IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 25 October 2023, accepted 16 December 2023, date of publication 25 December 2023, date of current version 3 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347249

RESEARCH ARTICLE

A Graph Neural Networks-Based Learning Framework With Hyperbolic Embedding for Personalized Tag Recommendation

CHUNMEI ZHANG[1], AORAN ZHANG[2], LI ZHANG[3], (Senior Member, IEEE), YONGHONG YU[4], WEIBIN ZHAO[4], AND HAI GENG[5]

[1]School of Artificial Intelligence, Nanjing Vocational College of Information Technology, Nanjing 210023, China
[2]School of Computer, Jiangsu University of Science and Technology, Zhenjiang 212100, China
[3]Department of Computer Science, Royal Holloway, University of London, TW20 0EX Surrey, U.K.
[4]College of Tongda, Nanjing University of Posts and Telecommunication, Yangzhou 225127, China
[5]School of Tourism Management, Nanjing Institute of Tourism and Hospitality, Nanjing 211100, China

Corresponding author: Chunmei Zhang (zhangcm@njcit.cn)

This work was supported in part by the Future Network Scientific Research Fund Project under Grant FNSRFP-2021-YB-54.

ABSTRACT Learning high-quality representations of users, items, and tags from historical interactive data is crucial for personalized tag recommendation (PTR) systems. Currently, most PTR models are committed to learning representations from first-order interactions without considering the exploitation of high-order interactive relations, which can be beneficial for avoiding sub-optimal learning. Although several PTR models equipped with graph neural networks (GNN) have been proposed to capture higher-order semantic relevance from raw data, they all carry out representation learning in Euclidean space, which can still easily result in sub-optimal learning due to embedding distortion. In order to further improve the quality of representation learning for PTR, the paper proposes a novel PTR model based on a lightweight GNN framework with hyperbolic embedding, namely GHPTR. GHPTR explicitly injects higher-order relevance into entity representation through the message propagation and aggregation mechanism of GNN and leverages hyperbolic embedding to alleviate the embedding distortion problem. Experimental results on real-world datasets have demonstrated the superiority of our model over its Euclidean counterparts and state-of-the-art baselines.

INDEX TERMS Tag recommendation, graph neural networks, hyperbolic geometry, representation learning, embedding.

I. INTRODUCTION

Social tagging gained popularization with the growth of social networking websites. These sites allow users to add terms or keywords, which are most known as tags, to images, videos, and other online items. Social tagging is an efficient tool for users to annotate and organize online items and a dependable aid for websites in delivering information services. It has become indispensable in numerous web platforms and applications. Meanwhile, many personalized tag recommendation (PTR) systems [1] have been developed with the popularity of social tagging. These systems aim to promote a virtuous circle of social tagging services and facilitate the tagging process by automatically suggesting lists of candidate tags for users to select.

Like the general recommender systems oriented to users' preferences, the PTR is usually modeled as a ranking problem, and the learning-to-rank (L2R) techniques have been widely adopted to tackle it. The dominant paradigm for L2R-based PTR is learning to represent entities including users, items, and tags from their ternary interactions in a low-dimensional embedding space, then generating a ranked list of tags based on learned embeddings. Among such learning techniques, those [2], [3], [4], [5], [6] related to tensor

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Mei.

factorization used to be the most competitive because the interactions of triples (user, item, tag) constitute the primary content of raw data, which can be represented as three-order tensors. However, most tensor factorization-based models are committed to learning shallow representations from direct (a.k.a first-order) interactive relations, and the learned representations can not precisely characterize entities' properties for the lack of semantics. Furthermore, the ternary interactions derived from the social tagging system naturally constitute a tripartite graph [7]. From the perspective of graph learning, a considerable amount of semantic relevance lurks in the high-order connected paths of the graph. Such high-order semantic information is beneficial to recommendations for their ability to reveal the underlying properties of entities, e.g., users' potential preferences on tags.

Recently, some studies [8] have incorporated graph neural networks (GNN) [9], [10], [11] into the framework of PTR models to leverage high-order relevance in raw data. By utilizing the message propagation and aggregation mechanism of GNN, these models are capable of encoding high-order semantic relevance into entity representations, thereby enhancing the performance of PTR models. However, despite the effectiveness of GNN-based PTR models, their abilities to express graph data are limited by Euclidean spaces. These models are designed to learn representations in Euclidean space, which aligns with people's intuition and is easy to visualize. More importantly, Euclidean space has complete and mature vector operators. Meanwhile, many data with graph structure always exhibit the properties of complex networks [12] such as scale-free and power-law distribution, and the tripartite graph of ternary interactions in the PTR system also has such properties [13]. Related studies have shown that Euclidean spaces are not the most suitable geometric representation for complex networks [14], [15]. The power-law distribution of networks suggests that their overall structure is tree-like. In a tree, the number of nodes increases exponentially with the depth of the tree, while the volume of Euclidean spaces increases polynomially with distance from the origin point. The distortion problem will arise when embedding tree-like data in Euclidean spaces, resulting in sub-optimal learning.

Hyperbolic space has emerged as a promising tool for modeling hierarchical or tree-like data in recent times [14], [15], [16], [17]. Unlike Euclidean space, which has zero curvature, hyperbolic space is a non-Euclidean space with constant negative curvature. When a disk is embedded into a two-dimensional hyperbolic space with curvature $c = -1$, its circumference $(2\pi \sinh r)$ and area $(2\pi(\cosh r - 1))$ grow exponentially with the radius r. On the other hand, in the two-dimensional Euclidean space, the corresponding circumference $(2\pi r)$ grows linearly and area (πr^2) grows quadratically. This makes hyperbolic space akin to a continuous version of a tree, making it well-suited for embedding graph data with lower distortion than in the Euclidean space.

Both GNN and hyperbolic embedding are universal learning algorithms. The universality of GNN lies in its message propagation mechanism, i.e., the aggregation of neighbor nodes, which is suitable for capturing the local structural properties of graphs. On the other hand, most graphs have global properties such as scale-free and power-law distribution. These properties can not be directly reflected by GNN but can be well presented by hyperbolic space. These two algorithms have recently been integrated to enhance the representation learning of user-item bipartite graphs for item recommendation and achieved promising results. In contrast to item recommendation models, PTR models must deal with user-tag-item tripartite graphs, which have a structure closer to complex networks than bipartite graphs. Therefore, how to introduce hyperbolic embedding and GNN to the framework of PTR models is a meaningful issue.

With the expectation of further enhancing the performances of PTR systems, in this paper, we propose a graph neural networks-based learning framework with hyperbolic embedding for PTR, namely GHPTR, which utilizes GNN to exploit high-order semantic relevance among entities and employs hyperbolic embedding to alleviate the problem of embedding distortion. In the first phase, GHPTR leverages the GNN to capture the semantic relevances in high-order connected paths and encode them into nodes' representations. Specifically, we derive two bipartite graphs from the tripartite interactive graph, i.e., the user-tag graph and the item-tag graph. Then the proposed model represents every node by explicitly aggregating representations of its multi-hop neighbors on each graph. Moreover, we remove feature transformation and non-linear activation components of GNN to make the proposed model more lightweight. The second phase of GHPTR accounts for modeling the interactions between nodes via embedding them into hyperbolic space and calculating the hyperbolic distances between embeddings for the final prediction. We conduct experiments on two real datasets to validate the effectiveness of the proposed model, and the experimental results have shown its superiority over state-of-the-art baselines.

