Hyperbolic Graph Learning for Social Recommendation

Yonghui Yang, Le Wu* *Member, IEEE*, Kun Zhang, Richang Hong, *Senior Member, IEEE*, Hailin Zhou, Zhiqiang Zhang, Jun Zhou, Meng Wang, *Fellow, IEEE*

Abstract—Social recommendation provides an auxiliary social network structure to enhance recommendation performances. By formulating user-user social network and user-item interaction graph, modern social recommendation architecture is built on learning user and item embeddings into Euclidean space with graph convolution operations. However, the Euclidean space suffers structure distortion when representing the nature power-law distribution of graphs, leading to sub-optimal results for graph based social recommendation. Recently, some studies have explored the alternative of graph embedding learning into hyperbolic space, which can preserve the hierarchy of real-world graphs. However, directly applying current hyperbolic graph embedding models for social recommendation is non-trivial as two challenges: network heterogeneity and social diffusion noise. First, due to the semantic gap existing between social networks and user-item interactions, how to tackle the heterogeneity issue of social recommendation under hyperbolic formulation? Second, explicit modeling of social diffusion easily introduces noise for user preference learning, especially for those active users with amounts of interactions. To tackle the above challenges, in this paper, we propose a *Hyperbolic Graph Learning based Social Recommendation (HGSR)* model. Firstly, we exploit social structure with hyperbolic social embedding pre-training, which could preserve the hierarchical properties of social networks. Secondly, we construct the heterogeneous graph based on user-item interactions and social networks, then treat the pre-trained social embeddings as an additional feature input for user preference learning. Such that, we combine explicit heterogeneous graph learning and implicit feature enhancement for the hyperbolic social recommendation, which can well tackle heterogeneity and social noise issues. We conduct empirical studies on four datasets, and extensive experiments demonstrate the effectiveness of our proposed model compared to state-of-the-art baselines. We release the source code at: https://github.com/yimutianyang/HGSR.

1 Introduction

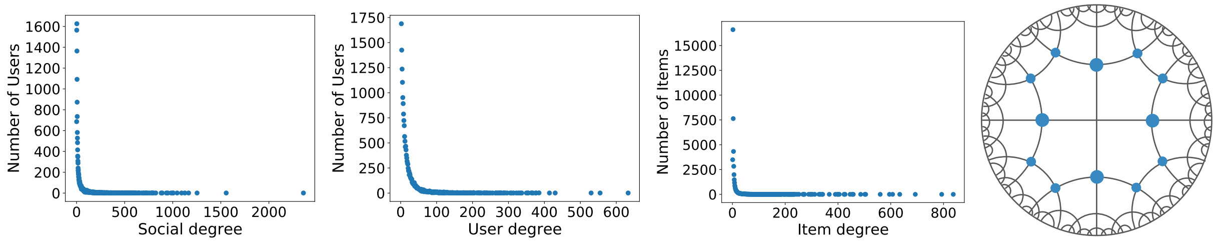
Recommender systems provide personalized suggestions by modeling users' preferences. As one of the basic paradigms, Collaborative Filtering (CF) has been widely deployed in recommendation systems, which learn users' unknown preferences based on user-item historical interactions [36], [14], [34]. Despite the wide applicability, the performance of CF is still far from satisfactory due to the limited user interaction data. With the ubiquitous social networks, social recommendation has emerged as an important research technique for personalized services. Social recommendation utilizes the additional user-user social networks to alleviate interaction data sparsity and improve recommendation performances [40], [10]. The underlying rationale is that users are influenced by their corresponding ego-centric social network, such that socially connected users have similar preferences [1].

Following the theory of social homogeneity and social influence [1], [2], [58], early works usually focus on first-order social connections, such as designing social regularization [27] or modeling social neighbor influence [10]. Recently, inspired by the great representation ability of graph neural networks, graph-based recommendation methods have achieved state-of-the-art performances [44], [12], [47]. Graph-based CF methods iteratively update user and item representations by propagating the collaborative signals with graph convolutions [44], [6], [12], [53]. Extending CF models, graph-based social recommendations enhance representations from both the social diffusion and interest propagation [7], [47], [46], [57]. E.g., DiffNet++ proposes to learn user representations by attentively aggregating neighbors from both social network and user-item interaction graph [46].

Despite the performance improvement, we argue that current graph-based social recommendation models are still far from satisfactory. The reason is that all the above graph-based recommendation models embed nodes into Euclidean space with Graph Convolutional Networks(GCNs) [12], [46], while neglecting the geometry properties hidden in graphs. As well recognized by sociologists, real-world graphs exhibit the power-law distribution [31]. For example, we illustrate the node degree distribution of a real-world social recommendation dataset Epinions in Fig.1(a-c), and the detailed statistics of Epinions are described in Table 4. As shown in this Fig, the degree distribution of nodes from the social graph and that of the user-item bipartite graph all show the power-law distribution. The power-law distribution is a consequence of tree-like hierarchical organization, showing that a small group of nodes organized in a hierarchical manner into increasingly large groups [32], [29]. By modeling the graph structure into Euclidean space, the

**Corresponding author*

- *Y. Yang, L. Wu, K. Zhang, R. Hong, M. Wang are with the School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China.*
  Emails: {yyh.hfut, lewu.ustc, kunzhang1028, hongrc.hfut, eric.mengwang}@gmail.com.
- *H. Zhou is with IVY MOBIL.*
  Email: harry@ivymobile.com.
- *Z. Zhang, J. Zhou are with Ant Group, Hangzhou, China.*
  Email: {lingyao.zzq, zhoujun}@antfin.com.



(a) Social network (b) User-item graph (c) User-item graph (d) Hyperbolic space

Fig. 1. (a-c): Illustration of the degree distribution of user-user social network, and user-item interaction graph on Epinions dataset. (d): Two-dimensional hyperbolic space visualization.

volume inside a Euclidean ball with a radius r only grows quadratically (i.e., the area of the sphere is πr^2), which leads to high structure distortion for embedding representation.

In contrast to Euclidean space, hyperbolic geometry has an area that is exponential with radius r, which provides a nice alternative to model underlying hierarchical structure data [29], [9], [19]. As illustrated in Fig.1(d), each node is represented with a blue dot in the two-dimensional hyperbolic space. As the radius r increases, the capacity of the nodes increases exponentially. Then, the tree-structured hierarchical property of node distributions can be well embedded in the hyperbolic space. Recently, researchers propose to model graph-based recommendations with hyperbolic geometry [37], [52], [51], [21], [41]. These models combine the complementary advantage of the expressiveness of GCNs and the hyperbolic geometry of node representation, showing better performance than purely graph embedding learning in the Euclidean space. In this paper, we study the problem of hyperbolic graph learning for social recommendation, which is non-trivial due to the following two challenges. First, due to the semantic gap existing between social networks and user-item interactions, how to tackle the heterogeneity issue of social recommendation under hyperbolic formulation? Besides, explicit modeling of social influence with graph convolution easily introduces noise for preference learning, especially for those active users with amounts of interactions.

In this paper, we propose a *Hyperbolic Graph Learning based Social Recommendation (HGSR)* model to tackle heterogeneity and social noise for the hyperbolic social recommendation. Technically, *HGSR* consists of two main stages: hyperbolic social pre-training and hyperbolic preference learning. Specifically, we first exploit social structure properties through a hyperbolic social pre-training module, which is optimized to reconstruct social networks. The hyperbolic social pre-training is designed to preserve the social hierarchical properties. Secondly, we design a social pre-training enhanced hyperbolic heterogeneous graph learning module, that formulates users' social network and user-item interactions as a heterogeneous graph, then treats the pre-trained social embeddings as an additional feature input for graph learning. Such that, we combine explicit heterogeneous graph learning and implicit feature enhancement to tackle the heterogeneity and social noise issues in hyperbolic social recommendation. We conduct experiments on four public datasets, extensive experimental results show that our proposed *HGSR* can significantly improve recommendation performances. Our main contributions can be summarized as follows:

- We formulate the social recommendation task under hyperbolic space learning, and propose a novel *Hyperbolic Graph Learning based Social Recommendation (HGSR)* model.
- We design a hyperbolic social pre-training module to preserve the social structure as features, and tackle the social recommendation from both explicit heterogeneous graph learning and implicit feature enhancement.
- Extensive experimental results on four real-world datasets clearly demonstrate the effectiveness of the proposed *HGSR* model, including high performance, generalization of the pre-trained feature, and applicability to various sparsity users.

2 Preliminaries

2.1 Hyperbolic Social Recommendation

Problem Statement. In a social recommendation platform, there are two kinds of entities: a user set U ($|U| = M$) and an item set V ($|V| = N$). Two kinds of behaviors are available in this scenario: user-item interactions and user-user social connections. Considering the most common recommendation scenarios are implicit feedback (such as click, like and purchase), we use $\mathbf{R} \in \mathbb{R}^{M\times N}$ to denote user-item interaction matrix, where $r_{ai} = 1$ if user a interacts with item i, otherwise it equals 0. Besides, user-user social network is denoted by $\mathbf{S} \in \mathbb{R}^{M\times M}$, where $s_{ba} = 1$ if user a follows user b, otherwise it equals 0. Given the user-item interaction matrix $\mathbf{R} \in \mathbb{R}^{M\times N}$ and user-user social network $\mathbf{S} \in \mathbb{R}^{M\times M}$. The goal of hyperbolic graph based social recommendation is to predict users' unknown preferences: $\hat{\mathbf{R}} = f(\mathbf{R}, \mathbf{S})$, where the function $f(\cdot)$ learned in hyperbolic space. The main notations are summarized in Table 1.

