RotDiff: A Hyperbolic Rotation Representation Model for Information Diffusion Prediction

Hongliang Qiao*
Harbin Institute of Technology
Shenzhen, China
21s151112@stu.hit.edu.cn

Shanshan Feng*
Wecar Technology Co., Ltd.
Shenzhen, China
victor_fengss@foxmail.com

Xutao Li†
Harbin Institute of Technology
Shenzhen, China
lixutao@hit.edu.cn

Huiwei Lin
Harbin Institute of Technology
Shenzhen, China
linhuiwei@stu.hit.edu.cn

Han Hu
Beijing Institute of Technology
Beijing, China
hhu@bit.edu.cn

Wei Wei
Huazhong University of Science and Technology
Wuhan, China
weiw@hust.edu.cn

Yunming Ye
Harbin Institute of Technology
Shenzhen, China
yeyunming@hit.edu.cn

ABSTRACT

The massive amounts of online user behavior data on social networks allow for the investigation of information diffusion prediction, which is essential to comprehend how information propagates among users. The main difficulty in diffusion prediction problem is to effectively model the complex social factors in social networks and diffusion cascades. However, existing methods are mainly based on Euclidean space, which cannot well preserve the underlying hierarchical structures that could better reflect the strength of user influence. Meanwhile, existing methods cannot accurately model the obvious asymmetric features of the diffusion process. To alleviate these limitations, we utilize rotation transformation in the hyperbolic to model complex diffusion patterns. The modulus of representations in the hyperbolic space could effectively describe the strength of the user's influence. Rotation transformations could represent a variety of complex asymmetric features. Further, rotation transformation could model various social factors without changing the strength of influence. In this paper, we propose a novel hyperbolic rotation representation model RotDiff for the diffusion prediction problem. Specifically, we first map each social user to a Lorentzian vector and use two groups of transformations to encode global social factors in the social graph and the diffusion graph. Then, we combine attention mechanism in the hyperbolic space with extra rotation transformations to capture local diffusion dependencies within a given cascade. Experimental results on five real-world datasets demonstrate that the proposed model RotDiff outperforms various state-of-the-art diffusion prediction models.

CCS CONCEPTS

• **Information systems → Social networks**; • **Computing methodologies → Neural networks**.

KEYWORDS

Social Networks, Diffusion Prediction, Hyperbolic Representation, Rotation Transformation

ACM Reference Format:
Hongliang Qiao, Shanshan Feng, Xutao Li, Huiwei Lin, Han Hu, Wei Wei, and Yunming Ye. 2023. RotDiff: A Hyperbolic Rotation Representation Model for Information Diffusion Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3615041

1 INTRODUCTION

Social media, such as Twitter and Sina Weibo, has a considerable impact on daily life. Massive information spreads rapidly on these platforms through social users. Interestingly, the behavior of social users is often influenced by their friends or some high-impact users. Indeed, the information diffusion process is affected by multiple social factors, such as social connections, diffusion dependencies, and potential hierarchies. We expect to infer complex social factors and identify which users are more likely to be affected in the future.

Information diffusion prediction problems have been widely investigated for decades. Traditional methods directly calculate the diffusion probabilities using predefined diffusion models [14, 20, 28], such as independent cascade (IC) or linear threshold (LT) models. Meanwhile, embedding-based methods [2, 3, 10, 38] embed diffusion dependencies of cascades into user embeddings and calculate diffusion probabilities through well-designed functions. Both classes of methods are constrained by restrictive assumptions and tedious feature extractions. Subsequently, due to the sequential nature of

*Both authors contributed equally to this research.
†The corresponding author.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615041

diffusion cascades, some researchers propose models [17, 34, 37] exploiting recurrent neural networks (RNNs) to predict the next-affected users. Despite progress has been achieved, such approaches focus merely on dependencies within historical diffusion sequences, while ignoring the impact of social relations. Recently, Graph Neural Networks (GNNs) demonstrate the powerful expressiveness of modeling graph data, enabling GNN-based methods [12, 18, 19, 31, 44] to achieve significant performance improvements.

However, existing methods still have limitations. There are two important social data components: the social graphs depict global social connections among users, while the historical diffusion cascades contain two types of relations. Firstly, all cascades imply global diffusion connections. Secondly, within each individual cascade, there are local diffusion dependencies among its users. Existing methods cannot effectively extract critical information from these two components. The underlying hierarchical structures in social data are typically ignored. Meanwhile, the existing methods model different data components separately, leading to the learned representation vectors cannot accurately measure the strength of user influence. Further, considering the directionality of influence propagation, the social factors in the social data have underlying asymmetric characteristics. Nevertheless, existing methods cannot accurately describe these asymmetric properties. All these deficiencies lead to the suboptimal performance of existing methods.

Hyperbolic representation learning and rotation transformation provide insights to alleviate existing limitations. We observed that hyperbolic representation methods [5, 25, 45] can effectively capture latent hierarchical structures. Due to exponential expansion property[22], embedding user nodes into a hyperbolic space enables a more precise characterization of the underlying distribution of influence. In this way, the strength of user influence could be more effectively distinguished by describing hierarchical features, which is challenging to achieve for most existing models based on Euclidean space. Consequently, we learn user embeddings in the hyperbolic space.

Besides, we found that rotation transformations could effectively characterize asymmetric properties of social factor. Rotation transformation could effectively represent a variety of relations through different angles. Another significant advantage of the rotation transformation is that it allows for the modeling of various complex relations without affecting the vector's length. Thus, rotation transformation could describe various social factors while preserving the strength of the users' influence. Inspired by this, we utilize rotation transformation to extract latent properties within the social graph and diffusion cascades.

Based on our findings, we address the diffusion prediction problem via rotation transformations in the hyperbolic space. We construct a diffusion graph from all historical diffusion cascades that captures global diffusion connections among users. We infer information from both the social graph and the diffusion graph to generate user embeddings in the hyperbolic space that could accurately measure the strength of each user's influence. Moreover, we represent the asymmetric social features on each graph with two sets of rotation matrices, respectively. We then use user embeddings combined with learned social features to model a given cascade and perform the prediction task.

Specifically, we propose a novel hyperbolic rotation representation model (RotDiff) for the diffusion prediction problem. RotDiff has three major components. First, the Lorentz rotation embedding trains user embeddings and captures social and diffusion connections through rotation transformations in the hyperbolic space. Second, the rotated Lorentz self-attention extracts diffusion dependencies within a cascade and constructs associated latent embeddings. Third, the prediction module determines the possibilities of the subsequent users who will be activated. Our main contributions are summarized as follows:

- We investigate reflecting the strength of user influence in the hyperbolic space. Through rotation transformations, we effectively model various complex asymmetric features in the information diffusion process while preserving the strength of influence. To the best of our knowledge, this is the first work to introduce rotation transformation into the information diffusion prediction problem.
- We develop a novel hyperbolic rotation diffusion representation model RotDiff for the information diffusion prediction problem. Rotdiff encodes global social factors into user embeddings and rotation matrices by joint modeling the social graph and the diffusion graph while further exploring local diffusion dependencies for a specific cascade.
- We conduct extensive experiments on five real-world datasets, demonstrating that the proposed model significantly outperforms various state-of-the-art diffusion prediction models.

2 RELATED WORK

2.1 Information Diffusion Prediction

During the past decades, information diffusion prediction problems have been widely studied. Most of the early solutions [14, 20, 28] apply predefined diffusion models, such as the independent cascade (IC) model and linear threshold (LT) model. Stringent assumptions and heavy computational complexity limit the performance of this class of methods. In addition, data-driven embedding-based methods [2, 3, 10, 38] represent nodes in the social network and diffusion cascades as vectors, inferring propagation relationships between users through vector calculations. However, existing embedding models cannot effectively capture the complex patterns in real-world diffusion cascades.