Our major contributions can be summarized as follows:

- We introduce a GNN with a lightweight architecture to the framework of PTR, which can exploit the local properties of interactive tripartite graph and reduce computational consumption.
- We utilize hyperbolic embedding to improve the expressive ability of the proposed model, which can better accommodate the global properties of interactive data and alleviate the problem of embedding distortion.
- We conduct extensive experiments on real-world datasets to verify the efficiency of the proposed model, and experimental results show that the proposed model can outperform the state-of-the-art baselines.

The rest of the paper is organized as follows: The related work is discussed in Section II. Section III presents the problem definition of the personalized tag recommendation and formalized description of hyperbolic space.

In Section IV, we describe the details of the proposed model. The effectiveness of the proposed model is demonstrated via extensive experiments in Section V, followed by conclusions and future works in Section VI.

II. RELATED WORK

This section briefly reviews the state-of-the-art related works, including PTR models, GNN-based recommendation models, and hyperbolic recommendation models.

A. PERSONALIZED TAG RECOMMENDATION MODELS

Social tagging systems have become popular in various web applications, making personalized tag recommendation (PTR) an attractive issue in the research of recommender systems. The core of historical tagging information in PTR system is the ternary interaction among entities, which includes user, item, and tag. These interactions can be represented by a three-order tensor. Consequently, most classical studies utilize tensor factorization techniques, such as the tucker decomposition (TD), to learn representations of involved entities in PTR tasks [2], [4], [5].

The TD model's computation cost becomes impractical for large-scale PTR tasks due to its model equation resulting in a cubic runtime with respect to the factorization dimension. To meet this challenge, Rendle et al. proposed the pairwise interaction tensor factorization (PITF) model [4], which explicitly models the pairwise interactions between entities with linear runtime. PITF is widely recognized for its superior performance, and many learning methods derived from it have been proposed to fit various PTR scenarios. Recently, to leverage the end-to-end learning capability of deep neural networks (DNN), several learning frameworks based on DNN [6], [18], [19] have been developed to further improve the performance of traditional PTR models.

Note that all the above models are conducted in Euclidean spaces. As we mentioned before, their capabilities of learning the representations of tree-like data are restricted by the polynomial expansion property of Euclidean space. Meanwhile, these models have overlooked the semantic relevance hidden in high-order interactions, and the embeddings they learned are only derived from first-order interactions.

B. GNN-BASED RECOMMENDATION MODELS

Graph Neural Networks (GNN) [9], [10] are a class of deep learning methods designed to perform inference on data described by graphs. GNN can be directly applied to graphs and provide an easy way to do node-level, edge-level, and graph-level prediction tasks. Since the target of the recommender system can be viewed as the link-prediction task, many recommendation models [20], [21], [22], [23], [24], [25], [26] have adopted GNN to improve representation learning. Representative GNN-based recommendation methods include but are not limited to SR-GNN [27], NGCF [21], LightGCN [22], and GraphRec [23]. Wu et al. proposed a session-based recommendation model using GNN, namely SR-GNN [27]. SR-GNN converts session sequences to graphs and utilizes GNN to capture the inner patterns of items' transitions. The NGCF model [21] employs the graph convolutional networks (GCN) to carry out message propagation and aggregation on the user-item interactive bipartite graph and fully explores the higher-order similarities between entities to achieve better collaborative filtering performance. In [22], He et al. found that the two most common components of GCN, i.e., feature transformation and nonlinear activation, contribute little to collaborative filtering and increase the difficulty of training. Therefore they simplified the GCN to a lightweight version called LightGCN. LightGCN retains only the aggregation component, which is closely related to collaborative filtering, and only performs linear message propagation on the bipartite graph to learn the representations of users and items. Fan et al. proposed a GNN-based learning framework GraphRec [23] to coherently model different bipartite graphs and strengths of social relations for the social recommendation.

The above GNN-based recommendation models are designed to deal with bipartite graph information, aiming at the traditional item recommendation tasks. The representative application of GNN in the study of PTR is the GNN-PTR model proposed in [8], which decomposed the tripartite graph of tagging information into two bipartite graphs and leveraged GNN to perform representation learning. GNN-PTR has achieved optimal performance on multiple real datasets. But in essence, GNN-PTR belongs to the recommendation models that operate in Euclidean space, so it still has limitations in accurately expressing ternary interactions in PTR systems.

C. HYPERBOLIC RECOMMENDATION MODELS

Most interactive data in the recommender systems actually possesses non-Euclidean properties such as power-law distribution, but the classical recommendation models, such as Bayesian Personalized Ranking (BPR) [28], Collaborative Metric Learning (CML) [29] and Neural Collaborative Filtering (NCF) [30] are designed in Euclidean space. These models may suffer from various degrees of embedding distortion. For this reason, some works [31], [32] make efforts to bridge the gap between hyperbolic space and recommender systems by modifying the matching functions of the recommendation models. The basic idea is to embed the representations of users and items into hyperbolic space, then use hyperbolic distance instead of the classical matching functions such as inner product or Euclidean distance to compute the semantic similarity between user and item. Vinh et al. studied the connection between metric learning in hyperbolic space and collaborative filtering. They devised a new method named HyperML [32] for one-class collaborative filtering. Hyperbolic metric embedding (HME) model [31] is designed for next-poi recommendation. HME jointly captures sequential transition, user preference, category, and region information in a unified approach by learning embeddings in a shared hyperbolic space. Subsequently, several models [33], [34], [35] enhanced by hyperbolic embedding have been proposed to better perform in traditional recommendation tasks or cope with new tasks. For example, Sun et al. [35] proposed HGCF to capture higher-order information in user-item interactions by incorporating multiple levels of neighborhood aggregation through a hyperbolic GCN module. To exploit mutual semantic relationships among users/items for collaborative filtering tasks, Li et al. [33] introduced a neighbor construction strategy to build user and item semantic neighborhoods and developed a deep framework with hyperbolic geometry to integrate constructed neighborhoods into the recommendation. Regarding the PTR tasks, Zhao et al. [36] proposed HPTR to learn the tagging information in hyperbolic space and utilize hyperbolic distance to model the entities' interactions.

HGCF and HPTR are the most relevant works for our model, the difference is that HGCF is suitable for the item recommendation task of binary interaction, and GHPTR is applicable to the PTR of ternary interaction. Moreover, HGCF is optimized by Riemann stochastic gradient descent, but GHPTR will adopt the tangent space optimization. Although HPTR is a PTR model based on hyperbolic embedding, it is only a shallow model without considering the higher-order semantic relevance, and the GHPTR makes up for this deficiency.

III. PRELIMINARIES

A. PROBLEM DEFINITION

The PTR system is different from item recommendation systems as it comprises three types of entities: users U, items I, and tags T. The historical interactions between these entities are represented as S which is a subset of $U \times I \times T$. An element $(u, i, t) \in S$ indicates that the user u has annotated the item i with the tag t. From the ternary relation set S, the PTR models usually deduce a three-order tensor $Y \in \mathbb{R}^{|U|\times|I|\times|T|}$, whose element $y_{u,i,t}$ is defined as follows:

$$y_{u,i,t} = \begin{cases} 1, & (u, i, t) \in S \\ 0, & otherwise, \end{cases} \tag{1}$$

where $y_{u,i,t} = 1$ indicates a positive instance, and the remaining data are the mixture of negative instances and missing values. In addition, the tagging information for a certain user-item pair (u, i) is defined as $\mathbf{y}_{u,i} = \{y_{u,i,t} | y_{u,i,t}, t \in T\}$.