Schema Illustration. As illustrated in Fig. 2, we first describe the schema of the proposed hyperbolic social recommendation from implicit and explicit modeling. The left is the implicit modeling process, given user-user social matrix $\mathbf{S}$ as input, we first use hyperbolic GNNs to pretrain social embeddings: $\mathbf{P} = \mathrm{HPre}(S)$, where $\mathrm{HPre}(\cdot)$ denotes the hyperbolic pre-training function. Then, the pre-trained social embeddings can be viewed as additional features to enhance the recommender. The right is the explicit modeling process, given user-user social matrix $\mathbf{S}$ and user-item interaction

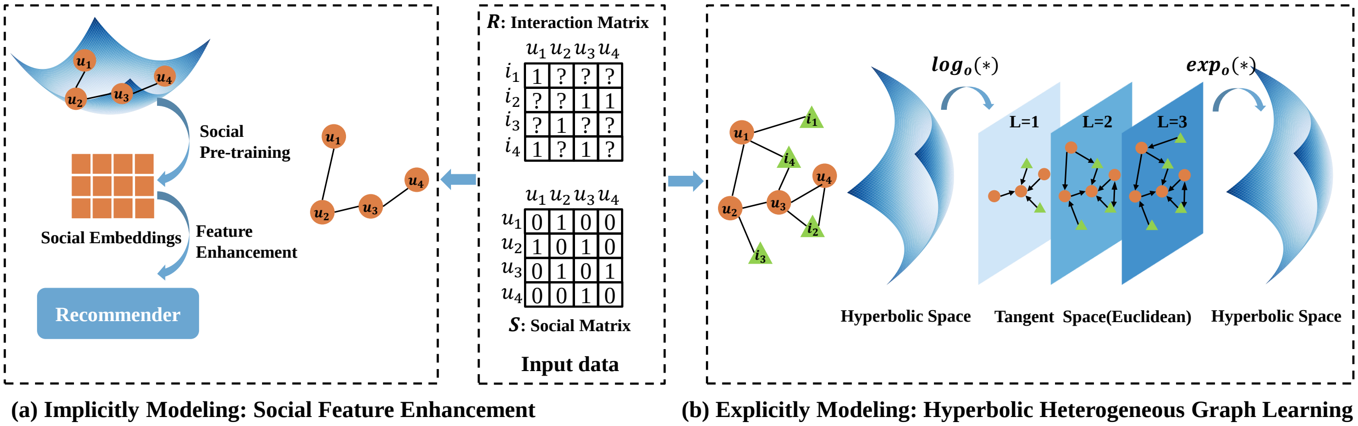


Fig. 2. Flow chart of hyperbolic graph social recommendation with implicit and explicit modeling. (a) Implicit modeling: we design the hyperbolic social pertaining module to extract social embeddings, then feed social embeddings as an additional feature to enhance recommender; (b) Explicit modeling: we construct the heterogeneous graph according to social network and interaction matrix, then perform social influence diffusion and interest propagation with hyperbolic graph learning.

TABLE 1
Mathematical Notations.

Notation	Description
U	Userset, $\|U\| = M$
V	Itemset, $\|V\| = N$
$\mathbf{R} \in \mathbb{R}^{M \times N}$	Interaction matrix
$\mathbf{S} \in \mathbb{R}^{M \times M}$	Social matrix
$\mathcal{G} = \{U \cup V, \mathbf{S}, \mathbf{R}\}$	Heterogeneous graph
$\mathbb{H}_k^d$	Hyperbolic space (Riemannian manifold with a curve k and a dimension d)
$\mathcal{T}_o\mathbb{H}_k^d$	Tangent space with an original point $\mathbf{o}$
$\mathbf{Z}^0$	Initialized social embedding matrix in tangent space
$\mathbf{P}$	Pre-trained social embedding matrix in hyperbolic space
$\mathbf{H^0}$	Fused user embedding matrix in tangent space
$\mathbf{Q}^0$	Initialized item embedding matrix in tangent space
$\mathbf{U}, \mathbf{V}$	Final user and item embedding matrices in hyperbolic space

matrix $\mathbf{R}$, we formulate these two kinds of behavior data as a heterogeneous graph $\mathcal{G} = \{U \cup V, \mathbf{S}, \mathbf{R}\}$. Borrowing the strength of capturing the hierarchical structure of hyperbolic learning [37], [4], we model the interest propagation and social influence diffusion process by hyperbolic heterogeneous graph learning. In this work, we argue that single implicit or explicit modeling is inefficient for social recommendation. Firstly, implicit modeling extracts general social features to preserve the social structure, while failing to capture the hidden recommendation patterns like user-user-item. Secondly, explicit graph diffusion modeling assumes that each social neighbor contributes to the user's interaction behavior. Although social information can supplement the sparse interactions, social diffusion usually disturbs active users' preference learning [47], which can also be verified in our experiments (Section 4.3). To this end, we combine implicit and explicit modelings, and propose *HGSR* for social recommendation. Next, we introduce the basic hyperbolic formulation in this paper.

2.2 Lorentz Formulation

Due to the high efficiency and stability, we select the Lorentz formulation to learn hyperbolic embeddings [30]. Here, we give a brief introduction to the correlated definitions and properties of the used Lorentz formulation.

Hyperbolic Manifold and (Euclidean) Tangent Space. Hyperbolic space is defined as a Riemannian manifold $\mathbb{H}_k^d$, where d is the space dimension and k is the curvature parameter (curvature $c = -1/k$):

$$\mathbb{H}_k^d = \{\mathbf{x} \in \mathbb{R}^{d+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -k, \mathbf{x}_0 > 0\}, \tag{1}$$

where $\langle , \rangle_{\mathcal{L}}$ denotes Lorentz inner product, which is defined as:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^{d} x_i y_i. \tag{2}$$

Furthermore, given an original point $\mathbf{o} \in \mathbb{H}_k^d$, we can define the corresponding tangent space (Euclidean space) $\mathcal{T}_o\mathbb{H}_k^d$ as the first-order approximation of $\mathbb{H}_k^d$ around point $\mathbf{x}$:

$$\mathcal{T}_o\mathbb{H}_k^d = \{\mathbf{v} \in \mathbb{R}^{d+1} : \langle \mathbf{v}, \mathbf{o} \rangle_{\mathcal{L}} = 0\}. \tag{3}$$

Hyperbolic Distance and Mapping Function. Given any point pair $\mathbf{x}, \mathbf{y}$ in hyperbolic space $\mathbb{H}_k^d$, the distance is computed as follows:

$$d_{\mathcal{L}}(\mathbf{x}, \mathbf{y}) = \sqrt{k} arcosh(-\frac{\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}}{k}). \tag{4}$$

After defining hyperbolic space $\mathbb{H}_k^d$ and tangent space $\mathcal{T}_o\mathbb{H}_k^d$, we next introduce the mapping functions between these two spaces. Specifically, the exponential and the logarithmic map function to map points between tangent space and hyperbolic space, which is defined as follows:

$$exp_x(\mathbf{v}) = cosh(\frac{||v||_{\mathcal{L}}}{\sqrt{k}})\mathbf{x} + \sqrt{k} sinh(\frac{||\mathbf{v}||_{\mathcal{L}}}{\sqrt{k}})\frac{\mathbf{v}}{||\mathbf{v}||_{\mathcal{L}}}, \tag{5}$$

where $||\mathbf{v}||_{\mathcal{L}} = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle_{\mathcal{L}}}$ is the Lorentz normalization of $\mathbf{v}$. The $exp_x(*)$ operation maps point from tangent space to hyperbolic space. Correspondingly, the $log_x(*)$ operation maps point from hyperbolic space to tangent space:

$$log_x(\mathbf{y}) = \sqrt{k} arcosh(-\frac{\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}}{k})\frac{\mathbf{y} + \frac{1}{k}\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}\mathbf{x}}{||\mathbf{y} + \frac{1}{k}\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}\mathbf{x}||}. \tag{6}$$

In this paper, we refer existing hyperbolic recommendation works [37], [52], [51], set a fixed curvature to -1 ($k = 1$), and select $\mathbf{o} = [-1, 0, 0, ..., 0]$ as the original point for inference.

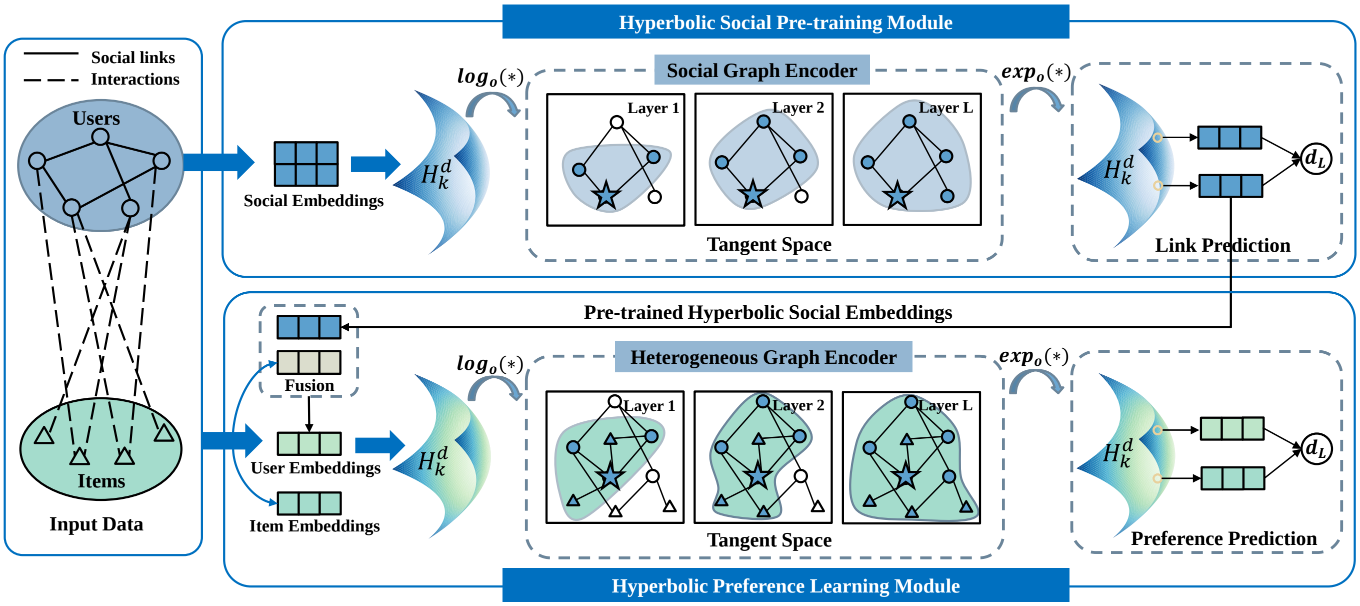


Fig. 3. The overall framework of our proposed *HGSR* model, which consists of two modules. The upper part is the hyperbolic social pre-training module and the bottom part is the hyperbolic preference learning module.

