HGENA: A Hyperbolic Graph Embedding Approach for Network Alignment

Fan Zhou*, Ce Li*, Xovee Xu*, Leyuan Liu*‡ and Goce Trajcevski†
*University of Electronic Science and Technology of China, Chengdu, China
†Iowa State University, IA, USA
‡Leyuan Liu is the corresponding author
Emails: fan.zhou@uestc.edu.cn, ce.lc@outlook.com, xovee@live.com, leyuanliu@uestc.edu.cn, gocet25@iastate.edu

***Abstract*—Cross-network alignment aims at identifying users who participate in different social networks, which benefits a variety of downstream social applications such as precise content delivery, fraud detection, and content/user recommender systems. Recent advances in network representations and graph neural networks have spurred various network structure-based methods for capturing underlying node similarities across social networks, thereby addressing the network alignment problem. However, most of the existing solutions rely on embedding methods that compute node similarity in Euclidean space, resulting in severe distortion or semantic loss when representing real-world social networks, which are usually scale free and with hierarchical structures. We address these issues by presenting a novel model: Hyperbolic Graph Embedding for Network Alignment (HGENA), which learns the structural semantics more efficiently by embedding nodes in hyperbolic space instead of Euclidean. HGENA overcomes the scalability issue since it requires far fewer dimensions in Riemannian manifolds and increases the capability of learning hierarchical structures, while enabling smaller distortion for tree-liked networks to facilitate node alignment. We also introduce alternative network mapping functions to compute node similarity across-network based on its distance on the Poincare ball. Experimental evaluations conducted on real world datasets demonstrate that HGENA achieves superior performance on social network alignment, especially for more tree-liked networks.**

***Index Terms*—Social networks, network alignment, hyperbolic space, hierarchical structure.**

I. Introduction

In recent years, we have witnessed the rapid growth of online social networks (OSN) such as Twitter, Facebook, TikTok, Instagram, Weibo, and Foursquare. Various OSNs may provide distinct social functions, e.g., people on these platforms posting microblogs, shopping goods, socializing with netizens, consuming and interacting online content such as news, music, and videos. Take Twitter and Foursquare as an example: a user can (re)tweet messages on Twitter but share their location-based activities on Foursquare. Users join in different networks for different social purposes, which raises a fundamental but challenging task in the research community, i.e., identifying and linking the same user across OSN platforms. This problem is also known as network alignment (NA) or user identity linkage in the literature [1]–[3], and plays a vital role in analyzing social networks and improving various applications such as cross-platform personalized recommendation, criminal behavior identification, mutual community detection, and user experience optimization [4]–[6].

This is the author version of the paper presented at IEEE GLOBECOM ’21
doi: 10.1109/GLOBECOM46510.2021.9685690

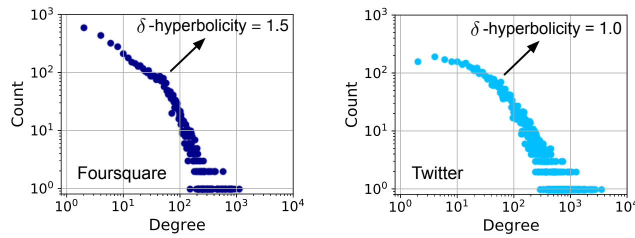

Fig. 1: Node degree distributions of two social networks.

Related work. Many research works have addressed the NA problem. Earlier studies typically focused on aligning the users by extracting informative features such as username, gender, behaviors, among many other user profiles [7], [8]. However, these features may not always be available due to privacy concerns. More importantly, these methods require manual feature engineering and effective features in one scenario may not generalize to other OSNs. Subsequently, structure-aware approaches [1], [2], [5], [6], [9]–[16], which aim at linking the users across OSNs based on the similarity of topological semantics, have been proposed to overcome this deficiency. These approaches are based on the fact that social relationships among users (e.g., follower and followee relationships) contribute to the network structures, which play an important role in addressing the NA problem. Inspired by the recent advances on deep graph representation techniques and graph neural networks (GNNs), some approaches embed user identities into low-dimensional feature vectors and then align them based on the similarity of their embeddings in Euclidean space. For example, DeepLink [2] samples a sequence of nodes from the graph and encodes the nodes using neural networks. dNAME applies graph convolution network (GCN) [17] to learn the structural similarities across OSNs while providing explainable alignment results using robust statistics. Recently, several models have begun to address different aspects of problems in NA, including efficiency [6], graph compression [3], community [15], [18], adversarial attack [16] etc.

Challenges. Although embedding-based methods have been successful in tackling the NA problem, the existing approaches capture the structure characteristic by learning network representations in the Euclidean space – which is restricted by the dimensions of the embedding space that may be prohibitively large to model complex relations. On the other hand, it has been shown that large-scale graphs such as social networks exhibit strong hierarchical structures since the node degrees follow power-law distributions [19], as shown in Fig. 1 – implying tree-liked structures of the networks. That is, a small fraction of users are very popular and attract more followers, while most of the others have very few followers. Nevertheless, the extremely imbalanced node distributions make the Euclidean space-based graph embedding distorted, because the number of nodes in Euclidean space increases polynomially with the radius, which restricts their ability to capture complex underlying hierarchical structures.

