HyperRole: Hyperbolic Graph Transformer for Role Discovery in Online Social Networks

Huijun Tang*, Ming Du*, Pengfei Jiao*, Huaming Wu† and Zhidong Zhao*
*School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
†Center for Applied Mathematics, Tianjin University, Tianjin 300072, China
Emails: {tanghuijune, mdu, pjiao}@hdu.edu.cn, whming@tju.edu.cn, zhaozd@hdu.edu.cn

IEEE INFOCOM 2025 - IEEE Conference on Computer Communications | 979-8-3315-4305-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOM55648.2025.11044749

***Abstract*—Role discovery assist in various applications of online social networks, such as water army detection, shopping recommendation, rumor tracing, etc. However, existing studies often overlook the significance of hierarchical structures in online social networks, which are crucial for understanding the roles played by different users. To address this gap, we propose a novel approach based on hyperbolic graph learning, called HyperRole, which effectively leverages the hierarchical structure of online social networks for role discovery. HyperRole first extracts structural features from users and constructs user sequences based on feature similarity, capturing the relationships between users across different scales. Then, we learn role information from structural features by hyperbolic graph Transformer to embed users into the hyperbolic space, preserving the hierarchical structure between users and enabling interactions between users of the same level that are far away from each other. Additionally, we leverage the hierarchical distance between the target user and other users within the same sequence to guide and modify the role information of the target user. Based on the generated user role embeddings, we train a multi-class classifier to classify roles. Extensive experiments on several real-world network datasets demonstrate that our model outperforms existing baseline methods, showcasing its superior performance.**

***Index Terms*—Online Social Networks, Role Discovery, Hierarchical Structure, Hyperbolic Graph Transformer**

I. Introduction

The rapid development of online social networks brought about a diversity of complex systems, e.g., currency trading systems, recommender systems, etc. To address the real-world challenges [1]–[3] posed by these complex systems, it is necessary to gain a deeper understanding of the users within the network. The concept of roles [4], [5] points out that the behavioral logic of users and their interactions are closely related to their roles within the network. Users positioned at the center of the network typically wield significant influence, while those connecting different regions serve as bridges [6]. Role discovery [7] has thus become a key area of study in online social networks, aiming to delve into the roles and functions of individual users to better understand the network's intrinsic evolutionary mechanisms.

Initial research [8]–[10] on role discovery utilized matrix decomposition methods, constructing feature matrices from

This work is supported by the Zhejiang Provincial Natural Science Foundation of China under Grant LDT23F01012F01 and by the National Natural Science Foundation of China under Grant Number 62401190, 62372146 and 62071327. Corresponding author: Pengfei Jiao.

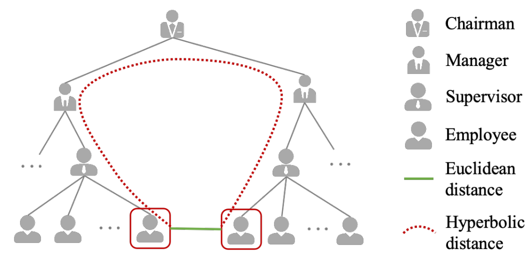


Fig. 1. Comparison of embedding distances for the hierarchical structure of hyperbolic and Euclidean spaces.

user attributes and network structure information. Various matrix decomposition techniques are then employed to analyze the role features of users from these matrices. Although these methods are computationally efficient and have low time complexity, the derived role features heavily depend on prior assumptions and are sensitive to noise present in online social networks. Random walk-based methods [11]–[14] utilize well-designed random walk strategies to obtain corresponding random walk sequences for each user, distinguishing the roles of different users by aggregating features along these sequences. While the generalization ability of random walks can effectively mitigate noise, the accuracy of role discovery is highly dependent on the parameter selection for the random walk strategy, which often involves high complexity. The development of graph learning [15] brings more expressive solutions to online social networks. Graph learning based methods [16]–[22] utilize Graph Neural Networks (GNNs) [23] and other graph learning models to learn complex network information hidden within the network to generate user embeddings. For solving the role discovery problem, these methods constrain the learning process to ensure that the generated user embeddings include rich role features, offering superior fitting and generalization capabilities.

Although existing graph learning-based methods achieve significant success, there are still two problems that need to be solved:

- **Distance distortion of hierarchical structure:** The hierarchical structure of online social networks can not be expressed non-destructively [24]. Even though advanced role discovery methods embed network information into a low-dimensional space and retain the original structural properties through graph learning models, they still can not completely avoid the problem of distance distortion due to over-squeezing [25]. As shown in Fig. 1, the chairman user has a large number of employee users, and embedding these employee users in the Euclidean space will result in their Euclidean distances being too close due to the problem of over-squeezing, which fails to correctly express the distances between these users.

- **Neglect of Long-distance dependencies in hierarchical structures**: Since online social networks have a hierarchical structure, some users with the same roles are distributed in the same stratum. However, due to the different parent nodes, it becomes challenging for the GNN to capture the connection between these users with the same roles but far away from each other.

Hyperbolic geometry, which studies spaces with constant negative curvature, offers powerful modeling capabilities for complex networks [26], [27]. To preserve the hierarchical structure of online social networks in the embedding space, we utilize a graph learning model to embed user feature information into hyperbolic space, thereby improving the accuracy of role discovery. Hyperbolic space is a Riemannian manifold with constant negative curvature, that better represents and distinguishes nodes across different hierarchies, effectively preserving the hierarchical structure of the original network and avoiding distance distortion. Unfortunately, most existing hyperbolic graph learning models [28]–[30] based on the improvement of GNNs inherit the limitation of GNNs, such as only obtaining information from neighborhoods. This limitation fails to address the problem of long-distance dependencies in hierarchical structures. To overcome this challenge, we propose leveraging Transformer's ability to capture global information. This approach enables the identification of connections between users with the same roles who are distant from each other within the hierarchical structure, thereby addressing the shortcomings of existing role discovery methods.

In this paper, we introduce a novel simplified hyperbolic graph Transformer model, called HyperRole, designed to address the role discovery problem in online social networks. HyperRole utilizes hyperbolic geometry to preserve the hierarchical structure of online social networks in hyperbolic space, effectively distinguishing different roles within the network. The combination of hyperbolic geometry and graph Transformer [31] gives the model the ability to allow long-distance users in the hierarchy to interact with information in the hyperbolic space. Specifically, HyperRole first extracts real statistical information from user information and network structure as structural features of users by feature extraction methods, which are categorized into local and higher-order features. These structural features reflect the role of users to some extent. Then, respectively, user sequences are constructed based on the similarity of the structural features to learn the local and higher-order relationships among users independently. By feeding these user sequences into the hyperbolic graph learning model, HyperRole learns the user embeddings from the structural features of users and mines the essential role information. Additionally, HyperRole designed a readout module to guide other users in the same sequence to adaptively integrate role features to the target user according to the hierarchical distance between them. Relying on the generated user embeddings, we train a multi-classifier to determine the role of the user. Our main contributions can be summarized as follows:

- We propose the first hierarchical structure role discovery method based on hyperbolic geometry in online social networks, aimed at preserving the hierarchical relationships among users in hyperbolic space. By leveraging the property of hyperbolic space, we can effectively differentiate between different hierarchies of users to more accurately recognize their roles in the network.
- We propose a Hyperbolic Graph Transformer framework to extract structural features, which enables remote users to interact with information in hyperbolic space. In particular, we construct hyperbolic distance-aware readout functions to guide the target user to learn role features based on hierarchical distances from other users.
- We evaluate the performance of the proposed HyperRole in role discovery tasks on several real-world networks. Extensive experiments justify our consideration of preserving the hierarchical structure in the network and the superiority of the model for role discovery.

