Bandyopadhyay, Armin Bashizade, Jiongqian Liang, P Sadayappan, and Srinivasan Parthasarathy. Atp: Directed graph embedding with asymmetric transitivity preservation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 265–272, 2019.

[Thost and Chen, 2021] Veronika Thost and Jie Chen. Directed acyclic graph neural networks. *International Conference on Learning Representations*, 2021.

[Tong *et al.*, 2020a] Zekun Tong, Yuxuan Liang, Changsheng Sun, Xinke Li, David Rosenblum, and Andrew Lim. Digraph inception convolutional networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[Tong *et al.*, 2020b] Zekun Tong, Yuxuan Liang, Changsheng Sun, David S Rosenblum, and Andrew Lim. Directed graph convolutional network. *arXiv preprint arXiv:2004.13970*, 2020.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Zhang *et al.*, 2021a] Xitong Zhang, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. Magnet: A magnetic neural network for directed graphs. *Advances in Neural Information Processing Systems*, 34, 2021.

[Zhang *et al.*, 2021b] Yiding Zhang, Xiao Wang, Chuan Shi, Xunqiang Jiang, and Yanfang Fanny Ye. Hyperbolic graph attention network. *IEEE Transactions on Big Data*, 2021.

[Zhou *et al.*, 2017] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[Zhou *et al.*, 2020] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[Zhou *et al.*, 2022] Honglu Zhou, Advith Chegu, Samuel S Sohn, Zuohui Fu, Gerard De Melo, and Mubbasir Kapadia. D-hypr: Harnessing neighborhood modeling and asymmetry preservation for digraph representation learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2732–2742, 2022.

[Zhu *et al.*, 2020] Shichao Zhu, Shirui Pan, Chuan Zhou, Jia Wu, Yanan Cao, and Bin Wang. Graph geometry interaction learning. In *Advances in Neural Information Processing Systems*, 2020.