In this study, we present an alternative perspective to solve the NA problem and propose **HGENA** – Hyperbolic Graph Embedding for Network Alignment – a novel NA model that learns network embeddings and aligns users in the hyperbolic space. Our work is inspired by the recent success of hyperbolic geometric learning [20]–[22] in hierarchical structures. Specifically, HGENA embeds the networks in Riemannian manifold with constant negative curvature, where the number of nodes within a ball area grows exponentially with the radius. This property allows us to account for the power-law distribution of nodes and increases the model's capability to learn complex structures and patterns on large networks. The main contributions of this study are three-fold:

- We present a novel way of embedding heterogeneous networks and aligning users across social networks in the hyperbolic space, enabling more efficient structural pattern learning and capturing latent hierarchies for more accurate network alignment.
- We propose a novel network alignment model, which, to our knowledge, is the first hyperbolic graph embedding solution in literature. It preserves the distance between nodes and computes the mapping similarities using Poincaré ball distance that can be easily trained with gradient descent optimizations.
- We conducted extensive experiments on real-world datasets and the results show that HGENA outperforms state-of-the-art methods in both alignment accuracy and efficiency.

We note that hyperbolic geometry has been previously adopted to the NA problem, e.g., in PERFECT [18] and HUIL [23]. However, in contrast to previous works, we derive the hyperbolic embedding with *graph neural networks* and leverage the *true Fréchet Mean operation* to promote the anchor node representation. Additionally, spaces with different curvatures contribute differently to the hyperbolic representation learning, which is set to a fixed number in PERFECT and HUIL. To address this, we study the effect of curvature to the NA task's performance (cf. IV-4).

II. Preliminaries

In this section, we present the necessary background regarding Riemannian manifolds and basic operations for the Poincaré ball model of hyperbolic space, followed by the formal definition of a network alignment problem.

Hyperbolic geometry. There are three types of the Riemannian manifolds: Euclidean geometry (constant vanishing curvature), spherical geometry (constant positive curvature) and hyperbolic geometry (constant negative curvature). In this paper, we focus on hyperbolic space expanding faster (exponentially) than Euclidean spaces (polynomially) [24].

A hyperbolic space $\mathbb{H}^d$ is a d-dimensional Riemannian manifold with constant negative curvature c. Different from the Euclidean space $\mathbb{R}^d$, there are several isomorphic models in hyperbolic space $\mathbb{H}^d$, including the hyperboloid model, the Beltrami-Klein model, the Poincaré half-plane model and the Poincaré ball model $\mathcal{B}^d$. For simplicity, this study is based on the Poincaré ball model, due to its conformality and convenient parameterization [25].

Gyrovector Spaces. We use Möbius gyrovector space [20], [26] to describe basic operations (vector addition, subtraction and scalar multiplication) in Poincaré ball model.

Let $\mathcal{B}^n_K$ be the n-dimensional Poincaré ball model with curvature $K < 0$, which is the Riemannian manifold $(\mathcal{B}^n_K, g^{\mathcal{B}_K}_x)$, where $\mathcal{B}^n_K = \{x \in \mathcal{R}^n : \| x \| < -\frac{1}{K}\}$ is the open ball of radius $\frac{1}{\sqrt{|K|}}$. Its metric tensor is $g^{\mathcal{B}_K}_x = (\lambda^K_x)^2 g^E$, where $\lambda^K_x = \frac{2}{1+K\|x\|^2}$ is the conformal factor and $g^E = I_n$ is the Euclidean metric tensor [27]. The distance between two points $\mathbf{x}, \mathbf{y} \in \mathcal{B}^n_K$ is given by:

$$d^K_{\mathcal{B}}(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{|K|}} \operatorname{arccosh}\left(1 - \frac{2K\| \mathbf{x} - \mathbf{y} \|^2}{(1 + K\| \mathbf{x} \|^2)(1 + K\| \mathbf{y} \|^2)}\right) \tag{1}$$

and for $\mathbf{x}, \mathbf{y} \in \mathcal{B}^n_K$, the Möbius addition is defined as:

$$\mathbf{x} \oplus^K \mathbf{y} = \frac{(1 - 2K\langle \mathbf{x}, \mathbf{y} \rangle - K\| \mathbf{y} \|^2)\mathbf{x} + (1 + K\| \mathbf{x} \|^2)\mathbf{y}}{1 - 2K\langle \mathbf{x}, \mathbf{y} \rangle + K^2\| \mathbf{x} \|^2\| \mathbf{y} \|^2} \tag{2}$$

For the tangent vector $\mathbf{v} \neq 0$, and $\mathbf{x}, \mathbf{y} \in \mathcal{B}^n_K$ ($\mathbf{y} \neq 0$), the exponential map $\exp_{\mathbf{x}}(\mathbf{v}) : \mathcal{T}_{\mathbf{x}}\mathcal{B} \to \mathcal{B}$ and logarithmic map

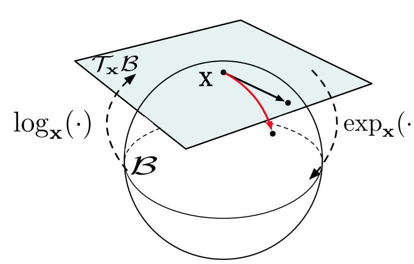

Fig. 2: The exponential and logarithmic maps.