3 METHODOLOGY

In this section, we introduce our proposed *Hyperbolic Graph Learning based Social Recommendation (HGSR)* method. We first present the overall architecture, followed by the specification of each module. After that, we present the objective function for model optimization. Finally, we discuss our model from space and time complexity, respectively.

3.1 Overall Architecture

As illustrated in Fig. 3, our model consists of two modules: hyperbolic social pre-training module and hyperbolic preference learning module. Among them, social pre-training module aims to extract general social features, that fully preserve social structure in hyperbolic space. After that, the hyperbolic preference learning module further combines the pre-trained social embeddings and heterogeneous graph structure to learn better user and item representations for recommendation. Combining implicit social feature enhancement and explicit heterogeneous graph learning, *HGSR* can make full use of social networks to enhance recommendation performances.

3.2 Hyperbolic Social Pre-training Module

To fully exploit social network with hierarchical properties, we design a hyperbolic social pre-training module to extract social embeddings. As illustrated in the upper part of Fig. 3, there are three components of hyperbolic social pre-training module: social embedding initialization, hyperbolic social encoder, and social link optimization.

3.2.1 Social Embedding Initialization

We first initialize user embeddings in hyperbolic space with a hyperbolic Gaussian sampling method [37], [51]. Let $\mathbf{P}^E \in \mathbb{R}^{M\times d}$ denote user embeddings in Euclidean space. Given the pre-defined original point $\mathbf{o} = [-1, 0, 0, ..., 0]$, we have the corresponding user embeddings $\mathbf{Z}^0 = [0, \mathbf{P}^E]$ in tangent space. Then, the initialized hyperbolic social embeddings $\mathbf{P}^0$ are defined as follows:

$$\mathbf{P}^0 = exp_{\mathbf{o}}(\mathbf{Z}^0). \tag{7}$$

3.2.2 Hyperbolic Social Graph Encoder

After initializing user embeddings in social networks, the hyperbolic social encoder is designed to model the high-order social influence diffusion process for user embedding learning. There are two steps in this encoder: social diffusion and embedding readout, we first introduce the social diffusion process. As the mean aggregation does not have closed form solution in hyperbolic space [8], [5], we need firstly map hyperbolic embeddings into tangent space, then perform social propagation on tangent space. Specifically, for user a, given her embeddings $\mathbf{z}_a^l$ in l^{th} convolution layer, the corresponding embeddings $\mathbf{z}_a^{l+1}$ in $(l+1)^{th}$ convolution layer is updated by:

$$\mathbf{z}_a^{l+1} = \mathbf{z}_a^l + \sum_{b\in\mathbf{S}_a} \frac{1}{|\mathbf{S}_a|}\mathbf{z}_b^l, \tag{8}$$

where $\mathbf{S}_a$ denotes social neighbors who a follows, and $|\mathbf{S}_a|$ denotes the number of social neighbors. After L social graph convolution layers, we obtain $L+1$ user embedding matrices $[\mathbf{Z}^0, \mathbf{Z}^1, ..., \mathbf{Z}^L]$, then use sum-pooling strategy to fuse the high-order social information:

$$\mathbf{Z} = \sum_{l=1}^{L} \mathbf{Z}^l. \tag{9}$$

Following the existing hyperbolic graph recommendation works [37], [51], we discard 0^{th} user embeddings, which means that user representations only rely on the corresponding one-skip neighbors and high-order neighbors. Then, we project the learned user embeddings from tangent space back to hyperbolic space:

$$\mathbf{P} = exp_{\mathbf{o}}(\mathbf{Z}), \tag{10}$$

where $\mathbf{P}$ denotes the final hyperbolic social embeddings, which will be used for social link prediction.

3.2.3 Social Link Prediction

After obtaining hyperbolic social embeddings $\mathbf{P}$, as shown the upper part in Fig. 3, we apply hyperbolic distance function $d_{\mathcal{L}}$ to infer the propensity score $\hat{s}_{ab}$ that user a links with user b:

$$\hat{s}_{ab} = \frac{1}{d^2_{\mathcal{L}}(\mathbf{p}_a, \mathbf{p}_b)}. \tag{11}$$

We employ the adaptive margin loss function for optimization [51], which is described as follows:

$$\mathcal{L}_s = \sum_{u=0}^{M-1} \sum_{(a,b)\in D_u^s} max(d^2_{\mathcal{L}}(\mathbf{p}_u, \mathbf{p}_a) - d^2_{\mathcal{L}}(\mathbf{p}_u, \mathbf{p}_b) + m_{ua}, 0), \tag{12}$$

where $D_u^S = \{(a,b)|a \in \mathbf{S}_u \wedge b \notin \mathbf{S}_u\}$ denotes the pair-wise training data, and m_{ua} is an adaptive margin depending on positive sample (u, a):

$$m_{ua} = \delta(d^2_{\mathcal{L}}(\mathbf{p}_u, \mathbf{o}) + d^2_{\mathcal{L}}(\mathbf{p}_a, \mathbf{o}) - d^2_{\mathcal{L}}(\mathbf{p}_u, \mathbf{p}_a)), \tag{13}$$

where $\delta(\cdot)$ is the sigmoid function. We use Riemannian SGD [3] to optimize the above social reconstruction loss and obtain the optimal social embeddings. Next, the pre-trained social embeddings as additional features, are fed into the hyperbolic preference learning module in an implicit feature enhancement manner.

3.3 Hyperbolic Preference Learning Module

With tacking the semantic gap existing between social networks and user-item interaction graph, we propose the hyperbolic preference learning module based on heterogeneous graph learning. As illustrated in the bottom part of Fig. 3, we formulate user-item interactions and social network as a heterogeneous graph, and treat the pre-trained social embeddings as feature input to enhance recommendation performances. As such, we combine implicit feature enhancement and explicit graph modeling for preference learning, which can better tackle heterogeneity and diffusion noise in social recommendation. Following, we introduce three components of this module: hyperbolic embedding fusion, hyperbolic heterogeneous graph learning, and preference prediction.

3.3.1 Hyperbolic Embedding Fusion

Same to the hyperbolic social pre-training module, we firstly initialize user and item preference embeddings $\mathbf{U}^E \in \mathbb{R}^{M\times d}, \mathbf{V}^E \in \mathbb{R}^{N\times d}$ in Euclidean space. Then, we project the pre-trained hyperbolic social embeddings into tangent space: $\mathbf{Z} = log_{\mathbf{o}}(\mathbf{P})$. The embedding fusion process is performed on tangent space:

$$\mathbf{H}^0 = g(\mathbf{Z}, [0, \mathbf{U}^E]), \tag{14}$$

where $\mathbf{H}^0$ denotes the fused user embeddings in tangent space, and $g(\cdot)$ denotes fusion function. We try several fusion strategies such as MLP, concatenation, pooling, and find sum-pooling is the most effective. For items, we have their tangent embeddings $\mathbf{Q}^0 = [0, \mathbf{V}^E]$. Based on the fused user embeddings and item embeddings in tangent space, we have the initialized hyperbolic user and item embeddings $\mathbf{U}^0 = exp_{\mathbf{o}}(\mathbf{H}^0)$, $\mathbf{V}^0 = exp_{\mathbf{o}}(\mathbf{Q}^0)$.

3.3.2 Hyperbolic Heterogeneous Graph Encoder

Considering that user preferences are influenced by both social neighbors and interacted items, we refer to DiffNet++ [46] and encode the high-order social influence diffusion and user-item propagation to preference learning. We first project the initialized hyperbolic user and item embeddings into tangent space $\mathbf{H}^0$ and $\mathbf{Q}^0$, then perform neighbor propagation on the heterogeneous graph. Specifically, for user a and item i, we update their embeddings $\mathbf{h}_a^{l+1}, \mathbf{q}_i^{l+1}$ on $(l+1)^{th}$ convolution layer as follows:

$$\begin{aligned} \mathbf{h}_a^{l+1} &= \mathbf{h}_a^l + \alpha \sum_{b\in\mathbf{S}_a} \frac{1}{|\mathbf{S}_a|}\mathbf{h}_b^l + (1-\alpha) \sum_{j\in\mathbf{R}_a} \frac{1}{|\mathbf{R}_a|}\mathbf{q}_j^l, \\ \mathbf{q}_i^{l+1} &= \mathbf{h}_i^l + \sum_{c\in\mathbf{R}_i^T} \frac{1}{|\mathbf{R}_i^T|}\mathbf{h}_c^l, \end{aligned} \tag{15}$$

where $\mathbf{h}_a^l$ and $\mathbf{q}_i^l$ mean user a and item i embeddings on l^{th} convolution layer, respectively. $\mathbf{S}_a$ and $\mathbf{R}_a$ denote user a's linked social neighbors and interacted items. $\mathbf{R}_i^T$ denotes the sub userset that interact with item i. Besides, we set a hyper-parameter α to balance social and interest weights when performing neighbor aggregation for user embedding learning, the parameter sensitivity was also conducted on experimental parts. In practice, we find that a simple weighted sum strategy has a better performance compared with the attention mechanism. After L convolution layers, we obtain $L+1$ user embedding matrices $[\mathbf{H}^0, \mathbf{H}^1, ..., \mathbf{H}^L]$ and $L+1$ item embedding matrices $[\mathbf{Q}^0, \mathbf{Q}^1, ..., \mathbf{Q}^L]$, we use sum-pooling to combine these embeddings:

$$\mathbf{H} = \sum_{l=1}^{L} \mathbf{H}^l, \mathbf{Q} = \sum_{l=1}^{L} \mathbf{Q}^l. \tag{16}$$