II. Preliminaries

A. Problem Statement

An online social network is represented by an undirected unweighted graph $G = \{U, E\}$ where $U = \{v_1, v_2, \cdots, v_M\}$ is the set of M nodes and $E \subseteq U \times U$ is the set of edges between nodes. The set of neighbors of node v is defined as $N(v) = \{u \in U | (u, v) \in E\}$. z_i is the attribute vector of the user v_i and $z_i \in \mathbb{R}^{d_i}$. The goal of the role discovery problem is to extract useful role information from the network structure and node information and to classify users in the network.

B. Hyperbolic Geometry

Before introducing the Hyperbolic Graph Transformer, we first present some necessary definitions related to hyperbolic geometry. We denote the Minkowski inner product as $\langle ., . \rangle_{\mathcal{L}} : \mathbb{R}^{d+1} \times \mathbb{R}^{d+1} \to \mathbb{R}$ and $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} := -x_0 y_0 + x_1 y_1 + ... + x_d y_d$. Then we define $\mathbb{H}^{d,c}$ and $\mathcal{T}_{\mathbf{x}}\mathbb{H}^{d,c}$ as follows:

$$\mathbb{H}^{d,c} : = \{\mathbf{x} \in \mathbb{R}^{d+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -c, x_0 > 0\}, \quad (1)$$

$$\mathcal{T}_{\mathbf{x}}\mathbb{H}^{d,c} : = \{\mathbf{v} \in \mathbb{R}^{d+1} : \langle \mathbf{v}, \mathbf{x} \rangle_{\mathcal{L}} = 0\}, \quad (2)$$

where $\mathbb{H}^{d,c}$ indicates the hyperboloid manifold within d dimensions with constant negative curvature $-1/c$ $(c > 0)$, $\mathcal{T}_{\mathbf{x}}\mathbb{H}^{d,c}$ denotes the tangent space centered at point $\mathbf{x}$ that performs Euclidean operations undefined in hyperbolic space, and $||\mathbf{v}||_{\mathcal{L}} = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle_{\mathcal{L}}}$ denotes the norm of $\mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathbb{H}^{d,c}$. We define the intrinsic distance function between two points $\mathbf{x}, \mathbf{y}$ in $\mathbb{H}^{d,c}$ as follows:

$$D_{\mathcal{L}}^{c}(\mathbf{x}, \mathbf{y}) = \sqrt{c}\,\mathrm{arcosh}(-\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}/c), \quad (3)$$

where $\mathrm{arcosh}(\cdot)$ is the inverse hyperbolic function.

We can establish a mapping between tangent space and hyperbolic space by exponential and logarithmic maps, which are only defined locally in general Riemannian manifolds while forming a bijection between the hyperbolic space and the tangent space at a point in the hyperbolic space. Let $\mathbf{o} := \{\sqrt{c}, 0, \ldots, 0\} \in \mathbb{H}^{d,c}$ denote the reference point in $\mathbb{H}^{d,c}$ to perform tangent space operations. Let $\mathbf{x}$ be a point in $\mathcal{T}_{\mathbf{o}}\mathbb{H}^{d,c}$ and map it by $\exp_{\mathbf{o}}^{K}(\cdot)$ to $\mathbb{H}^{d,c}$ with:

$$\exp_{\mathbf{o}}^{K}(\mathbf{x}) = \sqrt{c}\cosh(\frac{||\mathbf{x}||_2}{\sqrt{c}}), \sqrt{c}\sinh\left(\frac{||\mathbf{x}||_2}{\sqrt{c}}\right)\frac{\mathbf{x}}{||\mathbf{x}||_2}, \tag{4}$$

where $\cosh(\cdot)$ is the hyperbolic cosine function and $\sinh(\cdot)$ is the sine function. Similarly, we assume $\mathbf{y}$ as a point in $\mathbb{H}^{d,c}$ and map it by $\log_{\mathbf{o}}^{K}(\cdot)$ to $\mathcal{T}_{\mathbf{o}}\mathbb{H}^{d,c}$ with:

$$\log_{\mathbf{o}}^{K}(\mathbf{y}) = D_{\mathcal{L}}^{c}(\mathbf{o}, \mathbf{y})\frac{\mathbf{y} + \frac{1}{c}\langle \mathbf{o}, \mathbf{y}\rangle_{\mathcal{L}}\mathbf{o}}{||\mathbf{y} + \frac{1}{c}\langle \mathbf{o}, \mathbf{y}\rangle_{\mathcal{L}}\mathbf{o}||_{\mathcal{L}}}, \tag{5}$$

Since the tangent space $\mathcal{T}_{\mathbf{o}}\mathbb{H}^{d,c}$ is Euclidean and isomorphic to $\mathbb{R}$, we replace $\mathcal{T}_{\mathbf{o}}\mathbb{H}^{d,c}$ with $\mathbb{R}$ in the following for better readability and comprehensibility, without strictly distinguishing between them.

C. Overview of HyperRole

An overview of HyperRole is illustrated in Fig. 2.

Network Modeling. Given online social network data, which includes user information and user connection data, the network is modeled as graph data. User information serves as user attributes, while user connection data represents the network structure.

Structural Feature Extraction. Based on the generated graph, two structural feature extraction methods are applied to all users, extracting both local structural features and higher-order structural features to learn users' role features (Sec. III-A).

Hierarchical Structure Learning. Then, in order to effectively learn and utilize hierarchical structures in online social networks, we propose a simplified hyperbolic Transformer-based user embedding method, which effectively preserves hierarchical structures in the hyperbolic space and utilizes the Transformer and hierarchical distances to guide users to learn user embeddings from users at the same hierarchical level (Sec. III-B).

Role Discovery. Leveraging structural features for constraints, we generate user role embeddings with role features through the graph learning model. We regard the role discovery task in online social networks as the node classification task in the graph, resulting in role discovery for all users (Sec. III-C).

TABLE I
NOTATIONS AND THEIR DEFINITIONS.

Notation	Definition
$G(U,E)$	The network with users U and their edges E
M	The number of nodes
N	The length of user sequence
x^l	Local feature embeddings in Euclidean spaces
x^h	High-order feature embeddings in Euclidean spaces
d	The dimension of a vector
z_i	The attributes of user v_i
H	User embeddings in hyperbolic spaces
$(\cdot)^l, (\cdot)^h$	l, h corresponds to local and higher order
ℓ, L	The number of layers
σ	A nonlinear activation function
c	The reciprocal of the curvature of a hyperbolic space
$\mathbb{H}^{d,c}$	The hyperboloid manifold within d dimensions with constant negative curvature $-1/c$ ($c > 0$),
$\mathcal{T}_{\mathbf{x}}\mathbb{H}^{d,c}$	The tangent space centered at point $\mathbf{x}$
$\otimes^c, \oplus^c$	The multiplication and addition defined in $\mathbb{H}^c$
$\langle \cdot, \cdot \rangle_{\mathcal{L}}$	The Minkowski inner product
$D_{\mathcal{L}}^c(\mathbf{x}, \mathbf{y})$	The intrinsic distance between two points $\mathbf{x}, \mathbf{y}$ in $\mathbb{H}^{d,c}$

III. HyperRole

In this section, we present HyperRole, a role discovery approach based on hyperbolic graph Transformers.