$\log_{\mathbf{x}}(\mathbf{y}) : \mathcal{B} \rightarrow \mathcal{T}_{\mathbf{x}}\mathcal{B}$ in Poincaré ball model (see Fig. 2 for an example) are defined as:

$$\exp_{\mathbf{x}}^{K}(\mathbf{v}) = \mathbf{x} \oplus^{K} \left(\tanh\left(\sqrt{|K|} \frac{\lambda_{\mathbf{x}}^{K} \| \mathbf{v} \|}{2} \right) \frac{\mathbf{v}}{\sqrt{|K|} \| \mathbf{v} \|} \right), \tag{3}$$

$$\log_{\mathbf{x}}^{K}(\mathbf{y}) = \frac{2}{\sqrt{|K|} \lambda_{\mathbf{x}}^{K}} \tanh^{-1}\left(\sqrt{|K|} \| -\mathbf{x} \oplus^{K} \mathbf{y} \| \right) \frac{-\mathbf{x} \oplus^{K} \mathbf{y}}{\| -\mathbf{x} \oplus^{K} \mathbf{y} \|}. \tag{4}$$

A generalization of Euclidean operations to Poincaré ball model, e.g., Matrix-vector multiplication is provided in [20]:

$$A \otimes^{K} \mathbf{x} = \exp_{0}^{K}(A \log_{0}^{K}(\mathbf{x})), \tag{5}$$

and the bias translation can be computed as:

$$\mathbf{x} \oplus^{K} b = \exp_{\mathbf{x}}(P_{0 \rightarrow \mathbf{x}}^{K}(b)), \tag{6}$$

while the activation function is defined as:

$$\sigma^{K_1, K_2}(\mathbf{x}) = \exp_{0}^{K_1}(\sigma(\log_{o}^{K_2}(\mathbf{x}))). \tag{7}$$

We now provide formal definitions characterizing our problems:

*Definition 1 (**Hyperbolic Graph Representation Learning**):* Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with Euclidean space node features $(X_i^E)_{i \in \mathcal{V}}$, our purpose is to learn d-dimensional representation $(X_i^H)_{i \in \mathcal{V}}$ in the hyperbolic space, where the superscript ^{E} indicates features lie in the Euclidean space, and superscript ^{H} represents the hyperbolic space.

*Definition 2 (**Network Alignment**):* We have a pair of input networks $\mathcal{G}^s = \{\mathcal{V}^s, \mathcal{E}^s\}$ and $\mathcal{G}^t = \{\mathcal{V}^t, \mathcal{E}^t\}$, a set of observed anchor links $\mathcal{S}_{\text{anchor}} = \{(u, v) | u \in \mathcal{V}^s, v \in \mathcal{V}^t\}$, the task of network alignment is to predict those unobserved anchor links across the two social networks $\mathcal{G}^s$ and $\mathcal{G}^t$.

Note that the fully aligned paired networks are rare in real-world, we only study the partially aligned networks.

III. METHODOLOGY

We now present the details of HGENA, our hyperbolic graph embedding approach for network alignment. Focusing on aligning two networks, we show how to represent different networks using a hyperbolic graph convolutional neural network [21]. Moreover, with the hyperbolic embeddings of two networks in hand, we devised a novel method to learn the mapping across networks and find the corresponding anchor nodes based on the similarity in the hyperbolic space.

A. Hyperbolic Graph Network Representation

Given two graphs $\mathcal{G}^s$ and $\mathcal{G}^t$, it is crucial to capture the latent network structural semantics and to represent anchor nodes across the domains. An intuition solution is to directly learn node representations in hyperbolic space which, however, did not has closed-form solutions. Two frequently used operations are exponential and logarithmic maps, which can flexibly project representations between tangent and hyperbolic spaces.