Then, we project the fused tangent embeddings back to hyperbolic space to generate the final hyperbolic preference embeddings:

$$\mathbf{U} = exp_{\mathbf{o}}(\mathbf{H}), \mathbf{V} = exp_{\mathbf{o}}(\mathbf{Q}). \tag{17}$$

3.3.3 Preference Prediction

After obtaining the learned hyperbolic user and item preference embeddings, we predict the preference score between user a and item i based on their distance in hyperbolic space:

$$\hat{r}_{ai} = \frac{1}{d^2_{\mathcal{L}}(\mathbf{u}_a, \mathbf{v}_i)}. \tag{18}$$

We use adaptive margin loss for model optimization, which is proposed in HICF [51]. The margin loss function pulls the positive samples and pushes the negative samples to the margin, and the adaptive margin strategy assigns a higher margin to nodes which close to the root point in hyperbolic space [35]. Specifically, the adaptive margin loss is computed as follows:

$$\mathcal{L}_r = \sum_{a=0}^{M-1} \sum_{(i,j)\in D_a^r} max(d^2_{\mathcal{L}}(\mathbf{u}_a, \mathbf{v}_i) - d^2_{\mathcal{L}}(\mathbf{u}_a, \mathbf{v}_j) + m_{ai}, 0), \tag{19}$$

where $D_a^r = \{(i,j)|i \in \mathbf{R}_a \wedge j \notin \mathbf{R}_a\}$ denotes the pair-wise training data for user a, and m_{ai} denotes the adaptive margin which is learned by the positive pair (a, i):

$$m_{ai} = \delta(d^2_{\mathcal{L}}(\mathbf{u}_a, \mathbf{o}) + d^2_{\mathcal{L}}(\mathbf{v}_i, \mathbf{o}) - d^2_{\mathcal{L}}(\mathbf{u}_a, \mathbf{v}_i)). \tag{20}$$

We employ Riemannian SGD to optimize the loss function [3], [37]. For model training, we try two popular sampling strategies, random sampling [34] and popularity-based sampling [51]. The overall model implementation is illustrated in Algorithm 1.

Algorithm 1 The Algorithm of *HGSR*

Input: User-user social network $\mathbf{S}$, user-item interaction matrix $\mathbf{R}$;
Output: Parameter $\Theta_s = \mathbf{P}^E$ in hyperbolic social pre-training module, $\Theta_p = \{\mathbf{U}^E, \mathbf{V}^E\}$ in hyperbolic preference learning module;

```
Random initialize parameter Θ_s;
Hyperbolic social embedding initialization (Eq.(7));
while not converged do
    Sample a batch of training data for social pre-training;
    Compute hyperbolic social embeddings with graph convolutions (Eq.(8)-Eq.(10));
    Predict social links with (Eq.(11));
    Update parameter Θ_s with (Eq.(12));
end while
Random initialize parameter Θ_p;
Hyperbolic embedding fusion with pre-trained social embeddings (Eq.(14));
while not converged do
    Sample a batch training data for preference learning;
    Compute hyperbolic user and item embeddings with graph convolutions (Eq.(15)-Eq.(17));
    Predict rating preference with (Eq.(18));
    Update parameters Θ_p with (Eq.(19));
end while
Return Θ_s, Θ_p.
```

TABLE 2
Running time per epoch of different models.

Dataset	Flickr	Ciao	Epinions	Dianping
LightGCN	2.676(s)	2.620(s)	2.692(s)	6.535(s)
DiffNet++	2.902(s)	2.941(s)	2.782(s)	7.857(s)
HICF	2.028(s)	1.988(s)	1.963(s)	10.012(s)
HGSR	3.486(s)	2.341(s)	3.439(s)	16.781(s)

3.4 Discussion

3.4.1 Space Complexity

As illustrated in Algorithm 1, the parameters of *HGSR* are composed of two parts: hyperbolic social pre-training parameters $\Theta_s = \mathbf{P}^E$ and hyperbolic preference learning parameters $\Theta_p = \{\mathbf{U}^E, \mathbf{V}^E\}$. Specifically, our model needs to learn embeddings of $(2M + N)d$ size, while traditional collaborative filtering methods (e.g., BPR [34], LightGCN [12]) need $(M + N)d$ size, the additional part is the pre-trained user social embeddings Md. In general, the user's social network is stable and only needs to be pre-trained once, which is convenient and affordable for recommender systems.

3.4.2 Time Complexity

Compared to graph-based social recommendation models in Euclidean space [12], [46], our model only spends additional time on space transformation as shown in Eq. (5) and Eq. (6). The overall time cost mainly lies in layer-wise propagation. For the social pre-training module, the layer propagation consumptions are $O(|S^+|L_s d)$, where $|S^+|$ and L_s denote the number of non-zero elements in $\mathbf{S}$ and the number of average social neighbors, respectively. For preference learning module, the layer propagation consumptions are $O(|S^+|L_s d) + O(|R^+|(L_u + L_i)d)$, where $|R|^+$ denotes the number of non-zero elements in interaction matrix R, L_u and L_i are the number of average interacted users and average interacted items. Considering the sparse feedback and social connections, $\{L_s, L_u, L_i\} \ll min\{M, N\}$, so the total time complexity is acceptable in practice. For clarity representing the time complexity of the proposed model, we report the running time per epoch of several representative methods. We can find that social recommendation methods (DiffNet++, HGSR) spend more time than collaborative filterings (LightGCN, HICF). Intuitively, social recommendation has an additional social diffusion process, but the time cost is affordable overall.

TABLE 3
The statistics of four datasets.

Dataset	Flickr	Ciao	Epinions	Dianping
Users	8,358	7,375	18,202	59,426
Items	82,120	91091	47,449	10,224
Ratings	327,815	226307	298,173	934,334
Links	187,273	111,781	381,559	813,331
Rating Density	0.048%	0.034%	0.035%	0.154%
Link Density	0.268%	0.206%	0.115%	0.023%

4 EXPERIMENTS

In this part, we conduct extensive experiments on four public datasets to demonstrate the effectiveness of our proposed *HGSR* . We first introduce the experimental settings, and then report the overall performance compared to state-of-the-art baselines. Finally, we investigate each component and give a detailed analysis of *HGSR* .

4.1 Experimental Settings

4.1.1 Datasets

We select four widely used social recommendation datasets: Flickr, Ciao, Epinions, and Dianping. Among them, Flickr[1] is an online image-sharing social platform, and Dianping[2] is a large Chinese location-based social platform. Ciao[3] and Epinions[4] are two popular product review-based social platforms. For all datasets, we transfer the original ratings into binary values. We employ a rating filtering strategy that filters original rating values less than 3 and keeps the remaining ratings as positive feedback. After that, we randomly sample 80% interactions as training data, and the remaining 20% data as test data. The detailed statistics of all datasets are summarized in Table 4.

4.1.2 Baselines and Evaluation Metrics

We select several competing methods for comparisons with our proposed *HGSR* model, including Euclidean and hyperbolic methods. Detailed descriptions are listed as follows:

- **BPR** [34]: BPR is a classic collaborative filtering method. It designs the pairwise ranking loss function which is widely used in implicit feedback based recommendations.
- **GraphRec** [7]: GraphRec incorporates graph neural networks and social recommendation. It captures

1. http://flickr.com/
2. https://lihui.info/data/
3. https://www.ciao.co.uk/
4. http://www.trustlet.org/downloaded epinions.html

TABLE 4
Comparisons of all methods, with "R" representing interaction matrix input and"S" representing social network input. For user preference learning, we use "G" to denote graph formulation, "E" to denote Euclidean modeling, and "H" to denote hyperbolic modeling. Besides, we use "P" to denote social embeddings pre-training.

Models	Model Input		User Preference Learning			
	R	S	G	E	H	P
BPR	✓	×	×	✓	×	×
SocialRec	✓	×	✓	×	×	×
LightGCN	✓	×	✓	×	×	×
DiffNet++	✓	✓	✓	×	×	×
HGCF	✓	×	✓	×	✓	×
HRCF	✓	×	✓	×	✓	×
HICF	✓	×	✓	×	✓	×
HSR	✓	✓	✓	×	✓	×
HyperSoRec	✓	✓	✓	×	✓	×
HGSR	✓	✓	✓	×	✓	✓

both interactions and opinions in the user-item graph and joint social connections for recommendation.

- **LightGCN** [12]: LightGCN simplifies GCNs by removing feature transformation and non-linear activation, and achieves competitive performance for collaborative filtering.
- **DiffNet++** [46]: DiffNet++ is a SOTA graph-based social recommendation model in Euclidean space. It models the recursive social diffusion and interest diffusion process for embedding learning.
- **HGCF** [37]: HGCF is the first attempt to combine GCN and hyperbolic embedding learning for collaborative filtering. It models embeddings in hyperbolic space and designs a skip-connected graph encoder for information propagation.
- **HRCF** [52]: HRCF designs a geometric-aware hyperbolic regularize, which can tackle the over-smoothing issue and make better discrimination.
- **HICF** [51]: HICF investigates the recommendation performances of head/tail item groups on both Euclidean and hyperbolic models, and proposes an adaptive margin loss function with popularity-based sampling strategy to improve recommendation performances in hyperbolic space further.
- **HSR** [21]: HSR designs a hyperbolic aggregator on the user's social neighbors, and introduces an acceleration strategy and attention mechanism for social recommendation.
- **HyperSoRec** [41]: HyperSoRec proposes a hyperbolic social graph encoder with multi-aspect message modeling. It also designs an adaptive metric learning function to capture user influence and item interactions.