A. Structural Feature Extraction

Extracting structural features of users in the network to identify the roles of users is a general approach for role discovery. To improve the accuracy of role discovery, we extract local and higher-order features of the users in order to facilitate the graph learning model to learn more essential features of the roles. It is important to emphasize that, to accurately depict each user's role and learn role features in graph learning, the structural features must be based on real statistical information rather than low-dimensional embeddings derived from learning.

1) Local feature extraction: The role of a user in the social network is significantly influenced by connected users, and a user who connects to a large number of other users is extremely influential on a local scale. Therefore, we capture the local features of users to recognize the local role of users. Specifically, we utilize the ReFeX [32] method, a widely validated neighborhood information extraction method. ReFeX aggregates neighborhood features by simple mathematical computation over multiple iterations with binning and single encoding for each dimension, ultimately generating local features of users. We define x_i^l as the local embedding of v_i and $x_i^l \in \mathbb{R}^{d_l}$, where $i = 1, 2, \ldots, M$.

2) Higher-order feature extraction: Although local features reflect to some extent the roles of users on a local scale, the highly complex structure of social networks may lead to many users obtaining overly homogeneous local features. To address this problem, we distinguish the roles played by different users in similar network structures by recognizing the complex structures in which the users are located. In particular, we utilize the Graphlet Degree Vector (GDV) [33] method to generate higher-order features of users. GDV classifies node orbits by enumerating subgraph structures and partitioning them based on graph self-isomorphisms and then counts user occurrences in different orbits to represent higher-order features of users. We define x_i^h as the local embedding of v_i and $x_i^h \in \mathbb{R}^{d_h}$, where $i = 1, 2, \ldots, M$.

B. Hierarchical Structure Learning

After extracting the structural features of users in the social network, we built a graph learning module, the simplified

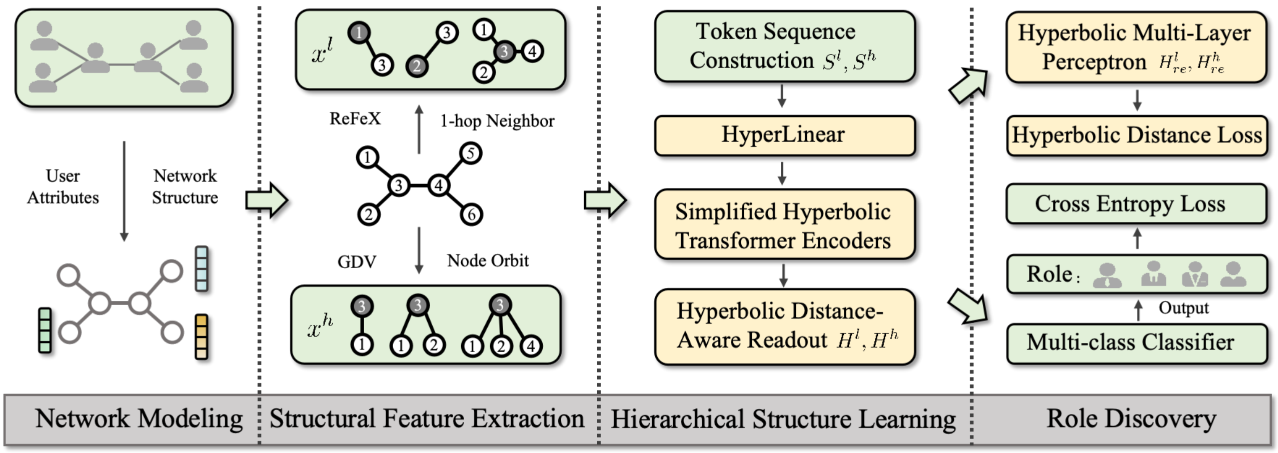


Fig. 2. The framework of HyperRole. HyperRole models online social networks as graph data, extracting the structural features of users. It generates node embeddings through a hierarchical structure learning module and learns users' role features from these structural features, which are then fed into a multi-class classifier for role discovery.

hyperbolic graph Transformer skeleton, for learning more essential role information from complex network structures and structural features in the social network.

1) Token Sequence Construction: Graph Transformers typically input all user features in the graph as token sequences, however, this behavior is unreasonable given the permutation invariance of users. Meanwhile, the proximity of relationships between users in social networks is ignored in the simple arrangement of token sequences. For example, in the token sequence, the board chairman and the department manager should be closer to each other, and should be farther away from the ordinary employees.

To avoid unnecessary connections between unrelated users and to learn local and higher-order features separately during the graph learning process, we construct two independent token sequences based on the similarity of structural features. For each node v_i, we define the set of the top N nodes whose embeddings are most similar to v_i's embedding as $\text{Top}_N(v_i)$:

$$\text{Top}_N(v_i) = \{v_j \in U \setminus v_i \mid \text{rank}(\text{Sim}(x_i^l, x_j^l)) \leq N\}, \quad (6)$$
$$\text{Top}_N^h(v_i) = \{v_j \in U \setminus v_i \mid \text{rank}(\text{Sim}(x_i^h, x_j^h)) \leq N\}, \quad (7)$$

where $\text{Sim}(a, b) = a \cdot b/(|a| \cdot |b|)$ indicates the similarity between a and b, and $\text{rank}(\cdot)$ indicates the position in the sorted list of the similarity, with the smallest distance having the highest rank. We construct the token sequences as follows:

$$S_i^l = \{z_i\} \cup \{z_n \mid v_n \in \text{Top}_N^l(v_i)\}, \quad (8)$$
$$S_i^h = \{z_i\} \cup \{z_n \mid v_n \in \text{Top}_N^h(v_i)\}, \quad (9)$$

where $i = 1, 2, \ldots, M$, $n = 1, 2, \ldots, N$, and $S_i^l, S_i^h \in \mathbb{R}^{(N+1)\times d_i}$ are token sequences of user v_i. To facilitate our discussion, when constructing S_i^l and S_i^h, we place the attribute z_i corresponding to v_i in the first positions of the sets, denoted as z_0, and we then insert the attributes of the points in the $\text{Top}_N^l(v_i)$ and $\text{Top}_N^h(v_i)$ sets into the sets S_i^l and S_i^h in descending order of similarity, and we record these indices as p_i^l and p_i^h respectively.

2) HyperLinear: The HyperLinear module learns low-dimensional embeddings that capture role information through graph learning. The embedding space is transformed from Euclidean space to Hyperbolic space. For input token sequence S_i^l and S_i^h, the HyperLinear module map them as follows:

$$H_i^{l,0} = \exp_{\mathbf{o}}^{c_0}(S_i^l) \otimes^{c_0} W^{l,0}, \quad (10)$$
$$H_i^{h,0} = \exp_{\mathbf{o}}^{c_0}(S_i^h) \otimes^{c_0} W^{h,0}, \quad (11)$$

where $W^{l,0}, W^{h,0} \in \mathbb{H}^{d_i \times d, c_0}$ is the learnable projection of the input layer of the HyperLinear module in the hyperboloid manifold within $d_i \times d$ dimensions with constant negative curvature $-1/c_0$, $\exp_{\mathbf{o}}^{c_0}(\cdot)$ is the operation of the exponential map transferring the embedding from $\mathbb{R}^d$ at the origin to $\mathbb{H}^{d,c_0}$, and $S \otimes^{c_0} W := \exp_{\mathbf{o}}^{c_0}(W \log_{\mathbf{o}}^{c_0}(S))$ is the multiplication defined in hyperbolic space.

