

# Information Diffusion Prediction with Graph Neural Ordinary Differential Equation Network

## ABSTRACT

Information diffusion prediction aims to forecast the path of information spreading in social networks. Prior works generally consider the diffusion process to be driven by user correlations or preferences. Recent works focus on characterizing the dynamicity of user preferences and propose to capture users' dynamic preferences by discretizing the diffusion process into structure snapshots. Despite their effectiveness, these works summarize user preferences from partially observed structure snapshots, ignoring that users' preferences are evolving constantly. Moreover, discretizing the diffusion process makes these models overlook abundant structure information across different periods, reducing their ability to discover potential participants. To address the above issues, we propose a novel Graph Neural Ordinary Differential Equation Network (GODEN) for information diffusion prediction, which incorporates neural ordinary differential equations (ODE) to model the continuous dynamics of the diffusion process. Specifically, we design two coupled ODE functions on nodes and edges to describe their co-evolution dynamic and infer user dynamic preferences based on the solution of ODEs. Besides, we extract user correlations from a heterogeneous graph to complement user encoding for prediction. Finally, to predict the future user infections of the observed cascade, we represent its diffusion pattern in terms of user and temporal contexts and apply a multi-head attention module to attend to different contexts. Experimental results confirm our approach's effectiveness on four real-world datasets, with our model outperforming the state-of-the-art diffusion prediction models.

## CCS CONCEPTS

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

## KEYWORDS

social network, information diffusion prediction, graph neural network, ordinary differential equations

## ACM Reference Format:

. 2024. Information Diffusion Prediction with Graph Neural Ordinary Differential Equation Network. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/nmnnnnn.nmnnnnn>

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or professional use, is granted by ACM, provided that the copies are not made for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

Conference'17, July 2017, Washington, DC, USA

© 2024 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . \$15.00

<https://doi.org/10.1145/nmnnnnn.nmnnnnn>

2024-04-13 12:09. Page 1 of 1–10.

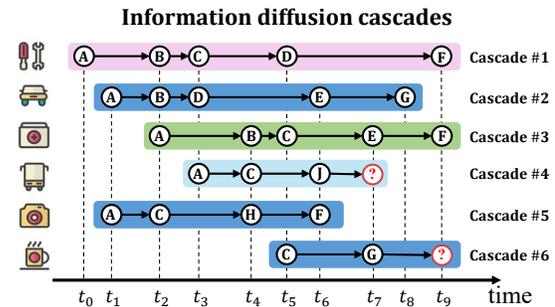


Figure 1: A simple example of information diffusion cascades. Icons on the left represent the topic of cascades. Suppose the overall diffusion process is split into three diffusion periods  $[t_0, t_4)$ ,  $[t_4, t_7)$ ,  $[t_7, t_9)$ .

## 1 INTRODUCTION

In recent years, multimedia social networks have become indispensable ways to publish and distribute information. Massive users interact with others online, facilitating the rapid dissemination of information and forming information cascades [37]. Effective prediction of future participants in information cascades has become a challenging but critical task for numerous social applications, such as social recommendation [13, 30] and disinformation control [29].

Information diffusion prediction problems have spurred significant research interest for decades. Existing works can be principally summarized into three categories. 1) *Feature engineering-based models* [1, 34] assume that the diffusion process abides by predefined diffusion functions or models. They generally extract representative features of users and cascades to compute the diffusion probabilities and fit them with predefined diffusion functions or models at the macro level. However, these models are hardly generalized to different domains due to their restrictive assumptions. 2) *Sequence-based models* [10, 26, 32] model cascades as sequence data and exploit sequence models, e.g., RNN or attention layer, to extract user correlations within diffusion paths. Despite their progress, their emphasis on sequential data caused them to neglect the impact of social relations, failing to extend user correlations beyond sequences. 3) *Graph-based models* [22, 27] introduce various graph structures to extend user correlation, such as social networks and diffusion networks. Most recently, some researchers [19, 31, 35] find that user preferences also have a crucial impact on the diffusion process. Considering the dynamic nature of users' preferences, they construct structure snapshots to discretize the diffusion process into multiple periods and introduce graph neural networks (GNNs) to describe users' preferences at each period, which achieves encouraging prediction performance.

However, current models with dynamic graphs focus on summarizing users' preferences at each period based on partially observed structure snapshots, which brings two natural deficiencies. For one,

59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116

current models ignore the fact that users' preferences are changing constantly. Users may participate in different cascades in the same diffusion period and show their varying preferences, *e.g.*, user A at the period  $[t_0, t_4]$  in Figure 1. Summarizing all user interactions at certain diffusion periods to describe their preferences brings more harm than good in prediction. For example, when we predict the subsequent users in Cascade #4, fusing users' preferences can hardly achieve prediction, since only user A participates in relevant cascade in period  $[t_0, t_4]$ . For another, current models concentrate on partially observed structures in each period, making them neglect abundant user correlation across different periods. As shown in Figure 1, when predicting the future diffusion trends in Cascade #6 after timestamp  $t_7$ , if we only consider user interactions within each diffusion period, we find User G interacts with nobody. However, from a global perspective, we could observe diffusion structure  $C \rightarrow D \rightarrow F$ ,  $C \rightarrow E \rightarrow F$ , and  $C \rightarrow H \rightarrow F$  from Cascade #1, Cascade #3, and Cascade #5, indicating that user F is a potential participant.