With the fast development of deep learning, researchers utilize deep neural network-based end-to-end frameworks to capture latent patterns in information diffusion. Particularly, recurrent neural networks (RNNs) have shown excellent performance in sequence prediction. Due to the sequential nature of cascades, RNN has been widely applied to capture dependencies within cascades. Temporal and sequential features are the main factors considered by RNN-based models. Topo-LSTM [34] builds a recurrent model inferring topological information of cascades by modeling dynamic directed acyclic graphs. FOREST [41] encodes historical information through RNN and combines the hidden state with social structure information. NDM [40] employs convolutional neural networks in conjunction with attention mechanism for cascade modeling. However, the performance of RNN-based models decreases as the sequence length increases. Meanwhile, relying solely on sequential features cannot characterize diffusion processes accurately.

Due to the recent success of graph neural networks (GNNs), models based on GNN have demonstrated their effectiveness on tasks of diffusion prediction. Some researchers represent the diffusion information as graph data for processing through GNNs. Inf-VAE [29] utilizes GCN to encode social homophily. DyDiff-VAE [35] integrates GCN into GRU as the recurrent function to infer higher-order social influence. DyHGCN [44] constructs a heterogeneous graph to extract both the individual preference and the neighbor influence via GCN. HyperINF [19] infers high-order relations and cross-diffusion relations from constructed user interactive hypergraph and the diffusion interactive graph, respectively. MS-HGAT [31] captures users' interaction preferences by learning from a series of sequential hypergraphs. However, social features in social graphs and diffusion cascades cannot be sufficiently characterized by the simple use of GNNs. Asymmetric properties in social factors are ignored. In addition, existing methods that create embeddings for various components and integrate them could lead to information loss, resulting in limited performance.

2.2 Hyperbolic Representation Learning

Recently, studies indicate that hyperbolic space could effectively model hierarchical structures. Based on hyperbolic geometry, a series of hyperbolic deep learning frameworks [5, 8, 13, 15, 25, 45] have been proposed. Several studies have exhibited that models based on hyperbolic graph neural networks outperform traditional deep models in multiple downstream tasks, e.g., recommender systems [11, 30, 39, 43] and knowledge graph completion [1, 36]. For diffusion prediction, existing methods hardly consider the impact of hierarchical features on the information diffusion process. H-Diffu [12] builds embedding vectors for the social graph and diffusion graph in separate hyperbolic spaces and aggregates different embeddings with attention mechanisms. Different from H-diffu, we consider a unified approach to learn embeddings for different components in the same hyperbolic space.

2.3 Rotation Transformation

In recent years, rotation transformation has been applied to solve knowledge graph embedding problems [7, 27, 32, 42]. These studies show that rotation transformation could model various relation patterns in the knowledge graphs, enhancing the expressive power of embedding vectors. Moreover, several researchers introduce applying rotation transformation in the hyperbolic space. Chami et al. [4] suggest a method combining hyperbolic reflections and rotations to model complex asymmetric relations. Feng et al. [9] propose a hyperbolic rotation model ROLE to capture the asymmetric proximity between nodes. Inspired by them, we utilize rotation transformations to model social connections and diffusion connections in the information diffusion process. To the best of our knowledge, this is the first work exploring social relations using rotation transformation in the hyperbolic space.

3 PRELIMINARIES

In this section, we first state the research problem and then introduce the basics of hyperbolic geometry.

3.1 Problem Definition

A cascade $C_m = \{(u_1^m, t_1^m), (u_2^m, t_2^m), \cdots, (u_{L_m}^m, t_{L_m}^m)\}$ records the diffusion process of information item m in chronological order, where (u_i^m, t_i^m) means user i performs a propagation action, e.g., forwards a Twitter message, about item m at time t_i^m, and L_m is the length of cascade. All observed historical cascades are denoted as $C = \{C_m\}$. The cascade C_m can be simplified as $C_m = \{u_1, u_2, \cdots, u_{L_m}\}$ if we only record the orders and ignore the specific timestamps. Further, we can split a cascade C_m into k $(1 \leqslant k \leqslant L_m - 1)$ seed users $S_{C_m}^k = \{u_1, \cdots, u_k\}$ and a target user u_{k+1}.

Quantifying and measuring various social factors in the social data is essential to information diffusion prediction. Next, we introduce two main components used in the paper: social graph and diffusion graph, and state problem definition of information diffusion prediction.

A social graph $G_S = (V_S, E_S)$ is a directed graph that can be used to describe the social connections between users in a social network. V_S is the node set representing social users, and E_S is the edge set representing social connections. If a following relation exists from user u to user v, then there is a directed edge $(u \rightarrow v)$ in the social graph. Similarly, a diffusion graph $G_D = (V_D, E_D)$ is a directed graph that contains user diffusion connections. The diffusion graph is constructed from historical cascades C. V_D is the node set representing users of historical cascades, and E_D is the edge set representing diffusion actions. If $u, v \in V_S$ and we observed user u performed an action before user v, i.e., for a given cascade $C_m, u, v \in C_m$ and $t_u < t_v$, then there is a directed edge $(u \rightarrow v)$ in the diffusion graph. Note that we assume $V_D \subset V_S$.

After establishing the social graph and the diffusion graph, we map every social user to a representation vector in the hyperbolic space to capture his or her related social connection and diffusion connection. In this way, we could explore users' roles in the information diffusion process.

Definition 1. ***(Information Diffusion Prediction Problem).*** *Given a social graph $G_S = (V_S, E_S)$ and a set of historical cascades C, our goal is to learn a diffusion representation model to predict the target user $\{u | u \in V_S \backslash S_{C_m}^k\}$, i.e., the next user who will be activated by the k seed users $S_{C_m}^k = \{u_1, ..., u_k\}$ in a given cascade C_m.*

3.2 Hyperbolic Geometry

A hyperbolic space can be viewed as a n-dimensional Riemannian manifold of constant negative curvature. There are a number of models that can be used to represent hyperbolic spaces, including the Poincaré ball model, the Klein model, and the Lorentz model. In fact, these models are mathematically equivalent and can be converted to each other. Considering the numerical stability and effectiveness, we choose the Lorentz model as the embedding space. In addition, we could use the L2-norms of the Lorentzian embeddings to deduce the hierarchical structures[23].

A d-dimensional Lorentz model is defined as the Riemannian manifold $\mathcal{L}_\gamma^d = (\mathbb{L}_\gamma^d, g_\mathbf{x}^\gamma)$, where $g_\mathbf{x}^\gamma = \text{diag}([-1, 1, 1, \cdots, 1])$ denotes the metric tensor. The point set $\mathbb{L}_\gamma^d$ in the Lorentz model satisfy $\mathbb{L}_\gamma^d = \{\mathbf{x} \in \mathbb{R}^{d+1} : \langle \mathbf{x}, \mathbf{x} \rangle_\mathcal{L} = -\gamma\}$, where $\mathbf{x} = (x_0, x_1, \cdots, x_d)$ is $d + 1$ dimensional Euclidean vector with $x_0 = \sqrt{\gamma + \sum_{i=1}^d x_i^2} > 0$