1) Mapping Euclidean features to hyperbolic space: Previous network alignment studies [1], [3], [5] mainly paid attention on Euclidean feature extraction and node representation learning. Considering the extension of the pre-trained Euclidean embeddings and features, we first map them to the Poincaré ball manifold using the exponential map. Given Euclidean features $\mathbf{x}^E \in \mathbb{R}^n$ and the reference point $(o = 0) \in \mathcal{B}_K^n$ that used to perform tangent space operations, we have $\langle (0, \mathbf{x}^E), o \rangle = 0$. Therefore, we interpret $(0, \mathbf{x}^E)$ as a point in $\mathcal{T}_o\mathcal{B}_K^n$ and use Eq. (3) to map it to $\mathcal{B}_K^n$ as:

$$\mathbf{x}^{\mathcal{B}} = \exp_{o}^{K}((0, \mathbf{x}^E)) = \tanh(\sqrt{|K|} ||\mathbf{x}^E||) \frac{\mathbf{x}^E}{\sqrt{|K|} ||\mathbf{x}^E||}. \tag{8}$$

2) Hyperbolic graph convolution: With the obtained node features in hyperbolic space $\mathbf{x}^{\mathcal{B}}$ and two social networks $\mathcal{G}^s$ and $\mathcal{G}^t$, we would like to establish a model architecture that can learn the structures of the hierarchical network. Towards this goal, we learn node embeddings using a hyperbolic graph convolutional neural networks (HGCN) [21], which generalizes the Euclidean GCNs [17] in the hyperbolic space. It consists of three layers, including hyperbolic linear feature transformation, attention-based neighborhood aggregation, and non-linear activation:

(i) Hyperbolic linear feature transformation. The feature transformation layer maps the embedding space of one layer to the next layer, it use Eq. (5) and (6) to perform weight matrix multiplications and bias addition, respectively. Therefore, the representation updating rule of the l-th layer is:

$$h_i^l = (W^l \otimes^{K_{l-1}} \mathbf{x}_i^{l-1}) \oplus^{K_{l-1}} b^l, \tag{9}$$

where W is the weight matrix, b is the Euclidean bias vector, and $\mathbf{x}_i^{l-1}$ denotes the embedding of previous layer.

(ii) Attention-based neighborhood aggregation. Given a node $x_i \in \mathcal{B}_K^n$, in order to aggregate the neighborhood information of x_i, HGCNs first employ the logarithmic map to project neighbors $(x_j)_{j \in \mathcal{N}(i)}$ to tangent space at the center node x_i. Subsequently, the neighbors are aggregated with weights $(w_{ij})_{j \in \mathcal{N}(i)}$ in the tangent space:

$$\mathbf{y}_i = AGG^K(\mathbf{x}_i) = \exp_{\mathbf{x}_i}^{K} \left(\sum_{j \in \mathcal{N}(i)} w_{ij} \log_{\mathbf{x}_i}^{K}(\mathbf{x}_j) \right). \tag{10}$$

(iii) Non-linear activation. With the aggregated embedding $\mathbf{y}_i$, a hyperbolic non-linear activation $\sigma(\cdot)$ is computed as Eq. (7) in the tangent space:

$$\mathbf{x}_i^l = \sigma^{\otimes^{K_{l-1}, K_l}}(\mathbf{y}_i^l), \tag{11}$$

where K_{l-1} is the input hyperbolic space curvature and K_l is the output hyperbolic space curvature.

3) Improved neighborhood aggregation: The attention-based neighborhood aggregation used in HGCNs is the tangent space aggregation, which is a kind of pseudo-Fréchet means. It has been proved by [28] that applying the mean operation in the tangent space results in worse performance on downstream tasks than the true Fréchet mean [29]. We adopt an improved neighborhood aggregation layer which is based on *Fréchet mean neural network*. This layer considers the Fréchet mean as an $\arg\min$ operation. Let $x_i \in \mathcal{B}_K^n$ be a point in the Poincaré ball, $(x_j)_{j\in\mathcal{N}(i)}$ be the neighbors of x_i, $(w_{ij})_{(j\in\mathcal{N}(i))}$ be their weights, and further let their weighted Fréchet mean be the solution to the following optimization problem:

$$\mu_{fr} = \arg\min_{y\in\mathcal{B}_K^n} f(\mathbf{y}), \tag{12}$$

$$f(\mathbf{y}) = \sum_{j\in\mathcal{N}(i)} w_{ij} \cdot d_{\mathcal{B}^n}^K(\mathbf{x}_j, \mathbf{y})^2 = \sum_{j\in\mathcal{N}(i)} \frac{w_{ij}}{\sqrt{|K|}} \tag{13}$$

$$\operatorname{arccosh}^2\left(1 - \frac{2K\|\mathbf{x}_j - \mathbf{y}\|^2}{(1+K\|\mathbf{x}_j\|^2)(1+K\|\mathbf{y}\|^2)}\right).$$

B. Network Alignment in Hyperbolic Space

Through the hyperbolic graph network representation learning, we obtain the node embeddings of a pair of networks in hyperbolic space. The final step in HGENA is to align two networks using the learned embeddings. Given a node from one of the paired graph, we need to find the corresponding anchor node in another graph. However, the learned embeddings for two graphs lie in different hyperbolic spaces, requiring a hyperbolic mapping generator that projects two networks to the same representation space.