As we focus on the item ranking task, we employ two widely used metrics: Recall@N and NDCG@N to evaluate the recommendation performances of various methods. Specifically, for a Top-N ranking list, Recall@N measures the percentage of hit items in the ground truth, and NDCG@N further assigns a higher score to the top-ranked items. All evaluation metrics are computed by an all-ranking protocol that selects all non-interacted items as candidates. All metrics are reported with average values with 10 times of repeated experiments.

4.1.3 Parameter Settings

We initialize all model embeddings with a Gaussian distribution with a mean value of 0 and a standard variance of 0.01, and the embedding dimension is fixed to 64. For Euclidean models, we use Adam with a learning rate of 0.001 and batch size of 1024 to optimize all models. For hyperbolic models, we use the Riemannian SGD [3], [59] with weight decay $1e^{-5}$ learning rate 0.001 and batch size 10000 to optimize all models. For fair comparisons, we refer to the parameters reported by original papers and fine-turn them with gird-search. For our proposed *HGSR* model, we set curvature c=-1 and search GCN layers in the range of $\{1, 2, 3, 4\}$. We implement our model with PyTorch[5] based on TITAN RTX.

4.2 Overall Comparisons

We report the overall recommendation performances of all methods under different Top-N settings from Table 5 to Table 8, and have the following observations:

- Firstly, graph-based recommendation models (GraphRec, LightGCN, DiffNet++) significantly outperform BPR, which demonstrates the superiority of learning preference by high-order graph formulation. Compared with LightGCN, DiffNet++ achieves better performance in most situations, it shows that leveraging social networks can alleviate data sparsity issues in CF and improve recommendation performances.
- Secondly, almost all hyperbolic recommendation models show better performance than Euclidean models. This phenomenon verifies the effectiveness of modeling graph embeddings in hyperbolic space due to its exponential growth capacity and structure preservation ability. When comparing hyperbolic recommendation models, HICF achieves the best performance by benefiting from adaptive margin learning and a negative sampling strategy. Besides, HSR and HyperSoRec only model the social diffusion process in hyperbolic space without graph convolutions on user-item graph, and lead to performance decrease compared to HICF.
- Our proposed *HGSR* model consistently outperforms all methods on all datasets, indicating the effectiveness of hyperbolic social pre-training and heterogeneous preference learning for social recommendation tasks. Compared to DiffNet++ (strongest baseline in Euclidean space), *HGSR* achieves significant improvement on all datasets (e.g., for NDCG@20 metric, about 97% improvement on Flickr, 18% on Ciao, 13% on Epinions and 11% on Dianping). Besides, compared with the best hyperbolic social recommendation model HyperSoRec, our model also achieves impressive improvements, e.g., 48.07% improvement of NDCG@20 on the Flickr dataset and 10.62% improvement on the Ciao dataset. Compared with the strongest hyperbolic CF baseline HICF, our model improves NDCG@20 by about 30.51% and 10.05% on Flickr and Ciao datasets.

5. https://pytorch.org/

TABLE 5
Performance comparisons with different Top-N values on Flickr dataset.

Models	N=10		N=20		N=30		N=40		N=50	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	0.0047	0.0044	0.0074	0.0050	0.0095	0.0056	0.0115	0.0061	0.0133	0.0066
GraphRec	0.0041	0.0041	0.0071	0.0048	0.0089	0.0053	0.0111	0.0059	0.0133	0.0065
LightGCN	0.0070	0.0064	0.0113	0.0074	0.0142	0.0082	0.0170	0.0091	0.0196	0.0098
DiffNet++	0.0078	0.0070	0.0117	0.0078	0.0154	0.0088	0.0183	0.0097	0.0202	0.0103
HGCF	0.0226	0.0102	0.0283	0.0107	0.0317	0.0112	0.0349	0.0118	0.0380	0.0124
HRCF	0.0241	0.0106	0.0290	0.0106	0.0329	0.0112	0.0364	0.0117	0.0395	0.0124
HICF	<u>0.0269</u>	<u>0.0116</u>	<u>0.0335</u>	<u>0.0118</u>	<u>0.0377</u>	<u>0.0123</u>	<u>0.0408</u>	<u>0.0128</u>	<u>0.0438</u>	<u>0.0134</u>
HSR	0.0215	0.0100	0.0271	0.0105	0.0305	0.0109	0.0343	0.0117	0.0377	0.0125
HyperSoRec	0.0170	0.0095	0.0226	0.0104	0.0278	0.0115	0.0319	0.0123	0.0346	0.0130
HGSR	**0.0385**	**0.0156**	**0.0452**	**0.0154**	**0.0495**	**0.0158**	**0.0527**	**0.0161**	**0.0559**	**0.0167**
Improvement	**43.12%**	**34.48%**	**34.93%**	**30.51%**	**31.30%**	**28.46%**	**29.17%**	**25.78%**	**27.63%**	**24.63%**

TABLE 6
Performance comparisons with different Top-N values on Ciao dataset.

Models	N=10		N=20		N=30		N=40		N=50	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	0.0320	0.0276	0.0478	0.0319	0.0582	0.0348	0.0676	0.0374	0.0760	0.0395
GraphRec	0.0342	0.0297	0.0527	0.0348	0.0653	0.0384	0.0757	0.0412	0.0845	0.0434
LightGCN	0.0356	0.0312	0.0560	0.0367	0.0708	0.0408	0.0805	0.0434	0.0890	0.0456
DiffNet++	0.0355	0.0304	0.0563	0.0363	0.0704	0.0404	0.0821	0.0436	0.0921	0.0462
HGCF	0.0336	0.0294	0.0540	0.0352	0.0692	0.0397	0.0815	0.0430	0.0912	0.0456
HRCF	0.0350	0.0298	0.0539	0.0354	0.0694	0.0400	0.0821	0.0433	0.0919	0.0460
HICF	<u>0.0378</u>	<u>0.0323</u>	<u>0.0600</u>	<u>0.0388</u>	0.0768	<u>0.0437</u>	0.0896	<u>0.0471</u>	0.0989	0.0496
HSR	0.0340	0.0282	0.0560	0.0346	0.0709	0.0389	0.0844	0.0426	0.0953	0.0453
HyperSoRec	0.0364	0.0318	0.0600	0.0386	<u>0.0772</u>	0.0434	<u>0.0910</u>	0.0470	<u>0.1018</u>	<u>0.0497</u>
HGSR	**0.0422**	**0.0357**	**0.0674**	**0.0427**	**0.0843**	**0.0475**	**0.0964**	**0.0508**	**0.1068**	**0.0535**
Improvement	**11.64%**	**10.53%**	**12.33%**	**10.05%**	**9.20%**	**8.70%**	**5.93%**	**7.86%**	**4.91%**	**7.65%**

TABLE 7
Performance comparisons with different Top-N values on Epinions dataset.

Models	N=10		N=20		N=30		N=40		N=50	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	0.0335	0.0235	0.0540	0.0297	0.0688	0.0338	0.0816	0.0371	0.0921	0.0397
GraphRec	0.0436	0.0315	0.0681	0.0387	0.0867	0.0437	0.1019	0.0476	0.1170	0.0512
LightGCN	0.0432	0.0314	0.0675	0.0385	0.0850	0.0434	0.1003	0.0473	0.1126	0.0503
DiffNet++	0.0468	0.0329	0.0727	0.0406	0.0901	0.0454	0.1052	0.0492	0.1192	0.0527
HGCF	0.0435	0.0318	0.0678	0.0389	0.0867	0.0442	0.1025	0.0483	0.1160	0.0516
HRCF	0.0449	0.0325	0.0695	0.0397	0.0882	0.0449	0.1046	0.0491	0.1186	0.0526
HICF	<u>0.0502</u>	<u>0.0359</u>	<u>0.0779</u>	<u>0.0440</u>	<u>0.0978</u>	<u>0.0494</u>	<u>0.1155</u>	<u>0.0540</u>	<u>0.1308</u>	<u>0.0577</u>
HSR	0.0418	0.0296	0.0678	0.0373	0.0877	0.0427	0.1036	0.0468	0.1180	0.0503
HyperSoRec	0.0474	0.0337	0.0757	0.0420	0.0956	0.0474	0.1130	0.0518	0.1288	0.0556
HGSR	**0.0519**	**0.0372**	**0.0822**	**0.0460**	**0.1019**	**0.0515**	**0.1202**	**0.0561**	**0.1351**	**0.0597**
Improvement	**3.39%**	**3.62%**	**5.52%**	**4.55%**	**4.19%**	**4.25%**	**4.07%**	**3.89%**	**3.29%**	**3.47%**

The above observations strongly demonstrate our proposed *HGSR* model can effectively exploit social networks and user-item interactions in hyperbolic space. Combining explicit heterogeneous graph learning and implicit social feature enhancement, *HGSR* significantly improves social recommendation performances.

4.3 Investigation of the proposed *HGSR*

Ablation Study. To exploit the effectiveness of each component of our proposed HGSR model, we conduct ablation studies on all datasets. As shown in Table 9, we compare Top-20 recommendation performances of *HGSR* and its variants. Among them, HGSR-w/o P denotes HGSR without hyperbolic social pre-training module, HGSR-w/o S denotes HGSR without social diffusion on preference learning module (only user-item graph propagation), and HGSR-w/o P+S denotes HGSR without pre-training and social diffusion modeling, our method degenerates to HICF. From Table 9, we observe that each variant of *HGSR* shows worse performance than HGSR, which demonstrates the effectiveness of each proposed component. Both implicit and explicit social modeling significantly improve recommendation performances.