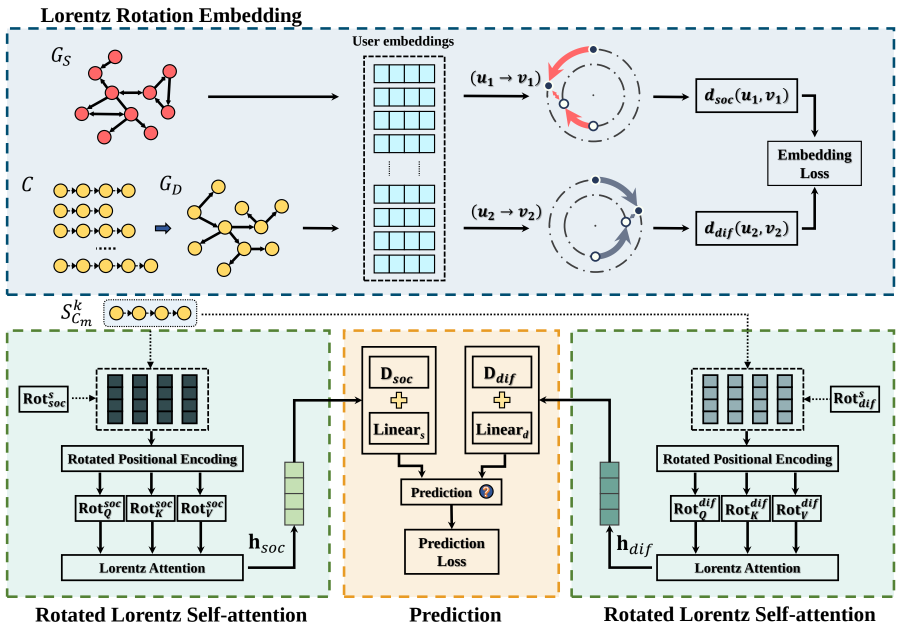


Figure 1: Overview of the RotDiff framework. The Lorentz rotation embedding trains user embeddings and encodes social relations in two groups of rotation transformations in the Lorentz model. The rotated Lorentz self-attention extracts diffusion dependencies within a cascade and generates corresponding latent embeddings. The prediction module estimates the probabilities of next-activated users.

and $\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}}$ means the Lorentzian scalar product which can be computed as:

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

where, in particular, we let $||\mathbf{x}||_{\mathcal{L}} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}}}$.

The Lorentz model has the constant negative curvature $-\gamma$, where $\gamma > 0$ reflects the curvature parameter. For any two points $\mathbf{x}, \mathbf{y} \in \mathbb{L}^d_\gamma$, there is always $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} \leqslant -\gamma$. Compared with the Poincaré model limiting all vectors in a ball, the vector distribution of the Lorentz model lies on the upper sheet surface of an unbounded hyperbolic model. When $\gamma = 1$, the Lorentz model can be view as an unit hyperboloid model [26].

According to [23], the squared Lorentz distance can reflect the hierarchical features and is easy to calculate. In the Lorentz model, the squared Lorentzian distance is defined as:

$$D_{SL}(\mathbf{x}, \mathbf{y}) = -2\gamma - 2 \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} . \tag{2}$$

In addition, rotation transformation is a kind of hyperbolic isometry and could be directly applied to Lorentzian vectors. Rotation transformation could be easily achieved as matrix multiplication. The rotation matrix $\mathbf{Rot} \in \mathbb{R}^{d \times d}$ is a block-diagonal matrix as:

$$\mathbf{Rot}_\Theta = \begin{bmatrix} \mathbf{R}(\theta_{r,1}) & & & \\ & \mathbf{R}(\theta_{r,2}) & & \\ & & \cdots & \\ & & & \mathbf{R}(\theta_{r,d/2}) \end{bmatrix}, \tag{3}$$

where, $\{\theta_{r,1}, \cdots, \theta_{r,d/2}\}$ denotes the parameters of the rotation, and the 2×2 block $\mathbf{R}(\theta_{r,i})$ is defined as:

$$\mathbf{R}(\theta_{r,i}) = \begin{bmatrix} \cos(\theta_{r,i}) & -\sin(\theta_{r,i}) \\ \sin(\theta_{r,i}) & \cos(\theta_{r,i}) \end{bmatrix} . \tag{4}$$

4 METHOD

The proposed RotDiff has three key modules as shown in Figure 1. We first build a Lorentz embedding for each social user and then encode social factors from the social and diffusion graphs using rotation transformations. Then, utilizing our proposed rotated Lorentz self-attention, we investigate propagation information inside a given cascade and construct latent embeddings that contain latent information for the future affected users. Finally, we employ the prediction module to estimate the possibility of the next-activated user.

4.1 Lorentz Rotation Embedding

The Lorentz rotation embedding maps social users to the Lorentzian embeddings and encodes social and diffusion connections into two groups of rotaion transformations. These two kinds of connections can potentially affect the diffusion of information. In previous work [12, 31, 44], different types of embeddings are learned independently from the social graph and diffusion cascades and then fused, which cannot depict users' influence and might result in information loss during the fusion process. Instead, our model assigns each user only a single vector and utilizes rotation transformations to capture different social factors.

As is shown in Figure 2, the basic idea is to globally minimize the distance between all relevant users. We say that two users are relevant if there exists an edge between two user nodes. We expect that user embeddings for relevant users are close to one another, and irrelevant users are far apart. We train rotation matrices to reduce the distance between two related embeddings without affecting the

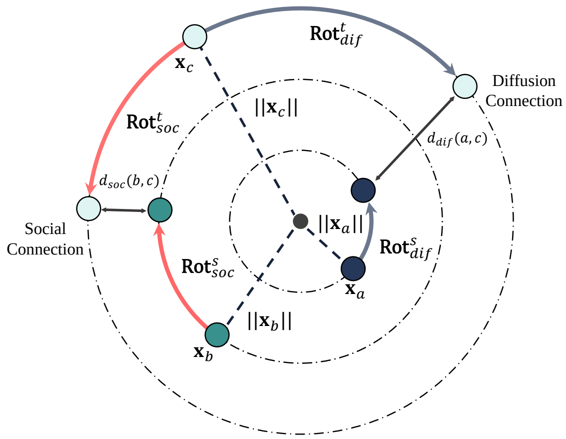


Figure 2: Illustration of the Lorentz rotation embedding. For edge $(a \to c)$ in G_D and edge $(b \to c)$ in G_S, each user node is assigned a Lorentzian vector. Two groups of rotation transformations Rot_{soc} and Rot_{dif} are used to model social connections and diffusion connections, respectively. After rotation, a pair of related nodes tends to have a smaller squared Lorentz distance. $d_{dif}(a,c) = D_{SL}(\mathbf{x}_a^{dif_s}, \mathbf{x}_c^{dif_t})$ and $d_{soc}(b,c) = D_{SL}(\mathbf{x}_b^{soc_s}, \mathbf{x}_c^{soc_t})$.

embeddings' modulus. In this way, we model different connections while preserving the strength of user influence.

Specifically, to generate user representations $\mathbf{x} \in \mathbb{L}^d_\gamma$, we first assign each user with a representation vector $\mathbf{z} \in \mathbb{R}^d$ and convert it into the Lorentz model, as the initial user representation $\mathbf{x}_0 \in \mathbb{L}^d_\gamma$, by a mapping function:

$$f : \mathbf{z} = (x_1, x_2, \cdots, x_d) \to \mathbf{x}_0 = (x_0, x_1, x_2, \cdots, x_d), \tag{5}$$

where $x_o = \sqrt{\gamma + \sum_{i=1}^{d-1} x_i^2} = \sqrt{\gamma + ||\mathbf{z}||^2}$, $||\cdot||$ is the L2-norm. Then, we utilize two groups of rotation transformations to learn social and diffusion roles for all users.

For the social graph, we take two rotation transformations $\mathbf{Rot}^s_{soc}$ and $\mathbf{Rot}^t_{soc}$ to encode social connections. Given an edge $(u \to v)$ in the social graph, let $\mathbf{x}_u$ and $\mathbf{x}_v$ represent the embeddings for user u and user v separately. Then we apply two rotation transformations to get rotated social embeddings as:

$$\mathbf{x}_u^{soc_s} = \mathbf{Rot}^s_{soc}(\mathbf{x}_u), \quad \mathbf{x}_v^{soc_t} = \mathbf{Rot}^t_{soc}(\mathbf{x}_v). \tag{6}$$

After the rotation operation, the squared Lorentzian distance D_{SL} $(\mathbf{x}_u^{soc_s}, \mathbf{x}_v^{soc_t})$ will be calculated to quantify the social connection between two users.