1) Hyperbolic mapping layer: In Euclidean space, the mapping function can be defined as $f : \mathbb{R}^m \to \mathbb{R}^n$, $f = \sigma(A\mathbf{x}+\mathbf{b})$ where $A \in \mathbb{R}^{n\times m}$, $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^n$ and σ is an activation function. Since learning the mapping function in the Euclidean space is not applicable to the embeddings $\mathbf{x}^s$ and $\mathbf{x}^t$ from hyperbolic space, we utilize the Poincaré ball multiplication and bias addition operations defined in Eq. (5) and (6) to perform hyperbolic mapping $g : \mathbb{B}^m \to \mathbb{B}^n$ as:

$$g = \sigma^K(A \otimes^K \mathbf{x} \oplus^K \mathbf{b}), \tag{14}$$

where $A \in \mathbb{R}^{n\times m}, \mathbf{x} \in \mathbb{B}^m, \mathbf{b} \in \mathbb{B}^n$. In this study, we use two layers to implement the hyperbolic mapping generator. The process of network embedding mapping in the hyperbolic space is illustrated in Fig. 3.

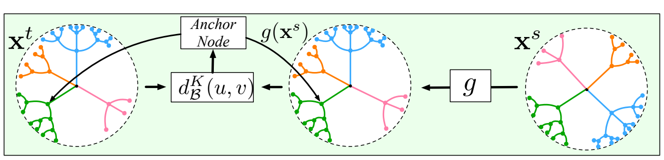

Fig. 3: Illustration of network mapping in the hyperbolic space.

2) Cross network information entropy: Comparing the similarities of the anchor node embeddings is the last step in the NA problem. We denote the observed source set as $\mathcal{S} \subset \mathcal{V}^s$ and $\mathcal{S} = \{u_1, u_2, \ldots, u_{|C|}\}$, the corresponding target set as $\mathcal{T} \subset \mathcal{V}^t$ and $\mathcal{T} = \{v_1, v_2, \ldots, v_{|C|}\}$. For a specific anchor node $u_i \in \mathcal{S}$, the node v_i with the same subscript as u_i is the corresponding anchor node. To find the real v_i for u_i, we compute all pairs of similarities between node u_i and the candidate node set $\mathcal{T}$. Note that we use Poincaré ball distance

TABLE I: Statistics of Two Datasets

Dataset	|V|	|E|	δ-hyperbolicity	# anchor links
Foursquare	5,313	76,972	1.5	3,148
Twitter	5,120	164,920	1.0	
Lastfm	2,138	4,259	3.0	1,561
MySpace	2,117	3,798	3.0	

Eq. (1) to measure the similarities between two nodes in the hyperbolic space. Therefore, a distance regularizer is used to cluster the anchor nodes together and push away others. To the end, the objective function for cross network alignment in HGENA is defined as:

$$\mathcal{L} = \sum_i d_{\mathcal{B}}^K(\mathbf{u}_i, \mathbf{v}_i) + \log \sum_i \sum_{j\neq i} \exp(-d_{\mathcal{B}}^K(\mathbf{u}_i, \mathbf{v}_j)). \tag{15}$$

IV. Experimental Results

In this section, we first present experimental settings and the analysis for dataset property. To demonstrate the effectiveness of HGENA, we then conduct extensive experiments and evaluate the performance of them.

Datasets We use the following two real-world social network datasets for evaluations. The statistics of them are described in Table I.

- Foursquare-Twitter (F-T): is published in [30], where nodes (users) of two social networks are partially aligned.
- Lastfm-MySpace (L-M): is published in [31] and publicly available at https://aminer.org/cosnet. Due to privacy concerns, it only provides partial anchor nodes.

To better understand the underlying structures of paired networks, we investigate the graph properties of Foursquare and Twitter. The undirected graph is established according to follower-followee relations. Previously we show the node degrees in two networks (cf. Fig. 1), which follows clear power-law distributions, i.e., a minority of nodes are connected to many neighbors (with higher degrees), while most are only linked to a few users. Further, the degree distributions reveal complex hierarchical structures as well as the scale-free property in two paired networks [32], [33]. This observation also proves our motivation to study the network alignment problem in hyperbolic space in a graph learning manner.

In addition to node degrees, we use Gromov δ-hyperbolicity [34] (a notion from geometric group theory) to measure how tree-like is the structure of the graph.