PTR aims to recommend a ranked list of tags to a certain user for annotating his target item. Usually, a matching function $\widehat{Y} : U \times I \times T \longrightarrow \mathbb{R}$ is employed to measure and predict users' preferences on tags $w.r.t$ their target items. The entry $\widehat{y}_{u,i,t}$ of $\widehat{Y}$ indicates the degree to which a user u prefers to annotate the item i with the tag t. After predicting the score $\widehat{y}_{u,i,t}$ of all candidate tag t for a given user-item pair (u, i), the PTR system generates a ranked list of Top-N tags according to the obtained scores. Formally, the ranked list of Top-N tags given to the user-item pair (u, i) is defined as follows:

$$Top(u, i, N) = \underset{t \in T}{\overset{N}{argmax}}\ \widehat{y}_{u,i,t}, \tag{2}$$

where N denotes the number of recommended tags.

B. HYPERBOLIC SPACE

Hyperbolic space is a smooth Riemannian manifold with constant negative curvature. Due to the exponential expansion rate of the volume, hyperbolic space is well-suited for embedding tree-like data that follows the power-law distribution. Since hyperbolic space is difficult to exhibit intuitively, it is always described by five isometric models [37], i.e., Lorentz (hyperboloid) model, Poincaré ball model, Poincaré half space model, Klein model, and hemisphere model, of which the Poincaré ball and the Lorentz are commonly used in representation learning tasks. Let $\mathcal{B}^d = \left\{ \boldsymbol{x} \in \mathbb{R}^d \mid \|\boldsymbol{x}\| < 1 \right\}$ be the an open d-dimensional unit ball, where $\| \cdot \|$ denotes the Euclidean norm. The Poincaré ball can be defined by the Riemannian manifold$\left(\mathcal{B}^d, g_{\boldsymbol{x}}^B\right)$, where $g_{\boldsymbol{x}}^B = \left(\frac{2}{1-\|\boldsymbol{x}\|^2}\right)^2 g^E$ is the Riemannian metric tensor, in which $\boldsymbol{x} \in \mathcal{B}^d$ and $g^{\mathbb{E}} = \mathbf{I}$ denotes the Euclidean metric tensor. The distance between points $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{B}^d$ is given as:

$$d_B(\boldsymbol{x}, \boldsymbol{y}) = \operatorname{arcosh}\left(1 + 2\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{\left(1 - \|\boldsymbol{x}\|^2\right)\left(1 - \|\boldsymbol{y}\|^2\right)}\right) \tag{3}$$

Lorentz model, the so-called hyperboloid model, can be defined as Riemannian manifold $\left(\mathcal{L}^d, g_{\boldsymbol{x}}^{\mathcal{L}}\right)$, where $\mathcal{L}^d = \left\{ \boldsymbol{x} \in \mathbb{R}^{d+1} : \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathcal{L}} = -1, x_0 > 0 \right\}$, in which $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^{d} x_d y_d$ denotes the Lorentzian scalar product, and where $g_{\boldsymbol{x}}^{\mathcal{L}} = \operatorname{diag}([-1, 1, \ldots, 1])$. Based on above definitions, the distance between two points on Lorentz is given as:

$$d_{\mathcal{L}}(\boldsymbol{x}, \boldsymbol{y}) = \operatorname{arcosh}\left(-\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}}\right) \tag{4}$$

IV. THE PROPOSED MODEL

In this section, we first elaborate the overall framework of our proposed model, followed by presenting each component in detail. Finally, we introduce the learning process of model parameters.

The overall framework of our proposed model is illustrated in Figure 1. The model consists of three layers: embedding layer, propagation layer and prediction layer. The function of the embedding layer is to get the initial representations of the nodes based on their IDs; The propagation layer is responsible for aggregating the neighbors' representations by message propagation, so as to enrich the semantics of nodes' representations; After combining the higher-order representation of each entity, the prediction layer projects the combined representation to the hyperbolic space through the exponential mapping, and then matches the entity on the basis of the hyperbolic distance. Finally, the model predicts

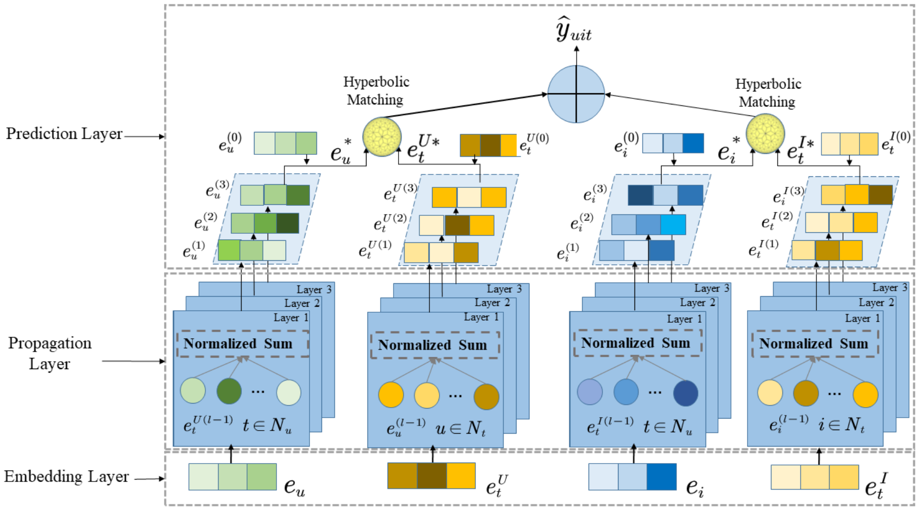


FIGURE 1. The framework of proposed model.

the user's preferred tags on the target item according to the matching degree (score).

A. EMBEDDING LAYER

In the embedding layer, we project all involved entities into a low-dimensional latent space according to their IDs. It should be noted that, to facilitate the optimization of the proposed model, the latent space here is not a hyperbolic space, but a tangent space of a hyperbolic space, which has the same flat property as a Euclidean space. The specific reasons are explained in Section IV-D. Specifically, a training instance of our model is a quadruple (u, i, t, t') where u denotes a user and i denotes an item. t corresponds to the positive tag, which had been assigned to the item i by the user u, and t' represents the negative tag which had not interacted with u and i. First, we perform a lookup operation in the corresponding embedding matrices according to the entity's IDs, then obtain the embedding of user u, item i, positive tag t, and negative tag t'. Formally,

$$\begin{aligned} e_u &= \mathbf{U}.\ \text{onehot}\ (u), e_i = \mathbf{I}.\ \text{onehot}\ (i), \\ e_t^U &= \mathbf{T}^U.\ \text{onehot}\ (t), e_{t'}^U = \mathbf{T}^U.\ \text{onehot}\ \left(t'\right), \\ e_t^I &= \mathbf{T}^I.\ \text{onehot}\ (t), e_{t'}^I = \mathbf{T}^I.\ \text{onehot}\ \left(t'\right) \end{aligned} \tag{5}$$

where $onehot(.)$ denotes the one-hot encoding operation. $\mathbf{U} \in \mathbb{R}^{|U|\times d}, \mathbf{I} \in \mathbb{R}^{|I|\times d}, \mathbf{T}^U \in \mathbb{R}^{|T|\times d}, \mathbf{T}^I \in \mathbb{R}^{|T|\times d}$($d$ is the embedding dimension) are the matrices of user embeddings, item embeddings, user-specific tag embeddings, and item-specific tag embeddings, respectively.