Data sparsity Analysis. Here we conduct data sparsity analysis to validate the contribution of the proposed hyperbolic social pre-training module. Specifically, we split

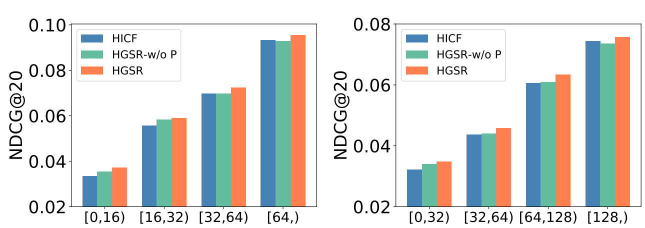
TABLE 8
Performance comparisons with different Top-N values on Dianping dataset.

Models	N=10		N=20		N=30		N=40		N=50	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	0.0428	0.0317	0.0716	0.0410	0.0947	0.0478	0.1147	0.0533	0.1328	0.0580
GraphRec	0.0482	0.0361	0.0799	0.0463	0.1050	0.0537	0.1263	0.0596	0.1451	0.0645
LightGCN	0.0470	0.0351	0.0804	0.0458	0.1064	0.0536	0.1283	0.0597	0.1467	0.0646
DiffNet++	0.0508	0.0380	0.0832	0.0484	0.1094	0.0561	0.1323	0.0624	0.1524	0.0676
HGCF	0.0497	0.0386	0.0829	0.0494	0.1088	0.0572	0.1301	0.0632	0.1486	0.0682
HRCF	0.0506	0.0394	0.0837	0.0501	0.1100	0.0580	0.1323	0.0643	0.1512	0.0694
HICF	0.0519	0.0402	0.0853	0.0509	0.1120	0.0589	0.1347	0.0653	0.1540	0.0705
HSR	0.0474	0.0352	0.0796	0.0457	0.1062	0.0536	0.1287	0.0597	0.1483	0.0649
HyperSoRec	0.0528	0.0397	0.0874	0.0508	0.1141	0.0587	0.1373	0.0652	0.1568	0.0704
HGSR	**0.0556**	**0.0422**	**0.0909**	**0.0536**	**0.1185**	**0.0619**	**0.1411**	**0.0682**	**0.1621**	**0.0737**
Improvement	**5.30%**	**4.98%**	**4.00%**	**5.30%**	**3.86%**	**5.09%**	**2.77%**	**4.44%**	**3.38%**	**4.54%**

TABLE 9
Ablation study of HGSR.

Models	Flickr		Ciao		Epinions		Dianping	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
HGSR-w/o P+S	0.0335	0.0118	0.0600	0.0388	0.0779	0.0440	0.0853	0.0509
HGSR-w/o P	0.0310	0.0111	0.0644	0.0409	0.0792	0.0443	0.0903	0.0532
HGSR-w/o S	0.0337	0.0116	0.0659	0.0417	0.0755	0.0427	0.0836	0.0460
HGSR	**0.0452**	**0.0154**	**0.0674**	**0.0425**	**0.0822**	**0.0460**	**0.0909**	**0.0536**

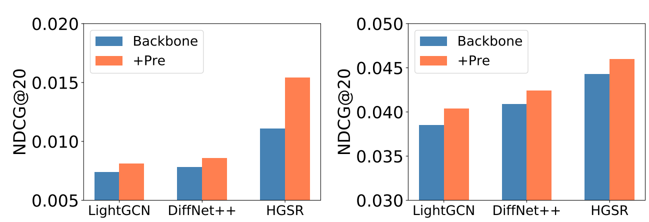
all users into different groups according to their training records, and compare their performances of different recommendation models. As shown in Fig. 4, we present comparisons on four sparsity user groups. Among, HGSR-w/o P denotes that HGSR without social pre-training module. Compared with the corresponding CF backbone (HICF), HGSR-w/o P achieves improvements in most sparse groups, while showing a little decrease in the densest group. It indicates that explicit graph learning alleviates the data sparsity issue, while also introducing noise for those active users, these experimental phenomenons are also revealed in DiffNet [47]. Luckily, we find that our proposed *HGSR* consistently outperforms HICF in each user group. This verifies the effectiveness of tackling the social diffusion noise issue by implicit feature enhancement with hyperbolic pre-trained social embeddings.



(a) Ciao Dataset (b) Epinions Dataset

Fig. 4. Performance comparisons under different sparsity user groups.

Generalization Analysis We conduct experiments to investigate the generalization of the pre-trained hyperbolic social embeddings. Specifically, we combine the pre-trained hyperbolic social embeddings with SOTA Euclidean and Hyperbolic recommendation models. As shown in Fig. 5, we select LightGCN, DiffNet++, and HGSR-w/o P as backbones, and compare their performances and corresponding variants (combined with the pre-trained hyperbolic social embeddings). We find that the pre-trained hyperbolic social embeddings can significantly improve each embedding-based recommendation method, either Euclidean models (i.e., LightGCN, DiffNet++) or Hyperbolic model (i.e., HGSR-w/o P). It demonstrates that our proposed hyperbolic pre-training module presents a good generalization ability, which can easily couple with other embedding-based recommendation methods and enhance their performances.



(a) Flickr Dataset (b) Epinions Dataset

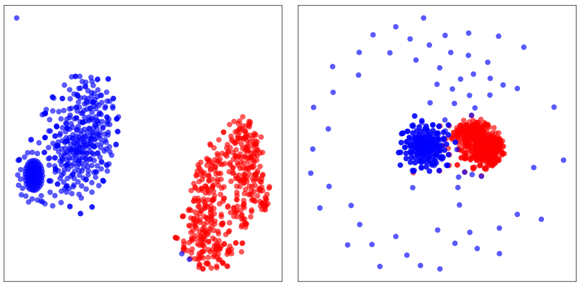
Fig. 5. Generalization of the pre-trained hyperbolic social feature.

4.4 Comparisons with Different Pre-training Methods

In this part, we compare the proposed hyperbolic social pre-training method with its counterpart in Euclidean space, we keep all model structures the same and only compare different spaces. As shown in Table 10, we present the experimental results of different social pre-training methods for recommendation. Among them, HGSR-w/o P is the backbone model without any pre-training process, SocialPre(E) denotes social pre-training in Euclidean space, and SocialPre(H) denotes social pre-training in Hyperbolic space. We can find that our proposed hyperbolic social pre-training method achieves better performances in all settings,

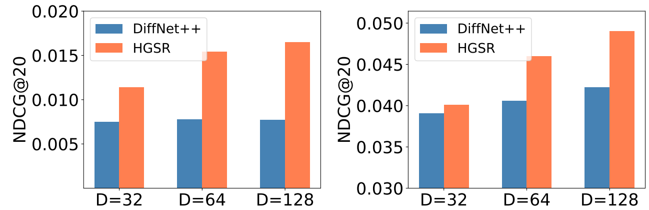
TABLE 10
Comparisons of social pre-training methods in different spaces.

Models	Flickr		Epinions	
	Recall@20	NDCG@20	Recall@20	NDCG@20
HGSR-w/o P	0.0310	0.0111	0.0792	0.0443
+SocialPre(E)	0.0381	0.0139	0.0805	0.0448
+SocialPre(H)	**0.0452**	**0.0154**	**0.0822**	**0.0460**



(a) Hyperbolic pre-training (b) Euclidean pre-training

Fig. 6. Visualization of the pre-trained social embeddings on the Epinions dataset, where red nodes are head users and blue nodes are tail users.



(a) Flickr Dataset (b) Epinions Dataset

Fig. 7. Performance comparisons under different embedding sizes.

verifying the superiority of improving recommendation with hyperbolic pre-training. Besides, we visualize the pre-trained social embeddings under hyperbolic learning and Euclidean learning. As we have no available labels to distinguish nodes in the social networks, we split users into head users and tail users according to their social neighbors. Then, we randomly sample 500 users from head and tail users, respectively, and illustrate their embedding distributions. As illustrated in Fig. 6, we observe that hyperbolic pre-training presents better discrimination than Euclidean pre-training, which can reflect the head/tail structure of the graph.

4.5 Detailed Model Analysis

Embedding Size. We compare our proposed *HGSR* model and DiffNet++ performances under different embedding sizes. As shown in Fig. 7, we report NDCG@20 of both models on the Flickr and Epinions datasets, where embedding sizes are selected from $\{32, 64, 128\}$. We observe that *HGSR* consistently outperforms DiffNet++, which verifies the effectiveness of promoting recommendation performances under different embedding sizes. Besides, we find that the performance of *HGSR* quickly increases when embedding size increases, while DiffNet++ is more slight, verifying the capacity of *HGSR* increases more than DiffNet++ with a larger embedding size.

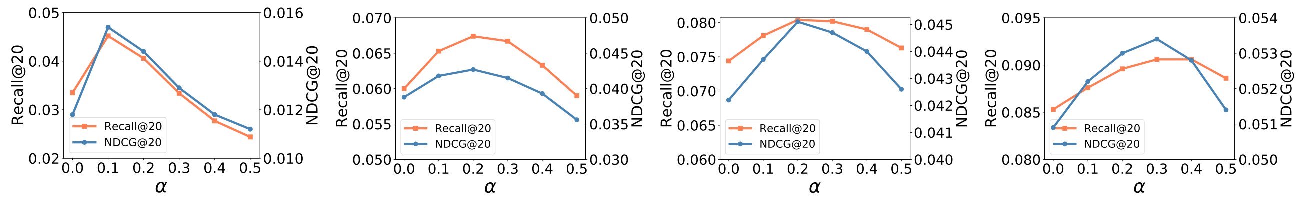
Parameter Sensitivity. To investigate the influence of the social propagation part on heterogeneous graph learning, we conduct experiments under different social balance weights. As illustrated in Fig. 8, we represent Recall@20 and NDCG@20 of various weights α on all datasets. Please note that $\alpha = 0$ means the social diffusion part disappears, we also remove the social pre-training module, and then *HGSR* degenerates to HICF. From the experimental results, we can find that *HGSR* achieves the best performance with different α for different datasets. Specifically, *HGSR* obtains the best performance with $\alpha = 0.1$ on Flickr, $\alpha = 0.2$ on Ciao and Epinions, and $\alpha = 0.3$ on the Dianping dataset. Different dataset properties have different social influences on user interaction behavior. Besides, *HGSR* doesn't show better performance when α increases, which means that a suitable social weight setting is important to exactly model user preferences.