Baselines. We compare our proposed HGENA against the following five NA baselines:

- *IONE* [5]: models the follower and followee relationships of each user as the input and output context vectors. Then the representations of nodes are used for multiple network alignment.
- *DeepLink* [2]: users random walk and word2vec to capture the underlying network structures. The duality of pair-networks models the patterns of cross-site analysis.
- *GCN* [17]: provides node representation learning through spectral graph convolutional networks. In this study, we

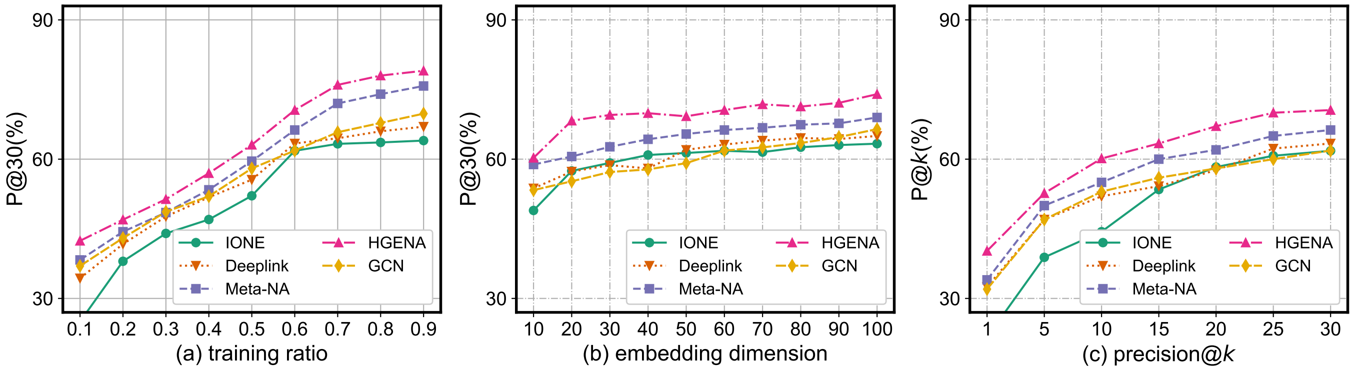

Fig. 4: Performance comparisons on Twitter-Foursquare with different (a) training ratio, (b) dimensions, and (c) precision@k.

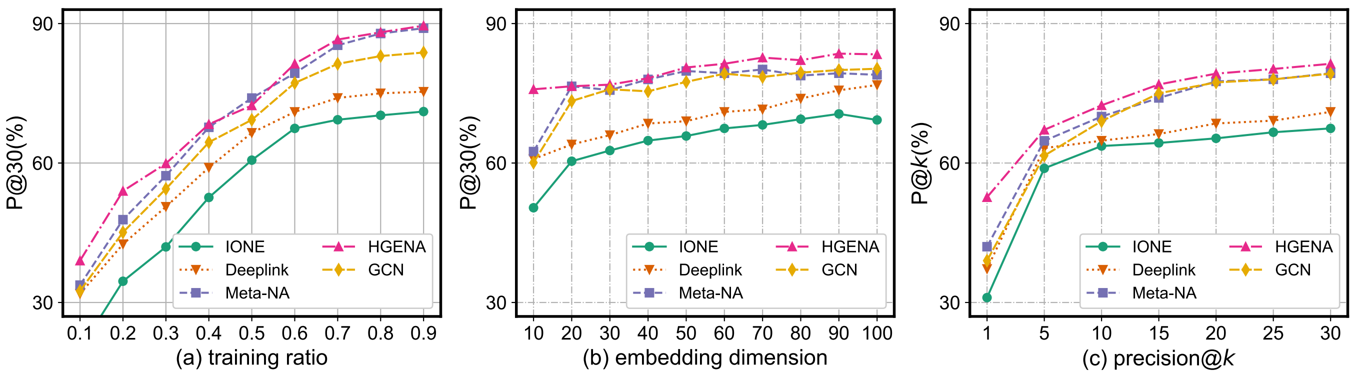

Fig. 5: Performance comparisons on Lastfm-Myspace with different (a) training ratio, (b) dimensions, and (c) precision@k.

use GCN to learn the embeddings for each node in paired networks and then map the anchor nodes by comparing their embedding vectors.

- *Meta-NA* [6]: considers NA problem as one-shot classification problem and conducts a graph meta-learning to tackle the multi-network alignment task flexibly.

Note that we omit comparison with PERFECT model [18] since we lack community labels of paired networks. All deep learning based models are trained with Adam optimizer, default embedding dimension is set to 60, and learning rate is chosen from $10^{\{-2,-3,-4,-5\}}$. Other parameters for each model follow the reported settings in corresponding paper.

Evaluation metrics. Recall that our primary goal is to retrieve the latent anchor nodes in $\mathcal{G}^t$ for $\mathcal{G}^s$. To evaluate the alignment accuracy in line with the baseline models, we use a widely used metric – Precision@k ($P@k$) – to assess the model performance, which is defined as follows:

$$P@k = \sum_{i}^{n} \mathbf{1}_i\{\text{success}@k\}/n. \quad (16)$$

Note that the higher the value, the better the performance.

Experimental Results. Now we discuss the experimental observations, shown in Fig. 5. We compare HGENA to several baseline approaches and study the influence of training data ratio (left), embedding dimension (middle), and the value of k (right). Overall, we can see that HGENA outperforms all baselines consistently in all the settings on both datasets.