3) Simplified Hyperbolic Transformer Encoder: In this module, we develop a simplified Transformer encoder in hyperbolic space, as shown in Fig. 3. Unlike the traditional Transformer, we omit the normalization operation, since normalization in hyperbolic space remains an unsolved problem. Experimental results indicate that the model achieves excellent performance even without normalization. Specifically, we rewrite the hyperbolic version of the Multi-head Self-Attention (MSA) by definition:

$$Q = H \otimes^c W^Q, K = H \otimes^c W^K, V = H \otimes^c W^V, \quad (12)$$
$$\text{MSA}(H) = softmax(\langle Q, K \rangle_{\mathcal{L}}) \otimes^c V, \quad (13)$$

where $\langle \cdot, \cdot \rangle_{\mathcal{L}}$ denotes the Minkowski inner product used to compute the similarity between two hyperbolic vectors and $W^Q, W^K, W^V \in \mathbb{H}^{d \times d, c}$ are learnable weight matrices.

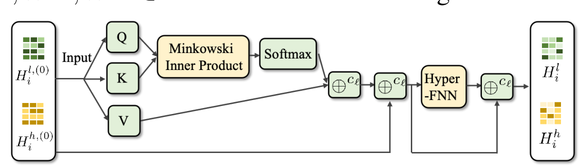


Fig. 3. The architecture of the simplified hyperbolic Transformer encoder.

Compared to traditional self-attention, we replace the original dot product operation with the Minkowski inner product and discard the degree-based scaling because this scaling is not significant for hyperbolic embeddings. The feedforward network that consists of hyperbolic linear transformations, called HyperFNN, is as follows:

$$\mathrm{FFN}(H) = (W \otimes^{c_\ell} H) \oplus^{c_\ell} b, \tag{14}$$

where $K_\ell, K_{\ell+1}$ represents the hyperbolic space curvature of the $\ell, \ell+1$-th layer. $W \in \mathbb{H}^{d\times d, c_\ell}$ and $b \in \mathbb{H}^{(N+1)\times d, c_\ell}$ are the learnable weight matrix and bias. $H \oplus^{c_\ell} b := \exp_{\mathbf{o}}^{c_\ell}(P^{c_\ell}_{\mathbf{o}\longrightarrow H}(b))$ is the addition defined in hyperbolic space. $P^{c_\ell}_{\mathbf{o}\longrightarrow H}(\cdot)$ is the parallel transport from $\mathcal{T}_{\mathbf{o}}\mathbb{H}^{d,c_\ell}$ to $\mathcal{T}_H\mathbb{H}^{d,c_\ell}$. The summarized proposed simplified hyperbolic Transformer encoder is as follows, take user v_i as an example:

$$H_i^{\prime l,(\ell)} = \mathrm{MSA}(H_i^{l,(\ell-1)}) \oplus^{c_\ell} H_i^{l,(\ell-1)}, \tag{15}$$

$$H_i^{l,(\ell)} = \sigma^{\otimes^{c_\ell, c_{\ell+1}}}(\mathrm{FFN}(H_i^{\prime l,(\ell)}) \oplus^{c_\ell} H_i^{\prime l,(\ell)}), \tag{16}$$

where $H_i^{l,(\ell)}$ denotes the output of the ℓ-th Transformer encoder. $\oplus^{K_\ell}$ is an addition operation defined in $\mathbb{H}^{d,c_\ell}$ and $\sigma^{\otimes^{c_\ell, c_{\ell+1}}} := \exp_{\mathbf{o}}^{c_{\ell+1}}(\sigma(\log_{\mathbf{o}}^{c_\ell}(S)))$ is nonlinear activations transferring the embedding from $\mathbb{H}^{d,c_\ell}$ to $\mathbb{H}^{d,c_{\ell+1}}$. We can obtain $H_i^{h,(\ell)}$ in the same process. Then we denote the local and high-order embedding sequences of the Hyperbolic Transformer encoder module as H^l and H^h, which are as follows:

$$H^l = (H_1^{l,(\ell)}, H_2^{l,(\ell)} \dots, H_M^{l,(\ell)}), \tag{17}$$

$$H^h = (H_1^{h,(\ell)}, H_2^{h,(\ell)} \dots, H_M^{h,(\ell)}), \tag{18}$$

Through this Transformer encoder, we enable long-distance users in the social network to associate as well. Leveraging the property of negative curvature in hyperbolic space, the model accurately learns low-dimensional embeddings of users, maintaining appropriate distances and preserving the original hierarchical structure in the embedding space.

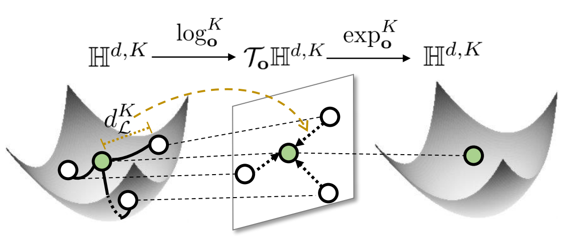


Fig. 4. The overview of hyperbolic distance-aware readout.

4) Hyperbolic Distance-Aware Readout: After the Transformer encoder, we obtain an embedding sequence corresponding to each user, containing relevant user feature information. To further enhance the learned role features, this module utilizes the hierarchical distances between users for guidance, as shown in Fig. 4, so that the target user adaptively learns the relationships between different users in the social network, resulting in the final user embedding. Here, we calculate the hyperbolic distance $D_{\mathcal{L}}^c(\cdot,\cdot)$ between user embeddings to measure the hierarchical distance. For the embedding sequence H_i^l corresponding to user v_i, the Hyperbolic Distance-Aware Readout module is as follows:

$$g_i^l = \log_{\mathbf{o}}^{c_\ell}(H_i^l), \tag{19}$$

$$\alpha_{i,n}^l = \frac{\exp(D_{\mathcal{L}}^c(g_{i,0}^l, g_{i,n}^l)W_a^\top)}{\sum_{n=1}^N \exp(D_{\mathcal{L}}^c(g_{i,0}^l, g_{i,n}^l)W_a^\top)}, \tag{20}$$

$$H_{i,out}^l = \exp_{\mathbf{o}}^{c_\ell}(\sum_{n=0}^N \alpha_{i,n} g_i^l), \tag{21}$$

where $i = 0, 1, \dots, M$, $n = 1, 2, \dots, N$, $\log_{\mathbf{o}}^{c_\ell}(\cdot)$ is the operation of the exponential map transferring the embedding from $\mathbb{H}^{d,c_\ell}$ to $\mathbb{R}^d$ at the origin, $g_i^l \in \mathbb{H}^{d\times N, c_\ell}$, $W_a \in \mathbb{R}^{1\times 1}$ denotes the learnable weight matrix, $exp(\cdot)$ is the exponential operation, and $h_{i,out}^l \in \mathbb{H}^{d,c_\ell}$ denotes the user embedding of v_i learned by the hyperbolic distance-Aware readout module. Let $H_{out}^l = (H_{1,out}^l, H_{2,out}^l, \dots, H_{M,out}^l)$. We can obtain H_{out}^h by the same process and $H_{out}^l, H_{out}^h \in \mathbb{H}^{M\times d, c_\ell}$.