Similarly, for the diffusion graph, we employ a rotation transformation $\mathbf{Rot}^s_{dif}$ to describe the users' ability to influence, and another rotation transformation $\mathbf{Rot}^t_{dif}$ to represent the users' tendency to be affected. Give an edge $(u \to v)$ in the diffusion graph, we get two rotated diffusion embeddings of user u and user v after rotation operations as:

$$\mathbf{x}_u^{dif_s} = \mathbf{Rot}^s_{dif}(\mathbf{x}_u), \quad \mathbf{x}_v^{dif_t} = \mathbf{Rot}^t_{dif}(\mathbf{x}_v). \tag{7}$$

After rotation operation, the squared Lorentz distance $D_{SL}(\mathbf{x}_u^{dif_s}, \mathbf{x}_v^{dif_t})$ will be calculated to estimate the diffusion connections between two users. For the social and diffusion graphs, we manage to minimize the total squared Lorentzian distance between all related nodes.

Next, we introduce how to learn user embeddings and two groups of rotation matrices.

For both two graphs, related nodes are intuitively expected to be close to one another in the embedding space. Instead, unconnected nodes should be far apart. We utilize a scoring function to estimate the relevance of user nodes through the squared Lorentzian distance. The score would increase as the distance between two embeddings decreases. Thus, the scoring function is defined as:

$$\begin{aligned} S_{uv}^{soc} &= -D_{SL}(\mathbf{x}_u^{soc_s}, \mathbf{x}_v^{soc_t}) + b_u^{soc} + b_v^{soc}, \\ S_{uv}^{dif} &= -D_{SL}(\mathbf{x}_u^{dif_s}, \mathbf{x}_v^{dif_t}) + b_u^{dif} + b_v^{dif}, \end{aligned} \tag{8}$$

where u denotes the source node, v denotes the target node, b_u^{soc} and b_v^{soc} represent the biases for nodes of the social graph, b_u^{dif} and b_v^{dif} represent the biases for nodes of the diffusion graph.

If there exists an edge $(u \to v)$ in the social graph, we could use a Softmax function to estimate the probability as:

$$P(v|u) = \frac{e^{S_{uv}^{soc}}}{E(u)}, \tag{9}$$

where $E(u) = \sum_{i \in V_S} e^{S_{ui}^{soc}}$ is the normalization term whose calculation traverses all nodes in V_S. Here, we use a negative sampling technique to simplify the computational complexity. Further, let C_u be the context user set that contains all tail nodes headed by u. Assuming that tail nodes are independent to each other in C_u, we use log probability approximation to optimize all nodes in the social graph as:

$$\begin{aligned} \log P(v|u) &\approx \log \sigma(S_{uv}^{soc}) + \textstyle\sum_{i \in N} \log \sigma(-S_{ui}^{soc}), \\ O_{soc} &= \textstyle\sum_{u \in V_S} \log P(C_u|u) = \sum_{u \in V_S} \sum_{v \in C_u} \log P(v|u), \end{aligned} \tag{10}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the Sigmoid function.

In the similar way, O_{dif} of the diffusion graph can be computed as:

$$O_{dif} = \sum_{u \in V_D} \log P(D_u|u) = \sum_{u \in V_D} \sum_{v \in D_u} \log P(v|u), \tag{11}$$

where D_u is the context user set of user u in the diffusion graph. Therefore, the final embedding loss is calculated as:

$$\mathcal{L}_{emb} = -\left(O_{soc} + O_{dif}\right). \tag{12}$$

In this way, the objective of learning user embeddings $\mathbf{X} \in \mathbb{L}_\gamma^{d \times |V_S|}$ in terms of Lorentzian vectors for all social users has been achieved. Later, we use the learned embeddings and rotation matrices to address the diffusion prediction tasks.

4.2 Rotated Lorentz Self-attention

For the given k seed users $S^k_{C_m}$ in the cascade C_m, we expect to aggregate the information of seed users to predict the next-affected user, i.e., the target user u_{k+1}. To do so, we need an operation that could explore local diffusion dependencies within the seed users. Attention mechanism could execute weighted calculations on vectors. However, the challenge is ensuring that the calculation outputs are still meaningful vectors in the Lorentz model. Further, the obtained result should be effective for our prediction task. To achieve our goal, we propose a novel rotated Lorentz self-attention module that aggregates seed users' embeddings and generates the latent embedding, which contains needed information to predict the next-activated user.

Inspired by [6], we apply Lorentz attention based on the concept centroid. The basic idea of Lorentz attention is that the weight aggregation of a set of Lorentz vectors is identical to calculate the centroid that has the shortest squared Lorentzian distance to the set [6]. However, the expressive power of simply using the Lorentz attention is limited. To address this issue, we further strengthen the expressive power of latent embeddings by rotation transformations.

We first introduce the Lorentz attention. Given three Lorentzian vector sets, the query set $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{|Q|}\}$, the key set $\mathbf{K} = \{\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_{|K|}\}$, and the value set $\mathbf{V} = \{\mathbf{v}_2, \mathbf{v}_2, \cdots, \mathbf{v}_{|V|}\}$, where $|V| = |K|$, the Lorentz attention result $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_|Q|\}$ is calculated as:

$$\mathbf{y}_i = \frac{\sum_{j=1}^{|K|} \alpha_{ij}\mathbf{v}_j}{\sqrt{\gamma}\left|\left\|\sum_{k=1}^{|K|} \alpha_{ik}\mathbf{v}_k\right\|_{\mathcal{L}}\right|}, \tag{13}$$

where the coefficient a_{ij} is calculated as:

$$\alpha_{ij} = \frac{\exp\left(\frac{-D_{SL}(\mathbf{q}_i, \mathbf{k}_j)}{\sqrt{d}}\right)}{\sum_{k=1}^{|K|} \exp\left(\frac{-D_{SL}(\mathbf{q}_i, \mathbf{k}_j)}{\sqrt{d}}\right)}. \tag{14}$$

where d is the dimension of vectors and $D_{SL}(\cdot)$ denotes the calculation of the squared Lorentzian distance.

Then, we introduce how our rotated Lorentz self-attention module is designed. We note that two sets of learned rotation matrices contain different social relations. Therefore, we generate two types of latent embeddings, namely social latent embedding and diffusion latent embedding, that utilize various learned relations separately.

We first introduce how we get the social latent embedding. Given k seed users $S_{C_m}^k = \{u_1, \cdots, u_k\}$ of the cascade C_m, we first map seed nodes with their user embeddings rotated by the rotation matrix Rot_{soc}^s to get $\mathbf{x}_i^{soc_s} = \text{Rot}_{soc}^s(\mathbf{x}_i)$ for $1 \leqslant i \leqslant k$. The purpose here is to further utilize seed users' global social connections learned from the social graph.

Next, we propose rotation positional encoding to add sequential information of the cascade to the seed users. Let Rot_i^{pe} as a constant rotation matrix that could rotate embeddings by a fixed angle θ_i, where θ_i is only related to the sequential position i. After that, we get $\mathbf{x}_{p,i}^{soc_s} = \text{Rot}_i^{pe}(\mathbf{x}_i^{soc_s})$.