B. PROPAGATION LAYERS

As the core component of GHPTR, the propagation layer focuses on capturing the semantic relevance from higher-order connected path in historical interactions. Due to semantic relevance lurks in bidirectional interactions between entities, the propagation layer leverages GNN to explicitly capture such relevance.

Since the tripartite graph of historical interactions contains multiple relations among nodes, how to deal with these relations in the propagation process will directly affect the learning results of the proposed model. Inspired by the work [4], which demonstrates that only the user-tag and item-tag relation are deterministic for modeling a user's tagging preference, we consider these two relations and decompose the tripartite graph into two corresponding bipartite graphs. For each type of relation, two kinds of messages are propagated along the corresponding bipartite graph. In the case of the user-tag graph, the propagated messages include information from tag nodes to user nodes and from user nodes to tag nodes, and so it is with the item-tag graph. Based on the above settings, we can introduce a GNN framework to the propagation layer. Considering that the propagation layer needs to perform high-order information aggregation on two bipartite graphs, and the parameters to be learned involve four embedding matrices, both of which will increase the computational consumption, thus we adopt a lightweight GNN framework as that in [22] to explicitly inject the high-order relevance into nodes' embeddings by aggregating their neighbors' embeddings. Formally, embeddings of a triple (u, i, t) obtained at l-th propagating layer are formulated as:

$$\begin{aligned} e_u^{(l)} &= \sum_{t\in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{\left|N_t^U\right|}} e_t^{U(l-1)} \\ e_i^{(l)} &= \sum_{t\in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{\left|N_t^I\right|}} e_t^{I(l-1)} \\ e_t^{U(l)} &= \sum_{u\in N_t^U} \frac{1}{\sqrt{|N_u|}\sqrt{\left|N_t^U\right|}} e_u^{(l-1)} \\ e_t^{I(l)} &= \sum_{i\in N_t^I} \frac{1}{\sqrt{|N_i|}\sqrt{\left|N_t^I\right|}} e_i^{(l-1)} \end{aligned} \tag{6}$$

where $e_u^{(l-1)}$, $e_i^{(l-1)}$,$e_t^{U(l-1)}$ and $e_t^{I(l-1)}$ denote the embeddings of the user, item, user-specific tag and item-specific tag obtained from the previous propagation layer, N_u denotes the set of tags that interacted with the user u and refer to this set as the user's neighborhood. Similarly, the neighborhoods of the item, user-specific tag, and item-specific tag are denoted as N_i, N_t^U, and N_t^I, respectively. $\frac{1}{\sqrt{|\cdot|}\sqrt{|\cdot|}}$ are symmetric square root normalization terms for avoiding amplification of the embedding scale caused by the graph convolution operation.

C. PREDICTING LAYER

The task of the predicting layer is to embed the nodes' representations encoded with higher-order relevance in hyperbolic space and model nodes' interactions via hyperbolic distance, and finally output the predicted score through a matching function. The specific process is as follows:

By stacking multiple propagation layers, we obtain the embedding sets of each entity. Every element in the set represents the semantic relevance of different-order neighbors, it is conducive to characterizing different properties of an entity, so we combine all corresponding elements into a single embedding. Formally,

$$\begin{aligned} e_u^* &= \alpha_1 e_u^{(1)} + \alpha_2 e_u^{(2)} + \cdots + \alpha_{l-1} e_u^{(l-1)} + \alpha_l e_u^{(l)} \\ e_t^{U*} &= \alpha_1 e_t^{U(1)} + \alpha_2 e_t^{U(2)} + \cdots + \alpha_{l-1} e_t^{U(l-1)} + \alpha_l e_t^{U(l)} \\ e_i^* &= \alpha_1 e_i^{(1)} + \alpha_2 e_i^{(2)} + \cdots + \alpha_{l-1} e_i^{(l-1)} + \alpha_l e_i^{(l)} \\ e_t^{I*} &= \alpha_1 e_t^{I(1)} + \alpha_2 e_t^{I(2)} + \cdots + \alpha_{l-1} e_t^{I(l-1)} + \alpha_l e_t^{I(l)} \end{aligned} \tag{7}$$

where a_l denotes the weight of a embedding in the l-th layer. In order to simplify our model, we empirically set the a_l to $\frac{1}{(L+1)}$, where L is the total number of propagation layers.

Based on the obtained higher-order representations, we define a matching function with hyperbolic distance for the final prediction. Given a triplet (u, i, t), the matching function $\widehat{y}_{u,i,t}$ can be defined as:

$$\widehat{y}_{u,i,t} = p\left(d_H(e_u^*, e_t^{U*}) + d_H(e_i^*, e_t^{I*})\right) \tag{8}$$

where $d_H(\cdot)$ denotes the hyperbolic distance function, $p(\cdot)$ is the transformation function for converting hyperbolic distances to the matching degree, here we take it as $p(x) = \beta x + c$ with $\beta \in \mathbb{R}$ and $c \in \mathbb{R}$.

Note that in order to adequately examine the influence of hyperbolic embedding on the performance of our proposed GHPTR model, we take Poincaré Ball and Lorentz as the geometric representation of hyperbolic space, and obtain two versions of the proposed model, denoted as GHPTR(P) and GHPTR(L). Thus the hyperbolic distance $d_H(\cdot)$ in this paper will be computed according to Equation 3 and Equation 4, respectively.

D. MODEL TRAINING

The strategy of training set construction is inspired by the work [4]: When observing a certain pair (u, i) in historical interactions S, it can be inferred that the user u should prefer tag t over tag t' iff the triple (u, i, t) can be observed from S and (u, i, t') can not be observed. Based on this idea, the training set D_S (i.e., the set of quadruple (u, i, t, t')) with the pairwise constraint is defined as:

$$D_S = \left\{(u, i, t, t') \mid (u, i, t) \in S \wedge (u, i, t') \notin S\right\} \tag{9}$$

The objective of model training is to maximize the gap between the matching scores $\widehat{y}_{u,i,t}$ of the positive triple (u, i, t) and negative triple (u, i, t'), so we adopt the Bayesian Personalized Ranking (BPR) optimization criterion [28] to learn model parameters $\Theta = \{\mathbf{U}, \mathbf{I}, \mathbf{T^U}, \mathbf{T^I}, \beta, c\}$, and build the objective function of proposed model as follows:

$$\mathcal{L} = \min_{\Theta} \sum_{(u,i,t,t')\in D_s} - \ln \sigma\left(\widehat{y}_{u,i,t} - \hat{y}_{u,i,t'}\right) + \lambda_\Theta \|\Theta\|_F^2 \tag{10}$$

As the Poincaré ball and Lorentz are both Riemannian manifolds with constant negative curvature, their related parameters need to be updated by Riemannian gradient, so the Riemannian stochastic gradient descent(RSGD) [38] has been widely adopted to optimize most of Hyperbolic embedding-based models [14], [31]. However, RSGD is challenging in practice. Concerning our model, its parameters consist of $\{\mathbf{U}, \mathbf{I}, \mathbf{T}\}$ that require to be projected into hyperbolic space and $\{\beta, c\}$ that with no requirement for projection. To avoid using two corresponding optimizers, we update all the parameters via tangent space optimization [16], [17].