C. Role Discovery

As outlined in Section III-B, HyperRole generates the corresponding user embeddings H_{out}^l, H_{out}^h by constructing two independent token sequences and learning the local and higher-order role features of the users from relevant users across the entire social network based on the simplified hyperbolic Transformer encoder. We reconstruct the role information in the user embeddings by a Hyperbolic Multi-Layer Perceptron (HMLP) similar to (14), which is as follows:

$$H_{re}^l = \mathrm{HMLP}(H_{out}^l) = (W \otimes^{c_\ell} H_{out}^l) \oplus^{c_\ell} b, \tag{22}$$

$$H_{re}^h = \mathrm{HMLP}(H_{out}^h) = (W \otimes^{c_\ell} H_{out}^h) \oplus^{c_\ell} b, \tag{23}$$

where $H_{re}^l \in \mathbb{H}^{M\times d_l, c_\ell}, H_{re}^h \in \mathbb{H}^{M\times d_h, c_\ell}$ are reconstructed local and higher-order features. To make user embeddings learn real structural features, we constrain the distance between user embeddings and structural features. As the user embeddings exist in a hyperbolic space while the structural features are in an Euclidean space, we map the structural features to hyperbolic space and compute the hyperbolic distance between the user embedding and the structural features. The Hyperbolic Distance loss $\mathcal{L}_1$ is as follows:

$$\mathcal{L}_1 = D_{\mathcal{L}}^c(H_{re}^l, \exp_{\mathbf{o}}^c(X^l)) + D_{\mathcal{L}}^c(H_{re}^h, \exp_{\mathbf{o}}^c(X^h)), \tag{24}$$

Finally, we train a multi-class classifier based on logistic regression for role discovery on the basis of $H_{out}^l \oplus^c H_{out}^h$ obtained. The classifier optimizes its own fitting curve with the training data, making it possible to output the corresponding roles based on the input user embeddings. The classifier is trained by the cross entropy loss function:

$$\mathcal{L}_2 = -\frac{1}{M}\sum_{i=1}^M [y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)], \tag{25}$$

where y_i is the true user role of user v_i, $\hat{y}_i$ is the user role of user v_i predicted by the classifier, and $log(\cdot)$ is the ogarithmic operation. The pseudocode of the HyperRole framework is summarized in Algorithm 1.

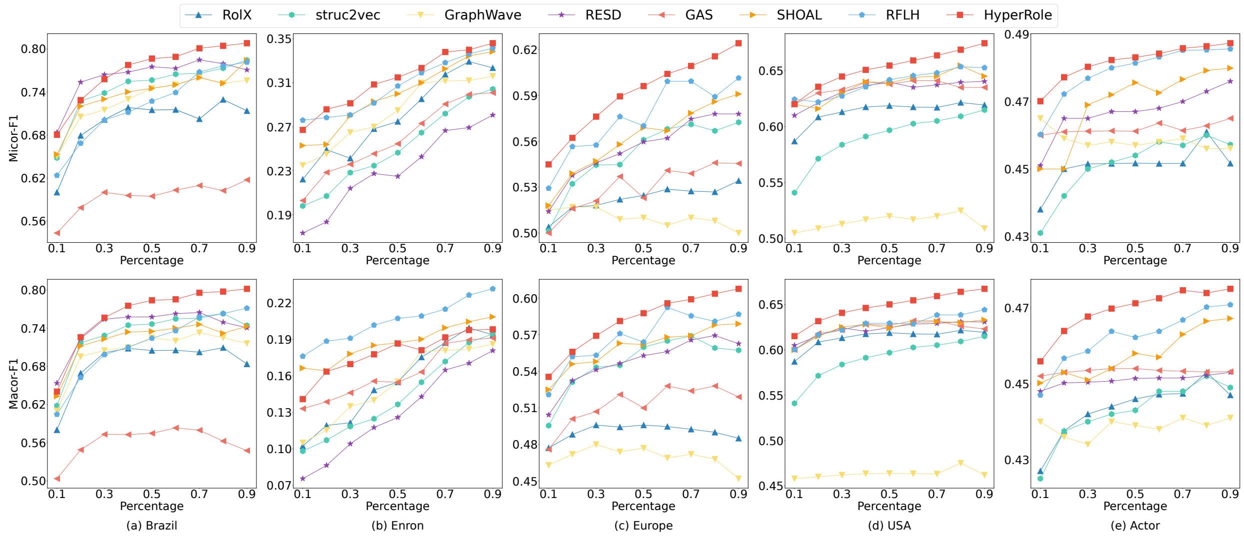


Fig. 5. Comparisons of results for all models under different division ratios for real-world datasets.

Algorithm 1 HyperRole

Input: Graph data $G = (V, E)$ extracted from the online social network

Output: Roles of predicted users $\hat{y}$

Extract local features X^l by ReFeX and higher-order features X^h by GDV
for $i = 1$ to M **do**
  Construct token sequences S_i^l and S_i^h by (8) and (9)
  Calculate $H_i^{l,(0)}, H_i^{h,(0)}$ by (10) and (11)
  for $\ell = 1$ to L **do**
    Calculate $H_i'^{l,(\ell)}, H_i'^{h,(\ell)}$ by (15)
    Calculate $H_i^{l,(\ell)}, H_i^{h,(\ell)}$ by (16)
  end for
  Calculate $h_{i,out}^l, h_{i,out}^h$ by (21)
end for
Calculate H_{re}^l, H_{re}^h by (22) and (23)
Calculate $\mathcal{L}_1$ by (24)
Use back-propagate on $\mathcal{L}_1$ to update parameters
Fix precession step model parameters
Input $H_{out}^l \oplus^c H_{out}^h$ into a multi-class classifier to get user roles $\hat{y}$
Calculate $\mathcal{L}_2$ by (25)
Use back-propagate on $\mathcal{L}_2$ to update parameters
Return: Predicted roles of users $\hat{y}$

IV. Experiments

A. Experimental Settings

1) Dataset: Five widely used real-world datasets with hierarchical structures are applied for evaluating HyperRole. The user categories in the datasets are strictly divided according to user roles. A brief description of the dataset used in the experiments is provided below, and detailed statistics of each dataset can be found in Table II.

- **Air-traffic networks** [11]: We adopt three air-traffic networks, namely the Brazil, Europe, and USA air-traffic networks, where users denote airports, and edges denote the existence of flights between airports. User categories are highly correlated with functions and roles played by airports in these networks, from which we can equate these air traffic networks to social networks, with airports viewed as users and edges representing the existence of connections between users.
- **Enron email network** [34]: This is a small email network within the Enron email system, where users correspond to employees' email addresses and edges between users indicate instances of email communication between employees. User labels are assigned based on employees' positions or departments.
- **Actor co-occurrence network** [35]: The Actor network is a subgraph network extracted from the Flim network [36]. In the Actor networks, users represent actors, connected edges indicate their simultaneous appearance in a Wikipedia page, and user categories are categorized according to the number of words on that page. Thus, the user category measures the influence of the user in the network.