With three extra rotation operations, the social latent embedding $\mathbf{h}_{soc} \in \mathbb{L}_\gamma^d$ is calculated as:

$$\mathbf{h}_{soc} = \frac{\sum_{i=1}^{k} \alpha_i \mathbf{z}_{V,i}^{soc}}{\sqrt{\gamma}\left|\left\|\sum_{i=1}^{k} \alpha_i \mathbf{z}_{V,i}^{soc}\right\|_{\mathcal{L}}\right|},$$
$$\alpha_i = \frac{\exp\left(\frac{-D_{SL}(\mathbf{z}_{Q,i}^{soc}, \mathbf{z}_{K,i}^{soc})}{\sqrt{d}}\right)}{\sum_{j=1}^{k} \exp\left(\frac{-D_{SL}(\mathbf{z}_{Q,i}^{soc}, \mathbf{z}_{K,j}^{soc})}{\sqrt{d}}\right)}, \tag{15}$$

where, $\mathbf{z}_{Q,i}^{soc} = \text{Rot}_Q(\mathbf{x}_{p,i}^{soc_s})$, $\mathbf{z}_{K,i}^{soc} = \text{Rot}_K(\mathbf{x}_{p,i}^{soc_s})$ and $\mathbf{z}_{V,i}^{soc} = \text{Rot}_V(\mathbf{x}_{p,i}^{soc_s})$. Here, $\text{Rot}_Q^{soc}, \text{Rot}_K^{soc}, \text{Rot}_V^{soc}$ are three trainable rotation matrices.

Similarly, we could generate the diffusion latent embedding $\mathbf{h}_{dif}$. Next, we will introduce how we use $\mathbf{h}_{soc}$ and $\mathbf{h}_{dif}$ to predict the next-activated user.

Table 1: Statistics of datasets used in our experiments.

Dataset	#Nodes	#Edges	#Cascades	#Ave Length
Android	9,958	48,573	679	41.05
Christianity	2,897	35,624	587	25.10
Memetracker	4,709	-	12,661	16.24
Twitter	12,627	309,631	3,442	32.60
Douban	12,232	396,580	3,475	21.76

4.3 Prediction

We utilize two kinds of latent embeddings to achieve the prediction. Our estimate of the likelihood is based on the squared Lorentz distance. Users who are closer to the latent embedding are more likely to be influenced by related seed users. Further, we utilize additional linear layers to extract information in latent embeddings as compensation for each potential target user. Therefore, for the given $S_{C_m}^k$, two likelihoods $\mathbf{y}_{soc} \in \mathbb{R}^{1\times|U_S|}$ and $\mathbf{y}_{dif} \in \mathbb{R}^{1\times|U_S|}$ are calculated as:

$$\mathbf{y}_{soc} = \mathbf{D}_{soc} + \mathbf{h}_{soc}^T\mathbf{W}_s + \mathbf{b}_s, \quad \mathbf{y}_{dif} = \mathbf{D}_{dif} + \mathbf{h}_{dif}^T\mathbf{W}_d + \mathbf{b}_d, \tag{16}$$

where $\mathbf{D}_{soc} \in \mathbb{R}^{1\times|U_S|}$ and its element $\mathbf{D}_{soc,i} = -D_{SL}(\mathbf{h}_{soc}, \mathbf{x}_i^{soc_t})$, similarly, $\mathbf{D}_{dif} \in \mathbb{R}^{1\times|U_S|}$, $\mathbf{D}_{dif,i} = -D_{SL}(\mathbf{h}_{dif}, \mathbf{x}_i^{dif_t})$. $\mathbf{W}_s \in \mathbb{R}^{d\times|U_S|}$ and $\mathbf{W}_d \in \mathbb{R}^{d\times|U_S|}$ are two trainable weight matrices, $\mathbf{b}_s \in \mathbb{R}^{1\times|U_S|}$ and $\mathbf{b}_d \in \mathbb{R}^{1\times|U_S|}$ are the biases.

Thus, the activation probabilities $\hat{\mathbf{y}} \in \mathbb{R}^{1\times|U_S|}$ for all social users are calculated as:

$$\hat{\mathbf{y}} = \text{Softmax}\left(\mathbf{y}_{soc} + \mathbf{y}_{dif} + \mathbf{M}^{pre}\right), \tag{17}$$

$\mathbf{M}^{pre}$ is used to mask the seed users in the cascade. $\mathbf{M}_i^{pre} = -\infty$, if user i is a seed user, else $\mathbf{M}_i^{pre} = 0$.

Finally, let $|\mathcal{S}|$ represents the total number of seed user sets, the prediction loss is calculated via the cross-entropy loss function as:

$$\mathcal{L}_{pre} = -\sum_{j=1}^{|\mathcal{S}|}\sum_{i=1}^{|U_S|} \mathbf{y}_{j,i}\log(\hat{\mathbf{y}}_{j,i}), \tag{18}$$

where $\mathbf{y}_{j,i} = 1$ if user u_i is the target user, $\mathbf{y}_{j,i} = 0$ otherwise.

4.4 Objective Function

RotDiff needs to compute two parts of losses. The first part is the embedding loss from the Lorentz rotation embedding, while the second part is the prediction loss from the prediction process. Hence, the final loss function is calculated as:

$$\mathcal{L}_{total} = \mathcal{L}_{emb} + \mathcal{L}_{pre}. \tag{19}$$

The objective of the RotDiff is to minimize the sum loss $\mathcal{L}_{total}$, which could be optimized through Riemannian Adam [21].

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets. We evaluate the performance of our RotDiff on five publicly available real-world datasets: (1) **Android** [29] consists of users' interactions for an Android topic on various channels from a community Q&A website Stack-Exchanges. Questions and answers between users constitute their friendship relationship. (2) **Christianity** [29] contains the friendship network and cascade interactions associated with the Christian topic on Stack-Exchanges. (3) **Memetracker** [24] compiles millions of news articles and blog posts from internet sources and monitors the most popular terms. Each meme is viewed as a piece of information, and each URL is handled as a user. This dataset has no underlying social graph. (4) **Twitter** [16] comprises records of Twitter messages shared between users on Twitter, where users can follow one another. (5) **Douban** [46] is a compilation of book-sharing behaviors on the Douban website. The co-occurrence connection of users is interpreted as their social relation. The statistics of the used datasets are shown in Table 1.

5.1.2 Baselines. To demonstrate the effectiveness of RotDiff, we compare it with the following state-of-the-art models.

- **NDM** [40]: a model combining convolutional neural network with attention mechanism.
- **Inf-VAE** [29]: a variational autoencoder framework that uses GCN to encode social homophily and positional-encoding to encode temporal influence.
- **FOREST** [41]: a recurrent model that employs GRU to learn sequential features and extracts network structure information via GCN.
- **DyHGCN** [44]: a model based on GCN and attention mechanism that learns user preference from both the social graph and the diffusion graph.
- **HyperINF** [19]: a GNN-based model inferring diffusion dependencies by dual-channel hypergraph neural networks.
- **MS-HGAT** [31]: a GCN-based method that constructs hypergraphs to depict interaction dependencies within a cascade.
- **H-diffu** [12]: a hyperbolic presentation method jointly modeling the diffusion cascades and social graphs in different hyperbolic spaces.

5.1.3 Implementation Details. Following the evaluation protocol of [12, 31, 41, 44], we regard the diffusion prediction task as a retrieval problem. We employ the suggested default parameters from the original papers for all baselines. The curvature parameter γ is set to 1. The default batch size is 64. The number of dimensions d is set to 64 to maintain the same settings as the baseline approaches [12, 31, 44]. The default number of negative samples n is 20 which allows for reasonable prediction performance and efficiency. We optimize our model with Riemannian Adam [21]. For each dataset, we randomly split 80% as the training set, 10% as the validation set, and the remaining 10% as the test set. Similar to the work [31, 41, 44], we use Hits@k (hits score on top k) and MAP@k (mean average precision on tok k) as evaluation metrics. In our experiments, we set k as 10, 50, and 100, respectively. We build our model[1] in PyTorch and run tests on an Ubuntu server equipped with a 12-core Intel Xeon(R) 2.10 GHz CPU and two GTX 3090 GPUs.