We recall that a d-dimensional hyperbolic space is a Riemannian manifold $\mathcal{M}$ with a constant negative curvature $-c(c > 0)$, the tangent space $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ at point $\boldsymbol{x}$ on $\mathcal{M}$ is a d-dimensional flat space that best approximates $\mathcal{M}$ around $\boldsymbol{x}$, and the elements $\mathbf{v}$ of $\mathcal{T}_{\mathrm{x}}\mathcal{M}$ are referred to as tangent vectors. In our work, We define all the parameters in the tangent space so that we can update them via powerful Euclidean optimizers(e.g., Adam). When it comes to calculating the hyperbolic distance d_H, we use the exponential map $\exp_{\boldsymbol{x}}^{H^d}(\mathbf{v})$ to recover the corresponding parameters (project $\mathbf{v}$ of tangent space back to hyperbolic space). The exponential map related to the Poincaré ball is formulated as follows:

$$\exp_{\boldsymbol{x}}^{\mathcal{B}^d}(\mathbf{v}) = \boldsymbol{x} \oplus \left(\tanh\left(\frac{\lambda_{\boldsymbol{x}}\|\mathbf{v}\|}{2}\right)\frac{\mathbf{v}}{\|\mathbf{v}\|}\right) \tag{11}$$

where $\oplus$ denotes the *Möbius* addition operator [39] that provides an analogue to Euclidean addition for hyperbolic space. Formally,

$$\boldsymbol{x} \oplus \boldsymbol{y} := \frac{\left(1 + 2\langle \boldsymbol{x}, \boldsymbol{y}\rangle + \|\boldsymbol{y}\|^2\right)\boldsymbol{x} + \left(1 - \|\boldsymbol{x}\|^2\right)\boldsymbol{y}}{1 + 2\langle \boldsymbol{x}, \boldsymbol{y}\rangle + \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2} \tag{12}$$

The corresponding exponential map of Lorentz is given as:

$$\exp_{\boldsymbol{x}}^{\mathcal{L}^d}(\mathbf{v}) = \cosh\left(\|\mathbf{v}\|\right)\boldsymbol{x} + \sinh\left(\|\mathbf{v}\|\right)\frac{\mathbf{v}}{\|\mathbf{v}\|} \tag{13}$$

V. EXPERIMENTS AND ANALYSIS

In this section, we first set up the experiments, and then present the performance comparison and result analysis.

A. EXPERIMENTAL SETUP

In our experiments, we choose two public datasets, i.e., LastFM and ML10M, to evaluate the performance of all compared methods. Similar to [4], we preprocess each dataset to obtain their corresponding p-core, which is the largest subset where each user, item, and tag has to occur at least p times. In our experiments, every datasets is the result of 5-core or 10-core preprocessing. The general statistics of datasets are summarized in TABLE 1.

TABLE 1. Description of datasets.

Dataset	Users	Items	Tags	Tag assignments
LastFM-core5	1348	6927	2132	162047
LastFM-core10	966	3870	1024	133945
ML10M-core5	990	3247	2566	61688
ML10M-core10	469	1524	1017	37414

We adopt the leave-one-out protocol to evaluate the recommendation performance of all compared methods. Specifically, for each pair (u, i), we select the last triple (u, i, t) according to the timestamp and transfer it from S to S_{test}. The remaining observed triples constitute the training set $S_{train} = S - S_{test}$. Similar to the item recommendation problem, the PTR provides a top- N ranked list of tags for a given pair (u, i), so we employ two typical ranking metrics to measure the performance of all compared methods, i.e., $Precision@N$ and $Recall@N$. Formally,

$$Precision@N = \frac{1}{|P_{S_{\text{test}}}|} \sum_{(u,i)\in P_{\text{test}}} \frac{|\ \text{Top}\,(u, i, t) \cap (u, i, t) \in S_{\text{test}}\ |}{N}$$

$$Recall@N = \frac{1}{|P_{S_{\text{test}}}|} \sum_{(u,i)\in P_{\text{test}}} \frac{|\ \text{Top}\,(u, i, N) \cap (u, i, t) \in S_{\text{test}}\ |}{|\{t \mid (u, i, t) \in S_{\text{test}}\}|} \tag{14}$$

For both metrics, we set $N = 3, 5, 10$ in the experiments.

B. BASELINES AND PARAMETER SETTINGS

In order to evaluate the effectiveness of our proposed model, we choose the following personalized tag recommendation models as baselines:

- PITF: PITF [4] explicitly models the pairwise interactions among users, items and tags by inner product, it is a strong competitor in the field of personalized tag recommendation.
- NLTF: NLTF [3] is a non-linear tensor factorization model, which enhances PITF by exploiting the Gaussian radial basis function to capture the nonlinear interactive relations among users, items and tags.
- ABNT: ABNT [6] utilizes the multi-layer perception to model nonlinear interactions between users, items, and tags, and employs attention networks to capture complex patterns of users' behaviors.
- DAE-PTR: DAE-PTR [42] utilizes the learning framework of denoising auto-encoder to enhance the robustness of features learned from historical tagging information.
- HPTR: HPTR [36] learns the representations of entities by modeling their interactive relationships in hyperbolic space and utilizes hyperbolic distance to measure semantic relevance between entities.
- GNN-PTR: GNN-PTR [8] is a graph-neural-networks enhanced tag recommendation model, which introduces GNN to the pairwise interaction tensor factorization framework for mining high-order similarity between embeddings.

We empirically set the parameters of baselines according to their corresponding literatures in order to recover their optimal performance: the dimension of embedding d is set to 64, and the learning rate η is tuned amongst $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\}$. For the ABNT model, the number of hidden layers is set to 2. All the parameters of HPTR and GHPTR are defined in tangent space $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}$ located at origin point ($\boldsymbol{x} = 0$) of hyperbolic space. The number of propagation layers is set to 3 for both GNN-PTR and GHPTR. We adopt Adam [40] as the optimizer for all involved models.

C. PERFORMANCE COMPARISION

The experimental results of all comparison models on each dataset are presented in the following tables.

From TABLE 2 to TABLE 5, we have the following observations:

(1) Among the baselines not equipped with hyperbolic embeddings, the GNN-PTR is superior to other models for all evaluation metrics, which indicates that the neighborhood aggregation implemented by message propagation mechanisms is efficient for enhancing tag recommendation. The reason for the poorer performance of the rest may be that they learn shallow representations from low-order interactions. Thus, the learned representations lack the semantics for revealing the user's tagging preference.

(2) Although HPTR is a shallow representation learning model, it outperforms the GNN-PTR in most cases of our experiments, and the reason for this may be that HPTR desires to better express the global structural properties (e.g., scale-free or power-law) of the interactive tripartite, graph so it leverages hyperbolic embedding to alleviate the distortion problem. GNN-PTR focuses on the local properties of the graph. Therefore, it utilizes GNN to capture high-order relevance within the neighborhood. This result implies that using global properties is more effective than local properties in improving the performance of tag recommendations.

TABLE 2. LastFM-5core.