1) Performance on different training ratio: The training ratio indicates the percentage of anchor nodes used for training. As illustrated in Fig. 4(a) and Fig. 5(a), the performances of all methods increase with the training data ratio. This result is intuitive as more local structure semantics of anchor nodes are utilized for training the models. Besides, our model requires less training data than baselines since it embeds the network in the hyperbolic space – which could better reflect the tree-liked structure than in Euclidean space. On the other hand, previous methods heavily rely on more anchor nodes to capture the sub-graph patterns. However, anchor nodes are expensive in real-world OSNs due to privacy protection policies. Labeling anchor nodes needs domain knowledge, which restricts the models relying on massive anchor nodes, let alone the label noises that would further deteriorate the model performance due to the introduced biases.

2) Performance on different embedding dimension: According to Fig. 4(b) and Fig. 5(b), we can observe that all Euclidean embedding based methods perform worse than HGENA when the embedding dimension is low. Because the complexities of graph embedding algorithms are highly dependent on spatial dimensions, the baselines rely on large dimensions to represent the complex network structures in the Euclidean space [21]. In contrast, HGENA requires fewer dimensions since the number of nodes in a certain volume increases exponentially with the radius, thereby encoding the imbalanced node distribution information without embedding distortion. This result also suggests that our approach is scalable and more efficient in embedding and aligning larger-scaled networks.

3) Performance on different precision@k: Fig. 4(c) and Fig. 5(c) show the different values of parameter k on two datasets. As we can see, previous network alignment approaches cannot effectively retrieve the anchor nodes when k is small. Our model, in contrast, achieves better anchor node ranking performance, because it explicitly encodes the latent hierarchy of the networks, which, arguably, can better reflect the local structures of anchor nodes.

Another observation is that the improvement on the dataset Lastfm-Myspace with high δ-hyperbolicity is limited, comparing to the Twitter-Foursquare. This suggests the hyperbolic space learning is more suited for the tree-like network structure, which indicates higher hyperbolicity (low δ). But for small dimensions, HGENA is still more capable for anchor node representation than GCN and any other Euclidean methods.

TABLE II: Influence of Curvature c

Variable c	**Twitter-Foursquare**		**Lastfm-MySpace**	
	p@10	p@30	p@10	p@30
$c = 0.5$	0.51	0.64	0.60	0.68
$c = 1.0$	**0.57**	**0.71**	0.68	0.75
$c = 10.0$	0.48	0.62	0.69	0.78
$c = 25.0$	0.46	0.61	**0.72**	**0.81**

4) Effect of variable curvature: Curvature c is an essential parameter of HGENA, as it measures how the embedding space we learned deviates from the flat planes [21], [35], and then determines the effect of hyperbolic space learning. Table II shows how c influences HGENA on two datasets, indicating that c would affect the model's performance differently in practice. For example, HGENA achieves the best performance when $c = 1.0$ on Twitter-Foursquare and $c = 25.0$ on Lastfm-MySpace, suggesting that the parameter c is highly dependent on the network structures. Generally, a smaller value is better for graphs exhibiting a stronger hierarchy.

V. Conclusion

We presented HGENA, a deep hyperbolic graph neural network model for network alignment, catering to both structural properties and imbalanced node distributions in social networks. We enabled embedding social networks and aligning users across domains in the hyperbolic space. Providing an alternative solution for network alignment beyond Euclidean space embedding and node linking, HGENA may provide foundation for studies that will capture more meaningful structural patterns and complex node semantics. As our future work, we are interested in exploiting rich node features to improve alignment performance, and incorporating the uncertainty [36]. In addition, we will extend the proposed model to other graph-related tasks in social networks, such as information cascade modeling and fake news detection.

Acknowledgement

Research supported in part by the National Natural Science Foundation of China (Grant No.62072077 and 62176043) and the NSF SWIFT grant 2030249.