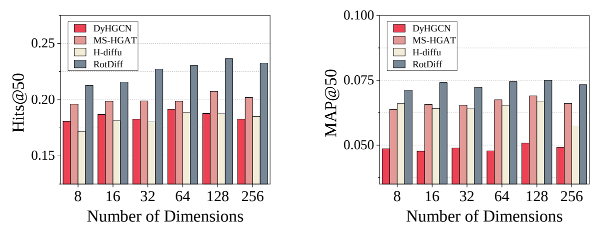
5.2 Experimental Results

5.2.1 Overall Results. The experimental results of the proposed model are presented in Table 2 and Table 3. In each table, we show the results of an evaluation metric on five datasets. The best scores are in boldface, and the second best scores are underlined. The reported results of the methods are the average values of five runs.

The RotDiff considerably outperforms all the state-of-the-art baselines on Hit@k and MAP@k metrics. Compared with previous models, RotDiff could effectively explore social connections and diffusion dependencies from the diffusion data to predict the target users in the future. In detail, methods applying GNNs to explore graph data generally perform well. Instead, NDM that utilizes CNN and attention mechanism gets limited performance. FOREST and Inf-VAE focus on modeling static user relations. Hence the obtained performance improvement is limited. DyHGCN and MS-HGAT achieve relatively high performance by creating a series of complex hypergraphs describing diffusion interactions. H-diffu as a hyperbolic embedding method has better performance than most methods based on Euclidean space.

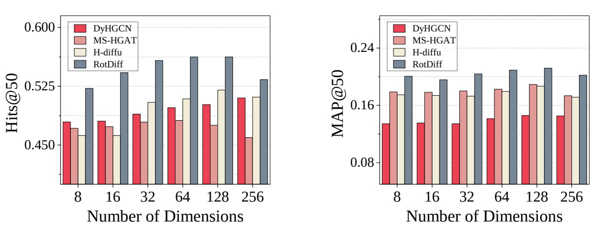
The experimental results of RotDiff demonstrate the effectiveness of our proposed representation model for diffusion prediction tasks. This suggests that compared to Euclidean methods, hyperbolic representations are more effective in preserving hierarchy features. Meanwhile, modeling the asymmetric features of social factors could accurately depict information diffusion processes. For datasets Android, Christianity, and Twitter, RotDiff achieves an average performance improvement of over 10%. Especially for Twitter, our model has improved up to 28% on MAP@k scores. For dataset Memetracker, despite the lack of the social graph, the performance is still improved over previous models by only modeling the diffusion graph, which indicates the flexibility of our model.

Overall, RotDiff consistently outperforms all the baselines, demonstrating its superiority in information diffusion prediction.



(a) Hits@50 (b) MAP@50

Figure 3: The effect of dimension d on Android.



(a) Hits@50 (b) MAP@50

Figure 4: The effect of dimension d on Christianity.

5.2.2 Effect of Embedding Dimension. We examine the influence of dimension d on two datasets Android and Christianity. The results are shown in Figure 3 and Figure 4. The results indicate that our proposed representation method has significant expressive power with different dimensions when compared with the other three baselines on Hits@50 and MAP@50 scores. The scores rise as d increases because larger dimensions may more accurately

[1] Code available at https://github.com/PlaymakerQ/RotDiff

Table 2: The prediction results of Hits@k on five datasets.

Dataset	Android			Christianity			Memetraker			Twitter			Douban		
Hits@k	@10	@50	@100	@10	@50	@100	@10	@50	@100	@10	@50	@100	@10	@50	@100
NDM	0.0339	0.0953	0.1572	0.1651	0.3510	0.4553	0.2083	0.3663	0.4583	0.1934	0.2941	0.3573	0.1013	0.2123	0.3125
Inf-VAE	0.0673	0.1573	0.2179	0.1774	0.3960	0.5215	0.2124	0.4077	0.4934	0.1476	0.3178	0.4512	0.1116	0.2214	0.3468
FOREST	0.0700	0.1514	0.2237	0.2632	0.4909	0.6056	0.2963	0.4780	0.5786	0.2552	0.3850	0.4607	0.1868	0.3084	0.3857
DyHGCN	0.0842	0.1915	0.2679	0.2594	0.4976	0.6047	0.2952	0.4864	0.5848	0.2901	0.4688	0.5719	0.1987	0.3289	0.3942
HyperINF	0.0848	0.1553	0.2236	0.2700	0.4460	0.5165	0.2483	0.4634	0.5949	0.2692	0.4442	0.5648	0.1834	0.3321	0.4016
MS-HGAT	0.1049	0.1987	0.2747	0.2781	0.4814	0.5703	0.2843	0.4966	0.6047	0.2996	0.4654	0.5735	0.2065	0.3504	0.4136
H-diffu	0.0981	0.1860	0.2623	0.2746	0.5089	0.6004	0.2195	0.4499	0.5720	0.2707	0.4533	0.5636	0.1984	0.3479	0.4155
RotDiff	**0.1144**	**0.2304**	**0.3130**	**0.3237**	**0.5625**	**0.6674**	**0.3066**	**0.5170**	**0.6206**	**0.3590**	**0.5246**	**0.6121**	**0.2216**	**0.3823**	**0.4637**

Table 3: The prediction results of MAP@k on five datasets.

Dataset	Android			Christianity			Memetraker			Twitter			Douban		
MAP@k	@10	@50	@100	@10	@50	@100	@10	@50	@100	@10	@50	@100	@10	@50	@100
NDM	0.0219	0.0244	0.0252	0.0676	0.0751	0.0765	0.1059	0.1131	0.1144	0.1296	0.1339	0.1348	0.0836	0.0879	0.0936
Inf-VAE	0.0426	0.0441	0.0482	0.1035	0.1194	0.1249	0.1345	0.1379	0.1446	0.1632	0.1725	0.1747	0.1044	0.1098	0.1142
FOREST	0.0381	0.0416	0.0426	0.1328	0.1433	0.1449	0.1553	0.1637	0.1751	0.1733	0.1790	0.1801	0.1086	0.1146	0.1183
DyHGCN	0.0458	0.0503	0.0514	0.1303	0.1415	0.1432	0.1611	0.1623	0.1725	0.1751	0.1832	0.1847	0.1048	0.1114	0.1148
HyperINF	0.0424	0.0461	0.0467	0.1629	0.1719	0.1732	0.1434	0.1545	0.1566	0.1679	0.1756	0.1774	0.1042	0.1139	0.1138
MS-HGAT	0.0633	0.0675	0.0685	0.1732	0.1825	0.1836	0.1542	0.1641	0.1657	0.1880	0.1951	0.1965	0.1122	0.1187	0.1198
H-diffu	0.0606	0.0643	0.0653	0.1689	0.1795	0.1808	0.1409	0.1514	0.1531	0.1777	0.1868	0.1885	0.1067	0.1017	0.1127
RotDiff	**0.0696**	**0.0745**	**0.0756**	**0.1981**	**0.2091**	**0.2105**	**0.1653**	**0.1691**	**0.1766**	**0.2406**	**0.2482**	**0.2495**	**0.1170**	**0.1254**	**0.1266**

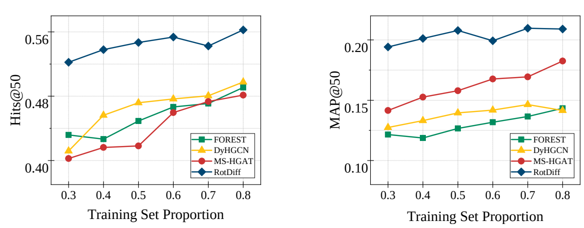
Table 4: The effect of curvature parameter γ.