TABLE II
The statistical information of real networks.

Dataset	#Users	#Edges	#Classes	Density(%)
Brazil	131	1,003	4	11.7792
Enron	143	2,583	7	25.4408
Europe	399	5,993	4	7.5478
USA	1,190	13,599	4	1.9222
Actor	7,779	26,752	4	0.0886

2) Evaluation metrics: We use two metrics together to measure the correctness of role discovery. Macro-F1 and Micro-F1 are extensions based on F1-score [37] and are widely used in multi-classification, which combine precision and recall to better handle class imbalance. Micro-F1 calculates the total precision and recall for all categories and then calculates the F1-score, while Macro-F1 calculates the F1-score after calculating the precision and recall for each class, and finally averages the F1-score for all categories. F1-score is as follows:

$$\text{F1-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{26}$$

3) Baselines: We compare HyperRole with the following baselines, which are divided into three types.

Matrix factorization based methods:

- RolX [8] gets user roles by non-negative matrix factorization of the structural feature matrix.
- GraphWave [9] analyzes user roles by spectral graph wavelet diffusion.

Random walk-based methods:

- struc2vec [11] constructs a multi-layer weighted graph based on the structural similarity of users to learn user embedding by random walks.

Graph learning based methods:

- GAS [16] introduces Graph Auto-Encoders (GAE) [38] to mine user roles from structural features
- RESD [17] utilizes Variational Graph Auto-Encoders (VGAE) [38] to learn the generation mechanism of structural features to improve the accuracy of role discovery.
- SHOAL [18] integrates local and higher-order features via Graph Isomorphism Network (GIN) and adversarial learning to enhance user embedding.
- RFLH [19] mines the connection between local and higher-order features by VGAE and Normalizing flows [39] to portray more essential user roles

4) Implementation details: For our model HyperRole, we follow the original article's guidance [32], [33] and adopt the same settings as other models with ReFeX and GDV methods for feature extraction. The number of iterations for ReFeX fusing neighboring features is set to 3, and the number of bins is set to 4. The subgraphs extracted by GDV are set to contain nodes ranging from 2 to 5, resulting in 73-dimensional higher-order features. For the Graph Transformer backbone, we set the Transformer layers to 2, the number of attention heads to 1, the number of MLP layers to 2, the hidden dimension to 128, and the hyperparameter $N \in \{3, 5, 7, 9, 11, 13, 15\}$. All curvatures are set as trainable parameters. During training, we set the Adam SGD optimizer to update the parameters and set the learning rate to 0.001. In addition, the batch size is set to 32 and the maximum training period is set to 100 epochs.

We develop the comprehensive model using the PyTorch framework and conduct all experiments in a Linux environment. To ensure the fairness of the entire comparison experiment, the hyperparameters of baselines are set as optimal according to the description within the above paper. All trials are performed on a single Nvidia GeForce RTX 3090 GPU. The software environment comprises Python version 3.6, PyTorch version 1.10.2, and CUDA version 11.3.

B. Result of Role Discovery

To evaluate the performance of HyperRole properly, we performed experiments with different training ratios on the five real datasets described in Section IV-A. We extract the graph-structured data from the dataset and randomly divide the data into a train set and a test set, with the ratio of the train set ranging from 10% to 90%, and input the graph-structured data into the model to output the corresponding role of users. The comparative results of role discovery are shown in Fig. 5.

Taking the Brazil network with the least number of users as an instance, our model HyperRole significantly outperforms the baseline when backed by a certain percentage of training data. The advantage of HyperRole over other baseline models becomes even more pronounced as the training set expands. This is attributed to the fact that HyperRole constructs input sequences based on structural feature similarity, thus effectively reducing the error messages received by the target users. In contrast, several other baseline models rely on GCNs to collect information from their neighbors, however, due to the possible role imbalance in the hierarchical structure of an online social network, which results in the target user receiving incorrect features from wrong categories. For example, users connected to a large number of leaf nodes can be misclassified as leaf nodes.

Compared to the Enron network, which is similar in size to the Brazil network, HyperRole still performs strongly in terms of Micro-F1 scores, but underperforms in terms of Macro-F1 scores. The reason for this difference is that Macro-F1 does not take into account the amount of data, but treats each category equally. In the Enron network, the limited number of users and the large number of categories lead to a sparse distribution of users in certain categories. HyperRole constructs input sequences by computing structural feature similarity may make different categories of users classified in the same input sequence to be learned incorrectly.

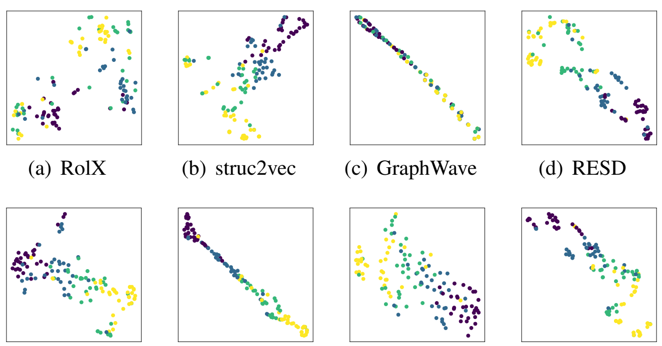
Our model also achieves the best results on all other networks. As the number of users in the network increases, the way HyperRole utilizes the Transformer to capture information from a global perspective shows a more pronounced advantage. It is worth noting that other graph learning models such as RESD and GAS show obvious limitations in large networks because they only learn user role information from local structural features. In contrast, SHOAL and RFLH incorporate both local and higher-order features into user embeddings and outperform RESD and GAS, which highlights the necessity of combining different structural features. Unlike SHOAL and RFLH, which add additional auxiliary modules to their graph learning models, HyperRole obtains excellent results by building a simple Transformer model on hyperbolic space, which effectively verifies that hyperbolic space can effectively capture the hierarchical structure in online social networks.

C. Visualization

In this subsection, we present the results of the visualization of the Brazil network. We first generate user embeddings from the model and then project them onto a two-dimensional space using T-SNE. Our expectation is that users with the same roles should be close to each other and users with different roles should be far away from each other in the 2D space.

TABLE III
TIME AND MEMORY COMPARISON OF DIFFERENT METHODS WITH DIFFERENT DATASETS.

Method	**Brazil**		**Enron**		**Europe**		**USA**		**Actor**	
	Time	#param	Time	#param	Time	#param	Time	#param	Time	#param
RolX [8]	0.3s	5K	0.4s	4K	0.4s	9K	0.7s	21K	3.1s	127K
GraphWave [9]	0.2s	0K	0.3s	0K	1.1s	0K	7.2s	0K	9.9m	0K
struc2vec [11]	5.5s	17K	6.9s	19K	21.7s	51K	2.4m	152K	13.8m	993K
RESD [17]	14.8m	66K	17.0m	55K	15.3m	66K	16.9m	63K	58.8m	76K
GAS [16]	2.2s	85K	2.6s	89K	3.6s	154K	6.5s	357K	13.2s	2038K
SHOAL [18]	3.6s	872K	3.6s	849K	3.5s	872K	3.5s	866K	5.0s	890K
RFLH [19]	24.3s	17K	24.1s	17K	26.6s	26K	57.9s	51K	8.9m	262K
HyperRole	**5.7s**	**33K**	**6.1s**	**36K**	**6.5s**	**102K**	**6.8s**	**304K**	**7.2s**	**1986K**



(a) RolX (b) struc2vec (c) GraphWave (d) RESD
(e) GAS (f) SHOAL (g) RFLH (h) HyperRole

Fig. 6. Visualization of user embeddings on Brazil network. Each subgraph shows the position of users mapped by the method in a two-dimensional space.