References

[1] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng, "Predict anchor links across social networks via an embedding approach." in *IJCAI*, 2016, pp. 1823–1829.
[2] F. Zhou, L. Liu, K. Zhang, G. Trajcevski, J. Wu, and T. Zhong, "DeepLink: A deep learning approach for user identity linkage," in *INFOCOM*, 2018, pp. 1313–1321.
[3] K. K. Qin, F. D. Salim, Y. Ren, W. Shao, M. Heimann, and D. Koutra, "G-CREWE: graph compression with embedding for network alignment," in *CIKM*, 2020, pp. 1255–1264.
[4] J. Zhang and P. S. Yu, "Community detection for emerging networks," in *SDM*, 2015, pp. 127–135.
[5] L. Liu, W. K. Cheung, X. Li, and L. Liao, "Aligning users across social networks using network embedding." in *IJCAI*, 2016, pp. 1774–1780.
[6] F. Zhou, C. Cao, G. Trajcevski, K. Zhang, T. Zhong, and J. Geng, "Fast network alignment via graph meta-learning," in *INFOCOM*, 2020, pp. 686–695.
[7] R. Zafarani and H. Liu, "Connecting corresponding identities across communities," in *AAAI*, 2009.
[8] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, "Exploiting innocuous activity for correlating users across sites," in *WWW*, 2013.
[9] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen, "Mapping users across networks by manifold alignment on hypergraph," in *AAAI*, 2014.
[10] L. Liu, X. Li, W. K. Cheung, and L. Liao, "Structural representation learning for user alignment across social networks," *TKDE*, pp. 1824–1837, 2019.
[11] M. Heimann, H. Shen, T. Safavi, and D. Koutra, "REGAL: representation learning-based graph alignment," in *CIKM*, 2018, pp. 117–126.
[12] C. Li, S. Wang, Y. Wang, P. S. Yu, Y. Liang, Y. Liu, and Z. Li, "Adversarial learning for weakly-supervised social network alignment," in *AAAI*, 2019, pp. 996–1003.
[13] A. Cheng, C. Zhou, H. Yang, J. Wu, L. Li, J. Tan, and L. Guo, "Deep active learning for anchor user prediction," in *IJCAI*, 2019, pp. 2151–2157.
[14] F. Zhou, Z. Wen, G. Trajcevski, K. Zhang, T. Zhong, and F. Liu, "Disentangled network alignment with matching explainability," in *INFOCOM*, 2019, pp. 1360–1368.
[15] H. Chen, H. Yin, X. Sun, T. Chen, B. Gabrys, and K. Musial, "Multi-level graph convolutional networks for cross-platform anchor link prediction," in *KDD*, 2020.
[16] Z. Zhang, Z. Zhang, Y. Zhou, Y. Shen, R. Jin, and D. Dou, "Adversarial attacks on deep graph matching," in *NeurIPS*, 2020.
[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[18] L. Sun, Z. Zhang, J. Zhang, F. Wang, Y. Du, S. Su, and P. S. Yu, "PERFECT: A hyperbolic embedding for joint user and community alignment," in *ICDM*, 2020, pp. 501–510.
[19] F. Zhou, X. Xu, G. Trajcevski, and K. Zhang, "A survey of information cascade analysis: Models, predictions, and recent advances," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 27:1–27:36, 2021.
[20] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," in *NeurIPS*, 2018, pp. 5345–5355.
[21] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *NeurIPS*, 2019, pp. 4868–4879.
[22] Çaglar Gülçehre, M. Denil, M. Malinowski, A. Razavi, R. Pascanu, K. Hermann, P. Battaglia, V. Bapst, D. Raposo, A. Santoro, and N. D. Freitas, "Hyperbolic attention networks," in *NeurIPS*, 2019.
[23] F. Wang, L. Sun, and Z. Zhang, "Hyperbolic user identity linkage across social networks," in *GLOBECOM*, 2020.
[24] F. Sala, C. D. Sa, A. Gu, and C. Ré, "Representation tradeoffs for hyperbolic embeddings," in *ICML*, 2018, pp. 4460–4469.

[25] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *NeurIPS*, 2017, pp. 6338–6347.
[26] A. A. Ungar, "A gyrovector space approach to hyperbolic geometry," *Synthesis Lectures on Mathematics and Statistics*, vol. 1, no. 1, pp. 1–194, 2008.
[27] Q. Liu, M. Nickel, and D. Kiela, "Hyperbolic graph neural networks," in *NeurIPS*, 2019, pp. 8230–8241.
[28] A. Lou, I. Katsman, Q. Jiang, S. J. Belongie, S.-N. Lim, and C. D. Sa, "Differentiating through the fréchet mean," in *ICLR*, 2020.
[29] M. R. Fréchet, "Les éléments aléatoires de nature quelconque dans un espace distancié," *Annales de l'institut Henri Poincaré*, vol. 10, pp. 215–310, 1948.
[30] J. Zhang and P. S. Yu, "Pct: Partial co-alignment of social networks," in *WWW*, 2016.
[31] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "Cosnet: Connecting heterogeneous social networks with local and global consistency," in *KDD*, 2015.
[32] Barabási and Albert, "Emergence of scaling in random networks," *Science*, vol. 286 5439, pp. 509–12, 1999.
[33] E. Ravasz and A.-L. Barabási, "Hierarchical organization in complex networks." *Physical Review. E*, vol. 67 2 Pt 2, p. 026112, 2003.
[34] E. Jonckheere, P. Lohsoonthorn, and F. Bonahon, "Scaled gromov hyperbolic graphs," *Journal of Graph Theory*, vol. 57, pp. 157–180, 2008.
[35] J. W. Robbin and D. A. Salamon, "Introduction to differential geometry," *ETH, Lecture Notes, preliminary version*, 2011.
[36] L. Sun, Z. Zhang, J. Zhang, F. Wang, H. Peng, S. Su, and P. S. Yu, "Hyperbolic variational graph neural network for modeling dynamic graphs," 2021.