Dataset	Android		Christianity	
Metrics (@50)	Hits	MAP	Hits	MAP
$\gamma = 0.1$	**0.2312**	0.0728	0.5402	0.2087
$\gamma = 0.3$	0.2296	0.0709	0.5513	0.2053
$\gamma = 0.6$	0.2258	0.0724	**0.5670**	0.2054
$\gamma = 1.0$	0.2304	**0.0745**	0.5625	**0.2091**
$\gamma = 1.5$	0.2289	0.0729	0.5603	0.2041
$\gamma = 2.0$	0.2120	0.0724	0.5491	0.2053

represent the complicated diffusion relationships. When the dimensionality exceeds a certain value (e.g., $d \geqslant 64$), the performance of RotDiff becomes saturated.

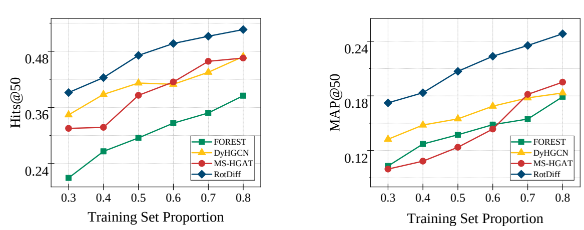
5.2.3 ***Effect of Curvature***. The hyperbolic space used in our model has a fixed curvature, and different curvature values may affect the prediction performance of the model. Thus, we investigate the effect of different curvatures parameter γ on Android and Christianity. The experimental results are shown in Table 4, where the best scores are in boldface. We observe that performance is not very sensitive to the value of γ. However, when the value of γ increases to 2, the performance of the model drops slightly. Empirically, we could simply set $\gamma = 1$ for most general cases.

5.2.4 ***Effect of Training Set Proportion***. The quantity of the training set may affect the performance of the model. Thus, we change the training set proportion to investigate model performance changes. To do so, we keep the validation and test sets unchanged and change the proportion of the training set to the whole dataset. The experimental results on two datasets are shown in Figure 5 and Figure 6. Hits@50 and MAP@50 are selected as evaluation metrics. On Christianity, RotDiff could outperform other three baselines on both metrics with just a small amount of training data (i.e., 30%). In terms of Twitter, the performance of all models exhibits a gradual improvement as the amount of training data increases. Nevertheless, Rotdiff only requires 50% of the training data to outperform other models under 80% of the training set. Overall, RotDiff achieves the best performance under any train set proportion, showing the effectiveness and stability of our approach.



(a) Hits@50 (b) MAP@50

Figure 5: The effect of training set proportion on Christianity.



(a) Hits@50 (b) MAP@50

Figure 6: The effect of training set proportion on Twitter.

5.3 Ablation Study

In this section, we investigate the effectiveness of the proposed modules of RotDiff through a series of ablation experiments on

Table 5: The results of ablation study.

Dataset	Christianity		Twitter	
Metrics (@50)	Hits	MAP	Hits	MAP
(1) w/o hyperbolic	0.5312	0.1928	0.5102	0.2287
(2) only soc-graph	0.5446	0.1993	0.5087	0.2300
(3) only dif-graph	0.5580	0.1974	0.5210	0.2319
(4) w/o Lo-rot-emb	0.5369	0.1955	0.4867	0.2137
(5) w/o all-att	0.5133	0.1845	0.4783	0.2032
(6) w/o Rot-in-att	0.5446	0.1924	0.5186	0.2295
RotDiff	**0.5625**	**0.2091**	**0.5246**	**0.2482**

various model components. Specifically, we conduct the following ablation studies: (1) **w/o hyperbolic.** Implement our model RotDiff in Euclidean space. (2) **only soc-graph.** Remove diffusion graph modeling, we only use Lorentz rotation embedding on the social graph. (3) **only dif-graph.** Remove social graph modeling, we only use Lorentz rotation embedding on the diffusion graph. (4) **w/o Lo-rot-emb.** Remove the whole module Lorentz rotation embedding. Instead, we just assign each user a trainable embedding with random initialization. (5) **w/o all-att.** Remove the whole rotated Lorentz self-attention. Instead, we use a general self-attention module proposed in the work [33]. (6) **w/o Lo-att.** Remove extra rotation operations of the attention module, and we only calculate the centroid using Equation (13).

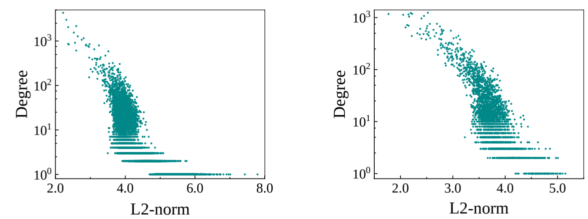
The experimental results of ablation study are reported in Table 5. We have the following observations. The result (1) indicates that Hits@50 and MAP@50 scores drop on both datasets when the model is deployed in Euclidean space. This suggests that the underlying hierarchical structures of the data can be captured in the hyperbolic space, leading to better prediction performance.

The results (2)-(4) are related to **Lorentz Rotation Embedding**. Scores on both datasets degrade to some extent when the Lorentz-rotated embeddings are removed. This suggests that social connections and diffusion dependencies are important for diffusion prediction. We observe that modeling merely the diffusion graph has better scores than modeling only the social network, indicating that diffusion dependencies have a more significant role than social connections. Moreover, when the model solely relies on historical cascade information via the attention mechanism without learning user representations from both graphs, its performance degrades significantly due to the absence of capturing crucial social factors. Obviously, the best results are obtained when modeling both graphs simultaneously.

The results (5) and (6) are related to **Rotated Lorentz Self-attention**. We find that after removing two classes of rotation transformations in the self-attention module, both the Hits@50 and MAP@50 scores would decrease to a certain extent. This shows that proposed rotation operations could further strengthen the expressive ability of user embeddings and latent embeddings. In addition, when a general self-attention module replaces the whole rotated Lorentz self-attention, the model's performance drops significantly. Since the output of the self-attention can no longer be guaranteed to be Lorentz vectors, it leads to a deviation in the feature information contained in the embeddings.

Overall, the results of the ablation study demonstrate the importance of each component. By integrating these components, the RotDiff model can obtain the best experimental results.

5.4 Visualization



(a) Android (b) Christianity

Figure 7: The correlation between the L2-norm of user embeddings and node degree.

We investigate the features of user embeddings in the hyperbolic space through the visualization method. In Figure 7, we present the correlation between the L2-norm of learned user embeddings and node degree on Android and Christianity. We have the following findings. First, the latent hierarchical features in the social data are captured. In general, user nodes with higher degrees have larger influence. Hence, the distribution of user influence in two datasets has apparent characteristics of power law distribution. Then, the larger the degree of user nodes, the smaller the L2-norms of related user embeddings. In the hyperbolic space, user nodes with strong influence strength tend to approach the origin. In contrast, user nodes with weak influence would move gradually away from the origin due to their limited connections with other nodes. In summary, the hyperbolic space is well-suited for solving problems related to social influence, including information diffusion prediction.

6 CONCLUSION

In this paper, we study the information diffusion prediction task. We observe that existing models cannot effectively describe underlying hierarchical structures and learn complex asymmetric social factors, leading to limited prediction performance. Embedding in the hyperbolic space can depict the underlying hierarchical strictures. Further, rotation transformation could effectively capture complex asymmetric features and enhance the expressive power of learned embeddings. Motivated by our observations, we embed social users in the hyperbolic space and extract various social relations with trainable rotation matrices. Thus, we propose a new hyperbolic rotation representation model RotDiff to address the diffusion prediction problem. RotDiff incorporates three main modules: the Lorentz rotation embedding, the rotated Lorentz self-attention, and the prediction module. We optimize the hyperbolic representations by learning the social and diffusion graphs simultaneously. We apply the proposed rotated Lorentz self-attention for a given diffusion cascade to infer the diffusion dependencies during information propagation. The empirical results on five real-world datasets demonstrate that our proposed model RotDiff significantly outperforms various state-of-the-art baselines.