Method	$Precision@3$	$Precision@5$	$Precision@10$	$Recall@3$	$Recall@5$	$Recall@10$
ABNT	0.1563	0.1353	0.1018	0.1569	0.2194	0.3298
NLTF	0.1949	0.1678	0.1191	0.2275	0.3239	0.4523
PITF	0.2127	0.1789	0.1274	0.2571	0.3479	0.4814
DAE-PTR	0.2243	0.1858	0.1311	0.3165	0.3843	0.4928
HPTR	0.2612	0.2229	0.1424	0.3597	0.4383	0.5191
GNN-PTR	0.2324	0.1913	0.1327	0.3244	0.4169	0.5454
GHPTR(P)	0.3030	**0.2398**	0.1546	0.3868	0.4770	0.5642
GHPTR(L)	**0.3043**	0.2390	**0.1547**	**0.3914**	**0.4776**	**0.5673**

TABLE 3. LastFM-10core.

Method	$Precision@3$	$Precision@5$	$Precision@10$	$Recall@3$	$Recall@5$	$Recall@10$
ABNT	0.2041	0.1767	0.1342	0.1981	0.2592	0.3534
NLTF	0.2544	0.2163	0.1351	0.2945	0.4118	0.5142
PITF	0.2515	0.2087	0.1458	0.3204	0.4158	0.5654
DAE-PTR	0.2597	0.2176	0.1425	0.3349	0.4312	0.5344
HPTR	0.2861	0.2255	0.1574	0.3501	0.4714	0.5771
GNN-PTR	0.2646	0.2142	0.1461	0.3476	0.4529	0.5874
GHPTR(P)	**0.3406**	0.2657	0.1682	0.4311	0.5252	**0.6170**
GHPTR(L)	0.3382	**0.2658**	**0.1772**	**0.4339**	**0.5267**	0.6119

TABLE 4. ML10M-5core.

Method	$Precision@3$	$Precision@5$	$Precision@10$	$Recall@3$	$Recall@5$	$Recall@10$
ABNT	0.1022	0.0829	0.0413	0.2391	0.2938	0.3444
NLTF	0.1323	0.0972	0.0597	0.2974	0.3561	0.4312
PITF	0.1497	0.1021	0.0641	0.3208	0.3909	0.4623
DAE-PTR	0.1522	0.0982	0.0618	0.3294	0.3941	0.4765
HPTR	0.1611	0.1106	0.0707	0.3616	0.4156	0.4766
GNN-PTR	0.1524	0.1055	0.0672	0.3331	0.3965	0.4851
GHPTR(P)	0.1904	0.1358	0.0788	0.4112	0.4703	0.5325
GHPTR(L)	**0.1915**	**0.1375**	**0.0797**	**0.4158**	**0.4799**	**0.5391**

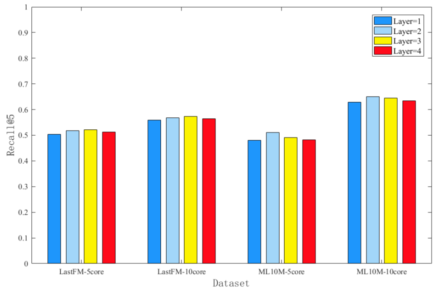
TABLE 5. ML10M-10core.

Method	$Precision@3$	$Precision@5$	$Precision@10$	$Recall@3$	$Recall@5$	$Recall@10$
ABNT	0.1183	0.0959	0.0601	0.2610	0.3714	0.4572
NLTF	0.1635	0.1142	0.0729	0.3388	0.4334	0.5340
PITF	0.1798	0.1272	0.0744	0.3770	0.4523	0.5205
DAE-PTR	0.1863	0.1315	0.0803	0.4063	0.4934	0.5437
HPTR	0.2189	0.1483	0.0825	0.4969	0.5485	0.5960
GNN-PTR	0.1933	0.1390	0.0842	0.4602	0.5460	0.6398
GHPTR(P)	0.2507	0.1706	0.0957	0.5683	0.6294	0.6884
GHPTR(L)	**0.2519**	**0.1726**	**0.0960**	**0.5713**	**0.6343**	**0.6898**

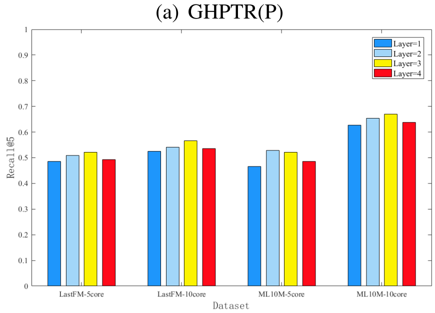
(3) GHPTR shows the best performance overall involved baselines. It surpasses Precision@10 of the best baselines by 8.6%, 12.5%, 12.7%, and 14.1% on Lastfm-core5, Lastfm-core10, ML10M-core5, and ML10M-core10, respectively. With respect to Recall@10, the improvements of GHPTR over best baselines are 5.7%, 5.0%, 11.1%, and 8.8% on the above four datasets. The main reason should be that we integrated GNN and hyperbolic geometry into the learning framework of personalized tag recommendation. In this way, the learned representations are endowed with the global and local structural properties of the raw data so that the proposed model is challenging to fall into sub-optimal learning, resulting in the enhancement of recommendation performance.

D. EFFECT OF PROPAGATION LAYERS

For the GHPTR model, the number of message propagation layers l is another important hyper-parameter, which controls the range of capturing the semantic relevance in the higher-order connected paths. In order to analyze the impact of l on the recommendation quality of our model, we conduct a set of extended experiments in this section. With Recall@5 as the indicator, we keep the same settings described in Section V-A and adjust the value of l in steps of 1 until l=4,



(a) GHPTR(P)



(b) GHPTR(L)

FIGURE 2. Effect of the parameter *l* for GHPTR.

reporting the result of Recall@5 obtained by the model for each l.

Figure 2 exhibits the performance of GHPTR under different l values on each dataset. As shown in the figure, for both model versions, their recommendation performance improves as the number of message propagation layers increases. When the number of propagation layers reaches 3 or 4, the performances of GHPTR on most datasets begin to decrease. The reason is that excessive stacking of propagation layers will introduce the semantic information of long-range neighbors into the representation of the target node, and the semantic relevance between these neighbors and the target is relatively weak. Therefore they become useless or even noisy information that will be finally encoded into the representations, decreasing the performance of our proposed model.

E. EFFECT OF EMBEDDING DIMENSION

In our proposed model, the dimension of embeddings d is an essential parameter since it controls the expressive ability of the whole model, so we conduct additional experiments to study the sensitivity of d to the performance of our model by tuning it within {8, 16, 32, 64, 128, 256, 512, 1024}. Here we choose Precision@5 to give an insight into the impact on performance with respect to the parameter d, and the experimental results are plotted in FIGURE 3. From the content of the figure, we can have the following observations and findings:

(1) In the beginning, the values of Precision@5 increase stably with the growth of d. When d exceeds 128, most of the curves are no longer in an uptrend, which indicates that merely increasing the dimension is not conducive to sustained improvement of recommendation. The main reason may be similar to the Euclidean embedding: the model will obtain sufficient learning ability when d reaches a certain threshold. After that, continuously increasing the embedding dimension can also lead to overfitting problems.