Propagation Layers. We empirically study the effects of different propagation layers of Graph encoder for preference learning. As shown in Table 11, we compare *HGSR* performances under different graph propagation layers. We can find that the recommendation performances increase quickly and then drop when the propagation layers keep deeper. This indicates that over-smoothing issues also limit the performances of hyperbolic graph learning for social recommendation. When propagation layer $L = 3$, our *HGSR* reaches the best performance on Flickr, Epinions, and Dianping Datasets, while $L = 4$ on Ciao datasets. The reason is that the Ciao dataset has the sparsest interactions on all datasets. Therefore, a proper propagation layer is important to balance the high-order message passing and over-smoothing issues simultaneously.

5 Related Work

5.1 Social Recommendation

Recommender systems provide personalized suggestions for each user by modeling users' preferences. Classical collaborative filtering methods [28], [34], [33] project both users and items into a low dimensional latent space, then recommend item lists based on inner product scores. With the development of deep learning, neural network based methods have been proposed to tackle collaborative filtering through modeling the non-linear interactions [13]. Although widely applied, CF methods are usually far from satisfactory due to users' sparse interactions. Following the social influence and social homogeneity theory [18], [20], [24], the social recommendation has emerged as a popular research direction, which utilizes the additional social network to alleviate data sparsity issue and improve recommendation performances. Early studies leverage social networks in shallow form, which can be divided into two classes: social regularization-based models [27], [15], [16] and user behavior enhancement-based models [10], [11]. Social regularization-based models assume that two connected users share a similar preference, and then an additional regularization term is added to the ranking optimization objective [27]. Instead of adding regularization, TrustSVD regards each social neighbor's interacted items as



(a) Flickr Dataset (b) Ciao Dataset (c) Epinions Dataset (d) Dianping Dataset

Fig. 8. Performance comparisons under different values of parameter α.

TABLE 11
Recommendation performances with different propagation layers L.

Models	Flickr		Ciao		Epinions		Dianping	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
L=1	0.0256	0.0103	0.0606	0.0371	0.0727	0.0402	0.0850	0.0496
L=2	0.0303	0.0124	0.0654	0.0408	0.0804	0.0451	0.0864	0.0504
L=3	**0.0452**	**0.0154**	0.0673	0.0425	**0.0822**	**0.0460**	**0.0909**	**0.0536**
L=4	0.0323	0.0126	**0.0674**	**0.0427**	0.0808	0.0455	0.0824	0.0486

the auxiliary feedback to predict the user's unknown preference [10]. Besides, CNSR proposes to leverage the global social network structure to learn user preference based on social embedding learning [48]. These social recommendation models achieved better recommendation performances than CF models. However, they focus on the first-order social structure and fail to fully exploit the global social network in the modeling process.

5.2 Neural Graph based Recommendation

Recently, GCNs have received success for graph learning based tasks [17], [39]. Researchers have adopted the key ideas of graph convolutions for graph-based recommendations, which have shown state-of-the-art performance [56], [12], [49], [54]. Different from traditional CF approaches that learn user and item embeddings with matrix factorization [28], [34], neural graph based CF methods formulate user-item interactions as a bipartite graph. Then, these neural graph models learn user and item embeddings by exploiting the high-order collaborative signal through multiple graph convolutions. LightGCN is a representative work in neural graph based CF models, the basic paradigm is that discard the additional feature transformation and non-linear activation in GCNs and only perform neighbors aggregation for embedding learning [12].

Some researchers also propose graph-based social recommendation models to extend neural graph CF models, which joint model user-user social influence diffusion and user-item interest propagation [7], [47], [46], [57], [26]. For example, GraphRec formulates interactions and opinions in a user-item bipartite graph, then joint user-user social graph for embedding learning [7]. DiffNet considers high-order social influence diffusion and models social influence diffusion from user-user social network for user representation learning. DiffNet++ extends DiffNet by combining social influence diffusion from user-user social network and interest propagation processes from user-item behavior with attention mechanism [46]. Besides, RecoGCN considers multi-relation social connections and proposes a relation-aware GCN model to formulate user embeddings [50]. MCNE designs a conditional GNN that aims to learn user similarity in both user-item interaction graph and user-user social networks [42]. ESFR proposes adversarial graph convolutional networks to enhance recommendation performances by social graph generation [57].

All these graph based (social) recommendation methods learn users and item embeddings in Euclidean space, then formulate the high-order graph structure to enhance recommendation. However, they fail to capture the hierarchical graph structure and representation distortion will lead to sub-optimal results. Considering the hierarchical structure of graphs, in this paper, we formulate the social recommendation task in hyperbolic space and propose the HGSR model.

5.3 Hyperbolic Learning and Applications in Recommendation

Hyperbolic space is a non-Euclidean space with a negative curvature, which has shown great potential for representation learning of tree-like hierarchical data. Poincare ball and Lorentz formulation are two efficient ways to learn representations in hyperbolic space [29], [30], [38]. Considering graph data usually present a hierarchical structure of power-law distribution, researchers have proposed a series of works that generalize hyperbolic embedding learning to graph neural networks [23], [5], [61], [25], such as HGNN [23], HGCN [5] and HGAT [61]. These works represent nodes in the hyperbolic space, and perform graph convolutions by injecting nodes from hyperbolic space to the (Euclidean) tangent space, which builds a bridge between GCN and hyperbolic learning. Besides, some works also learn heterogeneous graph embeddings in hyperbolic space [45], [43], [43]. Recently, researchers have applied hyperbolic graph learning to various recommendation tasks, such as collaborative filtering [37], [52], [51], sequential recommendation [22], [55]. HGCF proposes a hyperbolic GCN model for collaborative filtering [37], and HRCF designs an additional hyperbolic geometric regularization to enhance performance [52]. HICF analyzes recommendation performances between head and tail items under Euclidean modeling and hyperbolic modeling, and then proposes an adaptive margin learning method with popularity-based negative sampling [51]. There are also some studies that leverage hyperbolic learning to enhance social recommendation. HSCML introduces a hyperbolic metric learning based on social connections [60]. HSR designs a hyperbolic aggregator to combine social neighbors for representation learning [21]. HyperSoRec exploits hyperbolic social embeddings with multiple aspect learning [41].

These hyperbolic graph-based recommendation models show superior performances compared with their counterparts in Euclidean spaces. However, we argue that current solutions of hyperbolic social recommendation are still far from satisfactory. In fact, current hyperbolic graph methods only model social networks or user-item graph in hyperbolic space separately [21], [41], [37], [51], lacking efficient fusion of both kinds of graphs for representation learning. In this paper, we formulate the user-item graph and social network as a heterogeneous graph, then learn users' preferences with hyperbolic graph learning for the social recommendation. Compared with current hyperbolic heterogeneous graph embedding methods that directly use hyperbolic distance instead of Euclidean distance [45], [43], [43], our proposed HGSR uses graph neural networks to learn the heterogeneous graph in hyperbolic space, which can better preserve the graph structure.

6 Conclusion

In this paper, we propose a novel *HGSR* model for the hyperbolic social recommendation. To exploit the heterogeneity and the noise issue introduced by social influence diffusion, we design a social pre-training enhanced hyperbolic heterogeneous graph learning method. Specifically, we first pre-train social networks in hyperbolic space, which can preserve the hierarchical structure properties. Next, we feed the pre-trained social embeddings into a hyperbolic heterogeneous graph for preference learning. Such that, we combine explicit heterogeneous graph learning implicit social feature enhancement for hyperbolic social recommendation, which can effectively tackle heterogeneity and noise issues. Finally, extensive experimental results on four real-world datasets clearly demonstrate the effectiveness of our proposed model compared to state-of-the-art baselines, including high performance, generalization of the pre-trained feature, and applicability to various sparsity users. In the future, we aim to exploit more hyperbolic graph learning techniques for recommendation, such as more effective hyperbolic graph pre-training, hyperbolic self-supervised graph learning, and so on.

Acknowledgements

This work was supported in part by grants from the National Key Research and Development Program of China(Grant No. 2021ZD0111802), the National Natural Science Foundation of China(Grant No. U23B2031, 721881011, 61932009, 62376086, U22A2094), and the CCF-AFSG Research Fund (Grant No. CCF-AFSG RF20210006).