Fig. 6 depicts the spatial distribution of the various roles in the Brazil airport network. We note that although the RolX method presents a clear clustering phenomenon, it is not able to effectively differentiate between different user roles because it relies only on matrix factorization without additional constraints. Methods such as struc2vec, RESD and GAS show certain role differentiation abilities by utilizing local or higher-order features, but they are unable to sufficiently restrict users sharing the same role to a reasonable range. On the contrary, GraphWave, which relies on global features to distinguish users, is able to restrict users to similar spatial areas, but lacks the ability to distinguish between different roles. SHOAL and RFLH combine local and higher-order features, and show different abilities in role discovery. SHOAL focuses on restricting users with the same role to similar spatial areas, while a large number of users overlap complicates its differentiation. In contrast, RFLH can effectively distinguish users with the same role, although its restriction seems to be more relaxed. In comparison, our model HyperRole exhibits excellent differentiation. Except for users of incorrectly determined roles, the distinction between different roles is more obvious, and users of the same roles are close to each other and do not overlap. The experimental results demonstrate the effectiveness of embedding users in hyperbolic space.

D. Parameter Sensitivity Analysis

We investigate the impact of hyperparameters on the model's performance. To observe the trend change in scores more clearly, we carried out Min-Max Normalization on F1-scores and set the training ratio to 70%. As depicted in Fig. 7, our experiments are conducted on three distinct aeronautical networks, each representing a different network size.

We analyze the impact of the number of users N for sequence construction on the performance of extracting global features. We search for the optimal length of sequence construction and let $N \in \{3, 5, 7, 9, 11, 13, 15\}$. The Brazilian network achieves good results with a small N, obtaining the optimal result at $N = 9$. However, the score decreases rapidly as N continues to increase, due to the fact that an excessively long user sequence incorrectly classifies users with different roles in the same sequence. The experimental results show that each network has an optimal N such that each user can obtain as much feature information as possible from users with the same role while avoiding feature information from roles other than those expected.

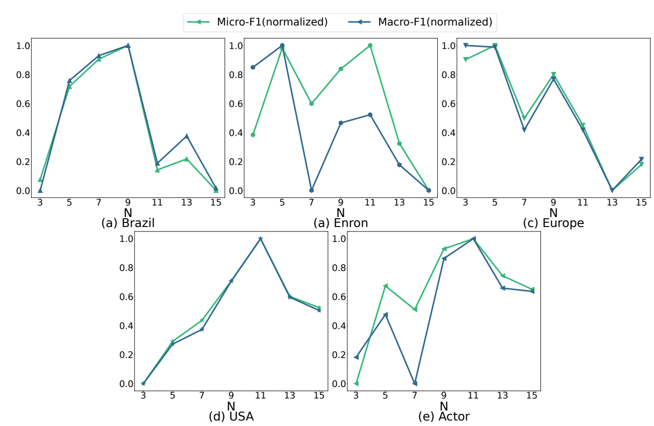


Fig. 7. The analysis of the length N of the node sequence.

E. Efficiency Comparisons

In this comparison experiment, we report the time and memory required by all methods on different datasets, where the number of parameters for GraphWave is 0K because it only involves mathematical computation and no parameter training. All methods still follow the experimental setup in Section IV-A with a 70% training set ratio. As shown in Table III, although our method HyperRole is not superior in terms of time and memory compared to non-graph learning methods, it far outperforms them in terms of correctness in the role discovery task. The time consumption of HyperRole is constant at a small order of magnitude compared to advanced graph learning methods. Due to the complex computation of the Transformer mechanism and hyperbolic geometry, HyperRole has a large overhead on large datasets, and otherwise, the memory required by HyperRole is within an acceptable range. It can be seen that HyperRole can obtain the best results among the current role discovery methods within a small time overhead.

V. Related Work

A. Role Discovery

Early role discovery methods applied some low-complexity matrix factorization methods to obtain information related to user roles from user features and network structure. RolX [8] utilizes the ReFeX method to extract user structural features and generates user embeddings using non-negative matrix factorization techniques. GraphWave [9] proposes a spectral wavelet diffusion method to generate role embeddings for each user without the requirement of heuristically defining user structural features and model training. EMBER [10], on the other hand, collects the degree information within a k-hop neighborhood of users and randomly selects users to compute similarity matrices and obtains user embedding matrices through implicit factorization.

Compared to the matrix factorization-based methods, random walk-based methods are more capable of recognizing the complex structure in which the user is located, and distinguishing the user roles by comparing the similarity of the walk sequences. struc2vec [11] constructs multi-layer weighted graphs to encode structural similarities between users and generates random walk sequences by measuring similarities between users at different scales. Node2bit [12] performs temporal random walks for directed temporal networks, and obtains the role features of each user by aggregating the user features in the random walk sequence. struc2gauss [13] learns user embeddings by learning them from Gaussian distributions, takes into account the noise present in complex networks, and computes user similarities for random sampling to constrain user role embeddings via the RoleSim [40] algorithm. Role2Vec [14] proposes a feature-based random walk method that categorizes topologically similar users into the same roles and assigns similar user role embeddings during training.

Graph learning-based methods focus more on mining potential connections between users from their structural features, and extracting richer implicit information about user roles from complex network structures through GNNs. GAS [16] to enhance the generalization ability of user embedding, encoding the structural features of the network by GCN [41], and then reconstructing the structural features to learn user roles by the multi-layer perceptron. To model the nonlinear relationship of the user's structural features, RESD [17] introduces VAE and compensates for the network information lost during the encoding process by adding node degree constraints. SHOAL [18] integrates local and higher-order structural features of the user by combining GAE and adversarial learning to more accurately recognize user roles. RFLH [19] exploits VGAE and Normalizing flows to mine the connections between different structural features of a user, generating more flexible and expressive user role embeddings.

Even though existing approaches to graph learning are extremely capable of learning structural features and mining complex network structures, they all ignore the widespread hierarchical structures in online social networks. Hierarchical structure is crucial to help us understand and analyze users' roles, so our model notes this problem and addresses it by introducing hyperbolic geometry.

B. Hyperbolic Graph Neural Networks

The hyperbolic graph neural network is born from the combination of hyperbolic geometry and graph neural networks, [28] first defines the graph neural network in hyperbolic geometric models (Lorentz model and Poincaré Ball model). HGCN [29] builds on its previous work by defining complete convolutional operations and nonlinear activations, and confirms that generated node embeddings preserve the hierarchical structure of the network. Since computing in hyperbolic space is different from Euclidean space, HGAT [30] develops a complete attention computing mechanism for hyperbolic space to migrate graph attention networks to hyperbolic space.