(2) Compared with GHPTR (L), the curve of GHPTR (P) exhibits less smooth, such observation is consistent with previous studies [14], [17], The reason lies in the Equation 3 of Poincaré ball distance, i.e., $d_{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{y}) = \operatorname{arcosh}\left(1 + 2\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{(1-\|\boldsymbol{x}\|^2)(1-\|\boldsymbol{y}\|^2)}\right)$, When the norm of $\boldsymbol{x}$ or $\boldsymbol{y}$ approaches 1, that is, when the embeddings are closer to the edge of the ball, the denominator of the equation rapidly approaches 0, resulting in instability of the calculation results.

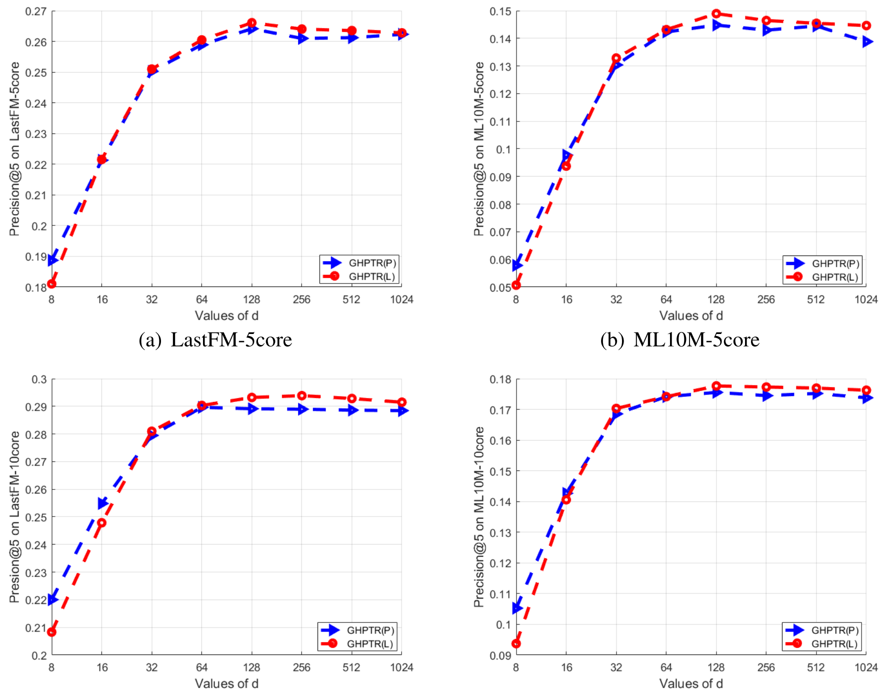
F. ABLATION STUDY

The learning framework of our GHPTR contains two components: a lightweight GNN workflow and a hyperbolic matching process. To study the rationality of these two components, we remove them from the proposed model and obtain two corresponding variants, denoted as GHPTR-H and GHPTR-G. We conducted an extended set of experiments to observe the performance of GHPTR and its variants on ML10M-10core and LastFM-10core, taking Precision@10 as the evaluation metric and setting all involved hyper-parameters the same as GHPTR in Section V-A. In addition, considering the relative stability of the Lorentz model, we choose it as the geometric representation of the hyperbolic space in this section. The experimental results on different embedding dimension d ranging from 16 to 256 are plotted in FIGURE 4.

As shown in FIGURE 4, we can get the following observations and inferences:

(1) The performance curves of the variants are all lower than that of the original model, indicating that each component of the GHPTR significantly affects recommendation quality. On the other hand, the performance of GHPTR-H is inferior to that of GHPTR-G, revealing that hyperbolic embedding contributes more to recommendation performance than GNN. More importantly, this result suggests that we should give priority to catch the global properties of interactive data when designing learning framework of PTR models.

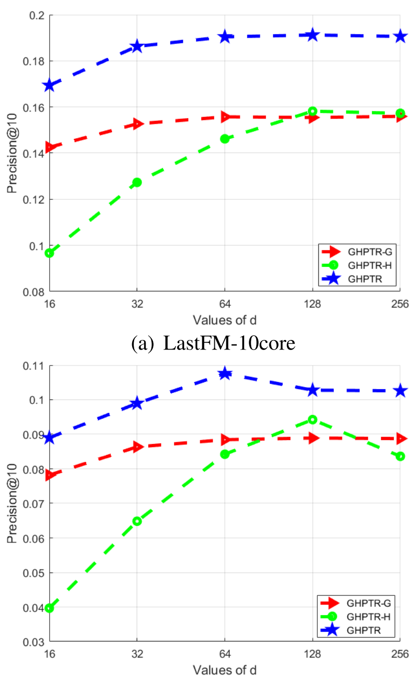
(2) Both GHPTR and GHPTR-G outperform GHPTR-H at lower embedding dimensions. With the increase of embedding dimension, the performance improvement of these two models is not as significant as that of GHPTR-H. This observation is consistent with studies [16], [17], which indicates that the advantage of hyperbolic spaces is reflected in the lower embedding dimensions because its exponential expansion property can endow the embedded model with considerable expressiveness in the lower dimension. In contrast, Euclidean space requires larger embedding dimensions to obtain sufficient learning ability. Furthermore, when embedding dimensions reach a certain threshold, they all will fall into overfitting problems.



(a) LastFM-5core (b) ML10M-5core

(c) LastFM-10core (d) ML10M-10core

FIGURE 3. Effect of the parameter *d* for GHPTR.



(a) LastFM-10core

(b) ML10M-10core

FIGURE 4. Ablation study of GHPTR.

VI. CONCLUSION

Existing hyperbolic embedding-based tag recommendation models only account for the macro properties of the data, overlooking the node-level properties. In comparison, GNN-based tag recommendation models are competent for exploiting the properties of nodes and their neighborhoods. In this work, in order to learn both global and local properties of historical interactions, we present a lightweight yet effective personalized tag recommendation model based on the integration of hyperbolic embedding and GNN. Through extensive experiments on two datasets, we are able to demonstrate the effectiveness of GHPTR over other baselines.

Although hyperbolic embedding is adept at representing tree-like data, we should not neglect the advantages of Euclidean space. Compared with hyperbolic space, the vector operators of Euclidean space is more efficient, and the relative distance between points can be better distinguished via Euclidean metrics. Considering the advantages of hyperbolic and Euclidean spaces, our future work will construct contrasting views from these spaces and carry out graph contrastive learning [41] to obtain more semantics for promoting personalized tag recommendations.

REFERENCES

[1] F. M. Belém, J. M. Almeida, and M. A. Gonçalves, "A survey on tag recommendation methods," *J. Assoc. Inf. Sci. Technol.*, vol. 68, no. 4, pp. 830–844, Apr. 2017.

[2] Y. Cai, M. Zhang, D. Luo, C. Ding, and S. Chakravarthy, "Low-order tensor decompositions for social tagging recommendation," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, Feb. 2011, pp. 695–704.

[3] X. Fang, R. Pan, G. Cao, X. He, and W. Dai, "Personalized tag recommendation through nonlinear tensor factorization using Gaussian kernel," in *Proc. AAAI*, 2015, pp. 439–445.

[4] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, Feb. 2010, pp. 81–90.

[5] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Tag recommendations based on tensor dimensionality reduction," in *Proc. ACM Conf. Recommender Syst.*, Oct. 2008, pp. 43–50.

[6] J. Yuan, Y. Jin, W. Liu, and X. Wang, "Attention-based neural tag recommendation," in *Proc. DASFAA*. Berlin, Germany: Springer, 2019, pp. 350–365.