References

[1] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *SIGKDD*, pages 7–15, 2008.
[2] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 2012.
[3] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *TAC*, 58(9):2217–2229, 2013.
[4] I. Chami, A. Gu, V. Chatziafratis, and C. Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. In *NeurIPS*, volume 33, pages 15065–15076, 2020.
[5] I. Chami, Z. Ying, C. Ré, and J. Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, pages 4869–4880, 2019.
[6] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *AAAI*, pages 27–34, 2020.
[7] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. Graph neural networks for social recommendation. In *WWW*, pages 417–426, 2019.
[8] M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310, 1948.
[9] O. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic neural networks. In *NeurIPS*, pages 5350–5360, 2018.
[10] G. Guo, J. Zhang, and N. Yorke-Smith. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *AAAI*, pages 123–129, 2015.
[11] G. Guo, J. Zhang, and N. Yorke-Smith. A novel recommendation model regularized with user trust and item ratings. *IEEE TKDE*, 28(7):1607–1620, 2016.
[12] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pages 639–648, 2020.
[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.
[14] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.
[15] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.
[16] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang. Scalable recommendation with social contextual information. *IEEE TKDE*, 26(11):2789–2802, 2014.
[17] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
[18] A. D. Kramer, J. E. Guillory, and J. T. Hancock. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790, 2014.
[19] M. Law, R. Liao, J. Snell, and R. Zemel. Lorentzian distance learning for hyperbolic representations. In *ICML*, pages 3672–3681, 2019.
[20] K. Lewis, M. Gonzalez, and J. Kaufman. Social selection and peer influence in an online social network. *Proceedings of the National Academy of Sciences*, 109(1):68–72, 2012.
[21] A. Li, B. Yang, F. K. Hussain, and H. Huo. Hsr: hyperbolic social recommender. *Inf. Sci.*, 585:275–288, 2022.
[22] Y. Li, H. Chen, X. Sun, Z. Sun, L. Li, L. Cui, P. S. Yu, and G. Xu. Hyperbolic hypergraphs for sequential recommendation. In *CIKM*, pages 988–997, 2021.
[23] Q. Liu, M. Nickel, and D. Kiela. Hyperbolic graph neural networks. In *NeurIPS*, pages 8228–8239, 2019.
[24] Q. Liu, B. Xiang, E. Chen, H. Xiong, F. Tang, and J. X. Yu. Influence maximization over large-scale social networks: A bounded linear approach. In *CIKM*, pages 171–180, 2014.
[25] X. Liu, Y. Zhu, and X. Wu. Joint user profiling with hierarchical attention networks. *Frontiers of Computer Science*, 17(3):173608, 2023.
[26] J. Luo, M. He, W. Pan, and Z. Ming. Bgnn: Behavior-aware graph neural network for heterogeneous session-based recommendation. *Frontiers of Computer Science*, 17(5):175336, 2023.
[27] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *WSDM*, pages 287–296, 2011.
[28] A. Mnih and R. R. Salakhutdinov. Probabilistic matrix factorization. In *NeurIPS*, pages 1257–1264, 2008.

[29] M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, pages 6338–6347, 2017.
[30] M. Nickel and D. Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *ICML*, pages 3779–3788, 2018.
[31] G. Pickering, J. Bull, and D. Sanderson. Sampling power-law distributions. *Tectonophysics*, 248(1-2):1–20, 1995.
[32] E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Phys. Rev. E*, 67:026112, Feb 2003.
[33] S. Rendle. Factorization machines. In *ICDM*, pages 995–1000. IEEE, 2010.
[34] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.
[35] F. Sala, C. De Sa, A. Gu, and C. Ré. Representation tradeoffs for hyperbolic embeddings. In *ICML*, pages 4460–4469, 2018.
[36] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. 2007.
[37] J. Sun, Z. Cheng, S. Zuberi, F. Pérez, and M. Volkovs. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *WWW*, pages 593–601, 2021.
[38] Y. Tan, C. Yang, X. Wei, C. Chen, L. Li, and X. Zheng. Enhancing recommendation with automated tag taxonomy construction in hyperbolic space. In *ICDE*, pages 1180–1192. IEEE, 2022.
[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In *ICLR*, 2018.
[40] F. E. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Auton Agent Multi-Agent Syst*, 16(1):57–74, 2008.
[41] H. Wang, D. Lian, H. Tong, Q. Liu, Z. Huang, and E. Chen. Hypersorec: Exploiting hyperbolic user and item representations with multiple aspects for social-aware recommendation. *TOIS*, 40(2):1–28, 2021.
[42] H. Wang, T. Xu, Q. Liu, D. Lian, E. Chen, D. Du, H. Wu, and W. Su. Mcne: an end-to-end framework for learning multiple conditional network representations of social network. In *SIGKDD*, pages 1064–1072, 2019.
[43] L. Wang, C. Gao, C. Huang, R. Liu, W. Ma, and S. Vosoughi. Embedding heterogeneous networks into hyperbolic space without meta-path. In *AAAI*, volume 35, pages 10147–10155, 2021.
[44] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua. Neural graph collaborative filtering. In *SIGIR*, pages 165–174, 2019.
[45] X. Wang, Y. Zhang, and C. Shi. Hyperbolic heterogeneous information network embedding. In *AAAI*, volume 33, pages 5337–5344, 2019.
[46] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE TKDE*, 2020.
[47] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang. A neural influence diffusion model for social recommendation. In *SIGIR*, pages 235–244, 2019.
[48] L. Wu, P. Sun, R. Hong, Y. Ge, and M. Wang. Collaborative neural social recommendation. *IEEE TSMC*, 51(1):464–476, 2018.
[49] L. Wu, Y. Yang, K. Zhang, R. Hong, Y. Fu, and M. Wang. Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. In *SIGIR*, pages 679–688, 2020.
[50] F. Xu, J. Lian, Z. Han, Y. Li, Y. Xu, and X. Xie. Relation-aware graph convolutional networks for agent-initiated social e-commerce recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 529–538, 2019.
[51] M. Yang, Z. Li, M. Zhou, J. Liu, and I. King. Hicf: Hyperbolic informative collaborative filtering. In *SIGKDD*, pages 2212–2221, 2022.
[52] M. Yang, M. Zhou, J. Liu, D. Lian, and I. King. Hrcf: Enhancing collaborative filtering via hyperbolic geometric regularization. In *WWW*, pages 2462–2471, 2022.
[53] Y. Yang, L. Wu, R. Hong, K. Zhang, and M. Wang. Enhanced graph learning for collaborative filtering via mutual information maximization. In *SIGIR*, pages 71–80, 2021.
[54] Y. Yang, Z. Wu, L. Wu, K. Zhang, R. Hong, Z. Zhang, J. Zhou, and M. Wang. Generative-contrastive graph learning for recommendation. In *SIGIR*, pages 1117–1126, 2023.
[55] J. Ye, Z. Zhang, L. Sun, Y. Yan, F. Wang, and F. Ren. Sincere: Sequential interaction networks representation learning on co-evolving riemannian manifolds. In *WWW*, pages 360–371, 2023.
[56] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*, pages 974–983, 2018.
[57] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui. Enhance social recommendation with adversarial graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, 2020.
[58] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *WWW*, pages 413–424, 2021.
[59] H. Zhang, S. J Reddi, and S. Sra. Riemannian svrg: Fast stochastic optimization on riemannian manifolds. *NeurIPS*, 29, 2016.
[60] S. Zhang, H. Chen, X. Ming, L. Cui, H. Yin, and G. Xu. Where are we in embedding spaces? In *SIGKDD*, page 2223–2231, 2021.
[61] Y. Zhang, X. Wang, C. Shi, X. Jiang, and Y. F. Ye. Hyperbolic graph attention network. *Trans. on BigData*, 2021.

Yonghui Yang is currently working towards a Ph.D. degree at Hefei University of Technology, China. He received his master's degree from the same University in 2021. He has published several papers in referred conferences and journals, such as SIGIR, IJCAI, ACM Multimedia, and IEEE TBD. His research interests include graph learning and recommender systems.

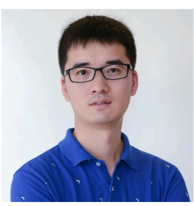
Le Wu is currently a professor at the Hefei University of Technology (HFUT), China. She received her Ph.D. degree from the University of Science and Technology of China (USTC). Her general area of research interests are data mining, recommender systems, and social network analysis. She has published more than 50 papers in referred journals and conferences, such as IEEE TKDE, SIGIR, WWW, and AAAI. Dr. Le Wu is the recipient of the Best of SDM 2015 Award, and the Distinguished Dissertation Award from the China Association for Artificial Intelligence (CAAI) 2017.

Kun Zhang received a Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2019. He is currently a faculty member at the Hefei University of Technology (HFUT), China. His research interests include Natural Language Understanding and Recommender Systems. He has published several papers in refereed journals and conferences, such as the IEEE TSMC:S, IEEE TKDE, ACM TKDD, AAAI, KDD, ACL, ICDM. He received the KDD 2018 Best Student Paper Award.

Richang Hong is currently a professor at Hefei University of Technology. He received the Ph.D. degree from USTC, in 2008. He has published more than 100 publications in the areas of his research interests, which include multimedia question answering, video content analysis, and pattern recognition. He is a member of the Association for Computing Machinery. He was a recipient of the Best Paper award in the ACM Multimedia 2010.

Hailin Zhou is the founding director of IVY MOBIL. Currently, he is also pursuing a Ph.D. candidate in Hefei University of Technology. He obtained a bachelor's degree from Hefei University of Technology. His research interests are user modeling and recommendation systems.

Zhiqiang Zhang is currently a Senior Staff Engineer at Ant Group. His research interests mainly focus on graph machine learning. He has led a team to build an industrial graph machine learning system, AGL, in Ant Group. He has published more than 40 papers in top-tier machine learning and data mining conferences, including NeurIPS, VLDB, SIGKDD, and AAAI.



Jun Zhou is currently a Principle Engineer at Ant Group. His research mainly focuses on machine learning and data mining. He has participated in the development of several distributed systems and machine learning platforms in Alibaba and Ant Financial, such as Apsaras (Distributed Operating System) and KunPeng (Parameter Server). He has published more than 40 papers in top-tier machine learning and data mining conferences.

Meng Wang is a professor at the Hefei University of Technology, China. He received his B.E. degree and Ph.D. degree in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, in 2003 and 2008, respectively. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored more than 200 book chapters, journal and conference papers in these areas. He is the recipient of the ACM SIGMM Rising Star Award 2014. He is an associate editor of IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), IEEE Transactions on Circuits and Systems for Video Technology (IEEE TCSVT), IEEE Transactions on Multimedia (IEEE TMM), and IEEE Transactions on Neural Networks and Learning Systems (IEEE TNNLS).