On the basis of HGAT, although there have been some studies [42] to investigate how to extend the attention mechanism to the whole globe in hyperbolic space, they only apply the hyperbolic geometry to a certain part of the process, not the whole process. It is obvious that the advantages brought by the local application of hyperbolic geometry are undermined by subsequent operations in Euclidean space, so we propose a simplified Hyperbolic Transformer that applies hyperbolic geometry to the whole model.

VI. Conclusion

In this paper, we propose a graph deep learning framework based on hyperbolic geometry and Transformer, called HyperRole, for role discovery of users in online social networks. HyperRole first extracts users' real statistical features in the network as structural features, and then computes the similarity of the structural features to construct two independent user sequences to mine the local and higher-order relationships between users, respectively. Then we build a simplified hyperbolic Transformer model to learn the user's role features from structural features to generate user embeddings that can correctly express the hierarchical structure of users in the network. Finally, we utilize the hyperbolic distance between user embeddings to guide the target user to obtain its own role features from other users in the same user sequence. Experimental results show that our HyperRole can effectively preserve the hierarchical structure in online social networks, which is one of the main reasons why HyperRole achieves optimal results in the role discovery task compared to other baselines. Moreover, compared with other complex graph deep learning models, the computational space overhead of HyperRole is much smaller, while the computational time overhead does not increase significantly.

References

[1] W. Lin, Z. Gao, and B. Li, "Guardian: Evaluating trust in online social networks with graph convolutional networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 914–923.

[2] W. Lin and B. Li, "Medley: Predicting social trust in time-varying online social networks," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[3] J. Wen, N. Jiang, J. Li, X. Liu, H. Chen, Y. Ren, Z. Yuan, and Z. Tu, "Dtrust: Toward dynamic trust levels assessment in time-varying online social networks," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[4] B. Mitra, S. Sural, J. Vaidya, and V. Atluri, "A survey of role mining," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, pp. 1–37, 2016.

[5] P. Jiao, Q. Tian, W. Zhang, X. Guo, D. Jin, and H. Wu, "Role discovery-guided network embedding based on autoencoder and attention mechanism," *IEEE Transactions on Cybernetics*, vol. 53, no. 1, pp. 365–378, 2021.

[6] P. Jiao, X. Guo, T. Pan, W. Zhang, Y. Pei, and L. Pan, "A survey on role-oriented network embedding," *IEEE Transactions on Big Data*, vol. 8, no. 4, pp. 933–952, 2021.

[7] R. A. Rossi and N. K. Ahmed, "Role discovery in networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1112–1131, 2014.

[8] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li, "Rolx: structural role extraction & mining in large graphs," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1231–1239.

[9] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1320–1329.

[10] D. Jin, M. Heimann, T. Safavi, M. Wang, W. Lee, L. Snider, and D. Koutra, "Smart roles: Inferring professional roles in email networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2923–2933.

[11] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 385–394.

[12] D. Jin, M. Heimann, R. A. Rossi, and D. Koutra, "Node2bits: Compact time-and attribute-aware node representations for user stitching," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 483–506.

[13] Y. Pei, X. Du, J. Zhang, G. Fletcher, and M. Pechenizkiy, "struc2gauss: Structural role preserving network embedding via gaussian embedding," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1072–1103, 2020.

[14] N. K. Ahmed, R. A. Rossi, J. B. Lee, T. L. Willke, R. Zhou, X. Kong, and H. Eldardiry, "Role-based graph embeddings," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2401–2415, 2020.

[15] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2020.

[16] X. Guo, W. Zhang, W. Wang, Y. Yu, Y. Wang, and P. Jiao, "Role-oriented graph auto-encoder guided by structural information," in *Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part II 25*. Springer, 2020, pp. 466–481.

[17] W. Zhang, X. Guo, W. Wang, Q. Tian, L. Pan, and P. Jiao, "Role-based network embedding via structural features reconstruction with degree-regularized constraint," *Knowledge-Based Systems*, vol. 218, p. 106872, 2021.

[18] W. Zhang, Y. Yu, T. Pan, L. Pan, P. Jiao, and W. Wang, "Generating structural node representations via higher-order features and adversarial learning," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1487–1492.

[19] M. Du, P. Jiao, H. Tang, W. Zhang, and J. Wu, "Role-oriented representation learning via fusioning local and higher-order feature," *Knowledge-Based Systems*, vol. 282, p. 111115, 2023.

[20] X. Guo, Q. Tian, W. Zhang, W. Wang, and P. Jiao, "Learning stochastic equivalence based on discrete ricci curvature." in *IJCAI*, 2021, pp. 1456–1462.

[21] Q. Tian, W. Zhang, P. Jiao, K. Zhong, N. Wu, and L. Pan, "Integrating higher-order features for structural role discovery," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2022, pp. 244–258.

[22] P. Jiao, K. Yu, Q. Bao, Y. Jiang, X. Guo, and Z. Zhao, "Graph contrastive learning with node-level accurate difference," *Fundamental Research*, 2024.

[23] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[24] F. Chen and K. Li, "Detecting hierarchical structure of community members in social networks," *Knowledge-based systems*, vol. 87, pp. 3–15, 2015.

[25] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," in *International Conference on Learning Representations*, 2020.

[26] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná, "Hyperbolic geometry of complex networks," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 82, no. 3, p. 036106, 2010.

[27] F. Papadopoulos, M. Kitsak, M. Á. Serrano, M. Boguñá, and D. Krioukov, "Popularity versus similarity in growing networks," *Nature*, vol. 489, no. 7417, pp. 537–540, 2012.

[28] Q. Liu, M. Nickel, and D. Kiela, "Hyperbolic graph neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[29] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[30] Y. Zhang, X. Wang, C. Shi, X. Jiang, and Y. Ye, "Hyperbolic graph attention network," *IEEE Transactions on Big Data*, vol. 8, no. 6, pp. 1690–1701, 2021.

[31] E. Min, R. Chen, Y. Bian, T. Xu, K. Zhao, W. Huang, P. Zhao, J. Huang, S. Ananiadou, and Y. Rong, "Transformer for graphs: An overview from architecture perspective," *arXiv preprint arXiv:2202.08455*, 2022.

[32] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos, "It's who you know: graph mining using recursive structural features," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 663–671.

[33] T. Hočevar and J. Demšar, "A combinatorial approach to graphlet counting," *Bioinformatics*, vol. 30, no. 4, pp. 559–565, 2014.

[34] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European conference on machine learning*. Springer, 2004, pp. 217–226.

[35] X. Ma, G. Qin, Z. Qiu, M. Zheng, and Z. Wang, "Riwalk: Fast structural node embedding via role identification," in *2019 IEEE international conference on data mining (ICDM)*. IEEE, 2019, pp. 478–487.

[36] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 807–816.

[37] N. Chinchor, "Muc-4 evaluation metrics," in *Proceedings of the 4th Conference on Message Understanding*, 1992, p. 22–29.

[38] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[39] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.

[40] R. Jin, V. E. Lee, and L. Li, "Scalable and axiomatic ranking of network role similarity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, pp. 1–37, 2014.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[42] S. Cho, S. Cho, S. Park, H. Lee, H. Lee, and M. Lee, "Curve your attention: Mixed-curvature transformers for graph representation learning," *arXiv preprint arXiv:2309.04082*, 2023.