[7] Z.-K. Zhang and C. Liu, "A hypergraph model of social tagging networks," *J. Stat. Mech., Theory Exp.*, vol. 2010, no. 10, Oct. 2010, Art. no. P10005.

[8] X. Chen, Y. Yu, F. Jiang, L. Zhang, R. Gao, and H. Gao, "Graph neural networks boosted personalized tag recommendation algorithm," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[9] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*. Red Hook, NY, USA: Curran Associates, 2017, pp. 1025–1035.

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

[11] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2008.

[12] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá, "Hyperbolic geometry of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 82, no. 3, Sep. 2010, Art. no. 036106.

[13] H. Halpin, V. Robu, and H. Shepherd, "The complex dynamics of collaborative tagging," in *Proc. 16th Int. Conf. World Wide Web*, May 2007, pp. 211–220.

[14] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proc. NIPS*. Red Hook, NY, USA: Curran Associates, 2017, pp. 6341–6350.

[15] M. Nickel and D. Kiela, "Learning continuous hierarchies in the Lorentz model of hyperbolic geometry," in *Proc. ICML*, 2018, pp. 3779–3788.

[16] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in *Proc. ACL*, 2020, pp. 6901–6914.

[17] I. Chami, R. Ying, C. Re, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Proc. NIPS*. Red Hook, NY, USA: Curran Associates, 2019, pp. 4868–4879.

[18] H. T. Nguyen, M. Wistuba, J. Grabocka, L. R. Drumond, and L. Schmidt-Thieme, "Personalized deep learning for tag recommendation," in *Proc. PAKDD*. Berlin, Germany: Springer, 2017, pp. 186–197.

[19] H. T. Nguyen, M. Wistuba, and L. Schmidt-Thieme, "Personalized tag recommendation for images using deep transfer learning," in *Proc. ECML-PKDD*. Berlin, Germany: Springer, 2017, pp. 705–720.

[20] S. Wang, L. Hu, Y. Wang, X. He, Q. Z. Sheng, M. A. Orgun, L. Cao, F. Ricci, and P. S. Yu, "Graph learning based recommender systems: A review," in *Proc. IJCAI*, 2021, pp. 4644–4652.

[21] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. SIGIR*, 2019, pp. 165–174.

[22] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. SIGIR*, 2020, pp. 639–648.

[23] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, May 2019, pp. 417–426.

[24] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *Proc. AAAI*, 2021, pp. 4503–4511.

[25] Y. Yu, W. Qian, L. Zhang, and R. Gao, "A graph-neural-network-based social network recommendation algorithm using high-order neighbor information," *Sensors*, vol. 22, no. 19, p. 7122, Sep. 2022.

[26] J. Sun, Y. Zhang, W. Guo, H. Guo, R. Tang, X. He, C. Ma, and M. Coates, "Neighbor interaction aware graph convolution networks for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1289–1298.

[27] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proc. AAAI*, 2019, pp. 346–353.

[28] H. Zhao and X. Wang, "Bi-group Bayesian personalized ranking from implicit feedback," in *Proc. 2nd Int. Conf. Comput. Sci. Softw. Eng.*, May 2019, pp. 452–461.

[29] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 193–201.

[30] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. WWW*, 2017, pp. 173–182.

[31] S. Feng, L. V. Tran, G. Cong, L. Chen, J. Li, and F. Li, "HME: A hyperbolic metric embedding approach for next-poi recommendation," in *Proc. SIGIR*, 2020, pp. 1429–1438.

[32] L. Vinh Tran, Y. Tay, S. Zhang, G. Cong, and X. Li, "HyperML: A boosting metric learning approach in hyperbolic space for recommender systems," in *Proc. WSDM*, 2020, pp. 609–617.

[33] A. Li, B. Yang, H. Huo, H. Chen, G. Xu, and Z. Wang, "Hyperbolic neural collaborative recommender," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9114–9127, Sep. 2023.

[34] C. Ma, L. Ma, Y. Zhang, H. Wu, X. Liu, and M. Coates, "Knowledge-enhanced top-k recommendation in poincaré ball," in *Proc. AAAI*, 2021, pp. 4285–4293.

[35] J. Sun, Z. Cheng, S. Zuberi, F. Pérez, and M. Volkovs, "HGCF: Hyperbolic graph convolution networks for collaborative filtering," in *Proc. WWW*, 2021, pp. 593–601.

[36] W. Zhao, A. Zhang, L. Shang, Y. Yu, L. Zhang, C. Wang, J. Chen, and H. Yin, "Hyperbolic personalized tag recommendation," in *Proc. DASFAA*. Berlin, Germany: Springer, 2022, pp. 216–231.

[37] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao, "Hyperbolic deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10023–10044, Dec. 2022.

[38] S. Bonnabel, "Stochastic gradient descent on Riemannian manifolds," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2217–2229, Sep. 2013.

[39] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," in *Proc. NIPS*. Red Hook, NY, USA: Curran Associates, 2018, pp. 5350–5360.

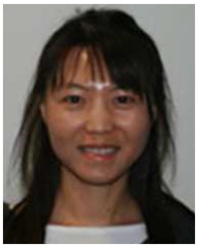
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[41] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Proc. NIPS*. Red Hook, NY, USA: Curran Associates, 2020, pp. 5812–5823.

[42] W. Zhao, L. Shang, Y. Yu, L. Zhang, C. Wang, and J. Chen, "Personalized tag recommendation via denoising auto-encoder," *World Wide Web*, vol. 26, no. 1, pp. 95–114, Jan. 2023.

CHUNMEI ZHANG received the M.Sc. degree from Nanjing Normal University, China. She is currently a Lecturer with the School of Artificial Intelligence, Nanjing Vocational College of Information Technology. Her main research interests include data mining and image processing.

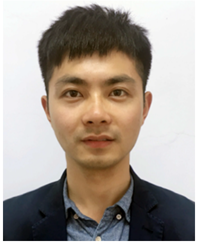
AORAN ZHANG is currently pursuing the M.Sc. degree with the School of Computer, Jiangsu University of Science and Technology, China. His main research interests include recommendation algorithms and machine learning.



LI ZHANG (Senior Member, IEEE) received the Ph.D. degree from the University of Birmingham, U.K. She is currently an Associate Professor and a Reader with the Department of Computer Science, Royal Holloway, University of London, U.K. She is also an Honorary Research Fellow with the University of Birmingham. Her research interests include artificial intelligence, machine learning, evolutionary computation, and deep learning. She has served as an Associate Editor for *Decision Support Systems*.

YONGHONG YU received the B.Sc. and M.Sc. degrees in computer science from Wuhan University, Wuhan, China, and the Ph.D. degree in computer application technology from Nanjing University. He is currently a Professor with the College of Tongda, Nanjing University of Posts and Telecommunications, China. His main research interests include machine learning, recommender systems, and image processing.

WEIBIN ZHAO received the Ph.D. degree in computer application technology from Nanjing University. He is currently an Associate Professor with the College of Tongda, Nanjing University of Posts and Telecommunications, China. His main research interests include recommender systems and natural language processing.



HAI GENG received the M.Sc. degree from Zhejiang Normal University. He is currently a Lecturer with the School of Tourism Management, Nanjing Institute of Tourism and Hospitality, China. His main research interests include machine learning and image processing.