ACKNOWLEDGMENTS

This work was supported by National Key Research and Development Program of China under No. 2021YFC3300200 and supported in part by the National Natural Science Foundation of China under Grant No. 62202124 and No. 61972111.

REFERENCES

[1] Yushi Bai, Zhitao Ying, Hongyu Ren, and Jure Leskovec. 2021. Modeling heterogeneous hierarchies with relation-specific hyperbolic cones. *Advances in Neural Information Processing Systems* 34 (2021), 12316–12327.

[2] Simon Bourigault, Cedric Lagnier, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari. 2014. Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 393–402.

[3] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. 2016. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the Ninth ACM international conference on Web Search and Data Mining*. 573–582.

[4] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6901–6914.

[5] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems* 32 (2019).

[6] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Fully Hyperbolic Neural Networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland.

[7] Yao Dong, Lei Wang, Ji Xiang, Xiaobo Guo, and Yuqiang Xie. 2022. RotateCT: Knowledge Graph Embedding by Rotation and Coordinate Transformation in Complex Space. In *Proceedings of the 29th International Conference on Computational Linguistics*. 4918–4932.

[8] Shanshan Feng, Lisi Chen, Kaiqi Zhao, Wei Wei, Fan Li, and Shuo Shang. 2021. Node2lv: squared lorentzian representations for node proximity. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2015–2020.

[9] Shanshan Feng, Lisi Chen, Kaiqi Zhao, Wei Wei, Xuemeng Song, Shuo Shang, Panos Kalnis, and Ling Shao. 2022. ROLE: Rotated Lorentzian Graph Embedding Model for Asymmetric Proximity. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[10] Shanshan Feng, Gao Cong, Arijit Khan, Xiucheng Li, Yong Liu, and Yeow Meng Chee. 2018. Inf2vec: Latent representation model for social influence embedding. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 941–952.

[11] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. 2020. Hme: A hyperbolic metric embedding approach for next-poi recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on research and development in information retrieval*. 1429–1438.

[12] Shanshan Feng, Kaiqi Zhao, Lanting Fang, Kaiyu Feng, Wei Wei, Xutao Li, and Ling Shao. 2022. H-Diffu: Hyperbolic Representations for Information Diffusion Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[13] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. *Advances in neural information processing systems* 31 (2018).

[14] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*. 241–250.

[15] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. 2018. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786* (2018).

[16] Nathan O Hodas and Kristina Lerman. 2014. The simple rules of social contagion. *Scientific reports* 4, 1 (2014), 1–7.

[17] Mohammad Raihanul Islam, Sathappan Muthiah, Bijaya Adhikari, B Aditya Prakash, and Naren Ramakrishnan. 2018. Deepdiffuse: Predicting the'who'and'when'in cascades. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 1055–1060.

[18] Lokesh Jain, Rahul Katarya, and Shelly Sachdeva. 2023. Opinion Leaders for Information Diffusion Using Graph Neural Network in Online Social Networks. *ACM Trans. Web* 17, 2 (apr 2023). https://doi.org/10.1145/3580516

[19] Hai Jin, Yao Wu, Hong Huang, Yu Song, Haohui Wei, and Xuanhua Shi. 2022. Modeling Information Diffusion With Sequential Interactive Hypergraphs. *IEEE Transactions on Sustainable Computing* 7, 3 (2022), 644–655.

[20] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 137–146.

[21] Max Kochurov, Rasul Karimov, and Serge Kozlukov. 2020. Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819* (2020).

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

[23] Marc Law, Renjie Liao, Jake Snell, and Richard Zemel. 2019. Lorentzian Distance Learning for Hyperbolic Representations. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 3672–3681.

[24] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 497–506.

[25] Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019. Hyperbolic graph neural networks. *Advances in Neural Information Processing Systems* 32 (2019).

[26] Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*. PMLR, 3779–3788.

[27] Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. Chronor: Rotation based temporal knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6471–6479.

[28] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. 2008. Prediction of information diffusion probabilities for independent cascade model. In *International conference on knowledge-based and intelligent information and engineering systems*. Springer, 67–75.

[29] Aravind Sankar, Xinyang Zhang, Adit Krishnan, and Jiawei Han. 2020. Inf-VAE: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. In *Proceedings of the 13th international conference on web search and data mining*. 510–518.

[30] Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. 2021. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference 2021*. 593–601.

[31] Ling Sun, Yuan Rao, Xiangbo Zhang, Yuqian Lan, and Shuanghe Yu. 2022. MS-HGAT: Memory-enhanced Sequential Hypergraph Attention Network for Information Diffusion Prediction. In *AAAI'22*.

[32] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[34] Jia Wang, Vincent W Zheng, Zemin Liu, and Kevin Chen-Chuan Chang. 2017. Topological recurrent neural network for diffusion prediction. In *2017 IEEE international conference on data mining (ICDM)*. IEEE, 475–484.

[35] Ruijie Wang, Zijie Huang, Shengzhong Liu, Huajie Shao, Dongxin Liu, Jinyang Li, Tianshi Wang, Dachun Sun, Shuochao Yao, and Tarek Abdelzaher. 2021. Dydiff-vae: A dynamic variational framework for information diffusion prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 163–172.

[36] Shen Wang, Xiaokai Wei, Cicero Nogueira Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew Arnold, Bing Xiang, Philip S Yu, and Isabel F Cruz. 2021. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *Proceedings of the Web Conference 2021*. 1761–1771.

[37] Zhitao Wang, Chengyao Chen, and Wenjie Li. 2018. A sequential neural information diffusion model with structure attention. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 1795–1798.

[38] Zhitao Wang, Chengyao Chen, and Wenjie Li. 2019. Information diffusion prediction with network regularized role-based user representation learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13, 3 (2019), 1–23.

[39] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*. 70–79.

[40] Cheng Yang, Maosong Sun, Haoran Liu, Shiyi Han, Zhiyuan Liu, and Huanbo Luan. 2019. Neural diffusion model for microscopic cascade study. *IEEE Transactions on Knowledge and Data Engineering* 33, 3 (2019), 1128–1139.

[41] Cheng Yang, Jian Tang, Maosong Sun, Ganqu Cui, and Zhiyuan Liu. 2019. Multi-scale Information Diffusion Prediction with Reinforced Recurrent Networks. In *IJCAI'19*. 4033–4039.

[42] Jinfa Yang, Xianghua Ying, Yongjie Shi, Xin Tong, Ruibin Wang, Taiyan Chen, and Bowei Xing. 2022. Learning Hierarchy-Aware Quaternion Knowledge Graph Embeddings with Representing Relations as 3D Rotations. In *Proceedings of the 29th International Conference on Computational Linguistics*. 2011–2023.

[43] Menglin Yang, Zhihao Li, Min Zhou, Jiahong Liu, and Irwin King. 2022. Hicf: Hyperbolic informative collaborative filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2212–2221.

[44] Chunyuan Yuan, Jiacheng Li, Wei Zhou, Yijun Lu, Xiaodan Zhang, and Songlin Hu. 2020. Dyhgcn: A dynamic heterogeneous graph convolutional network to learn users' dynamic preferences for information diffusion prediction. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 347–363.

[45] Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. 2021. Lorentzian graph convolutional networks. In *WWW'21*. 1249–1261.

[46] Erheng Zhong, Wei Fan, Junwei Wang, Lei Xiao, and Yong Li. 2012. Comsoc: adaptive transfer of user behaviors over composite social network. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 696–704.