To solve the above problems, we propose a novel Graph neural Ordinary Differential Equation Network (short for **GODEN**) for information diffusion prediction. The key idea of GODEN is to characterize the evolutionary dynamics of the diffusion process based on graph structures. To achieve this goal, we first leverage three types of relations to comprehensively characterize the diffusion process and apply GNN to obtain the initial state of users. Then, we design two coupled ODEs on users and their relations to characterize their co-evolution dynamics since they are deeply correlated in the diffusion process. We apply a channel attention mechanism to infer users' dynamic preferences based on the solution of the ODE function. Besides, we apply a graph neural network to capture users' static correlations from a global perspective to complement the users' dynamic preferences. To predict future user infections, we first represent the diffusion pattern of the observed cascade based on its user context and temporal context. Then, we apply a multi-head self-attention mechanism to attend to different contexts and solve the information diffusion prediction problem.

Our main contributions are summarized as follows:

- We propose two coupled ODEs to learn the evolution pattern for users and relations in the diffusion process and infer users' dynamic preferences.
- We extract users' static correlations to extend their dynamic preferences and represent the specific diffusion pattern of cascades by learning the user context and temporal context information to promote the prediction.
- We conducted extensive experiments on four public datasets. The results show that GODEN outperforms state-of-the-art models on the information diffusion prediction task, demonstrating its effectiveness.

## 2 RELATED WORK

### 2.1 Information Diffusion Prediction

The information diffusion prediction task aims to predict future user infections based on historical diffusion paths, which have been widely studied over decades. Existing diffusion prediction models are mainly categorized into three categories: (1) feature engineering models, (2) sequence-based models, and (3) graph-based models.

Early feature-engineering models believed the diffusion process adheres to specific diffusion models, such as the independent cascade model [11] and epidemic models. However, their stringent assumptions constrain their ability to characterize complex diffusion patterns in the real world. Some embedding-based models further improve these models. They encode users as embeddings by maximizing specific diffusion functions and infer future diffusion probability through vector calculations. Although effective, this class of methods still hardly generalizes to different social scenarios.

Sequential-based models emerged along with the fast development of deep learning. They [10, 23, 32] treat diffusion cascades as sequences and regard the information diffusion prediction as a sequential prediction task. Various sequential models, *e.g.*, LSTM [8] or attention mechanisms [21], are applied to capture sequential and temporal features in information diffusion. NDM [32] employs multi-head attention mechanisms to model the diffusion sequence and incorporated convolution neural networks (CNNs) to alleviate long-term memory loss in sequential modelling. HiDAN [28] builds a hierarchical attention network to jointly capture user dependency and the time decay effect in the diffusion sequence. However, users' social structures, as one of the critical channels of information diffusion, are generally overlooked by sequential-based models.

Due to the recent success of graph neural networks (GNNs), graph-based models have demonstrated their effectiveness in tasks of diffusion prediction. Various graph structures, mostly social graphs, are also widely exploited to extract non-sequential user correlations. SNIDSA [27] builds a novel recurrent model to jointly incorporate social information from users and sequential information from cascades. FOREST [33] encodes historical information in cascades with GRU and combines it with neighbour information from the social structure. CEGCN [22] jointly models users and cascades in the same heterogeneous graph and extracts collaborative diffusion patterns via graph neural networks. Most recently, some studies [24, 35] have found that users' preferences play a critical role in facilitating information diffusion. Since users' preferences change as time passes, they model the diffusion process as a series of structure snapshots and employ graph neural networks to capture users' dynamic preferences. DyHGNC [35] extracts neighbour influence and diffusion preferences as users' dynamic preferences via a heterogeneous GCN. MS-HGAT [19] introduces a sequential hypergraph to capture users' interaction preferences and integrate them with static social relations to predict information diffusion.

Despite effectiveness, models with dynamic graphs simply summarize users' preferences based on observed structure snapshots from each diffusion period, which ignores the continuous evolution of users' preferences and the abundant structure information across different periods.

### 2.2 Graph Neural Networks

GNN is a class of neural networks that operate directly on graph-structured data and have shown remarkable performance in various domains. Recently, some researchers have extended GNNs to dynamic domains to capture the chronological characteristics of graph structures. These models are generally split into discrete-time dynamic GNNs (DTDGNNs) and continuous-time dynamic GNNs

(CTDGNNs). DTDGNNs [4, 17] discretize dynamic graphs as multiple structure snapshots and apply static GNN to process each snapshot for node representation. For CTDGNNs [20, 25], they represent dynamic graphs as a series of node interactions with precise timestamps in chronological order. They design specific recurrent modules to aggregate historical messages to update the node state.

For diffusion prediction, existing dynamic-graph-based works rely on DTDGNNs to summarize the users' preferences at different periods, failing to consider the continuous evolution of user preferences. Unlike existing works, we consider modelling the continuous evolution of users' preferences with ODEs.

## 2.3 Ordinary Differential Equation

Neural ODEs [3] have been proposed as a new paradigm for generalizing discrete deep neural networks to continuous-time scenarios. They specify the dynamics of the hidden state using a neural network  $f_\theta$  with parameters  $\theta$ . Given an initial state  $\mathbf{x}(0)$  They define a hidden state  $\mathbf{x}(t)$  as a solution to the ODE initial-value problem (IVP):

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \frac{d\mathbf{x}}{d\tau} d\tau = \mathbf{x}(0) + \int_0^t f_\theta(\mathbf{x}(\tau), \tau) d\tau \quad (1)$$

$\mathbf{x}(t)$  can be evaluated at any desired time through a numerical ODE solver without any internal operations [3], which allows Neural ODE to be built as a block for the whole neural network. Due to their superior performance and flexible capability, neural ODEs have been widely adopted in various research fields, such as traffic flow forecasting [6] and sequential recommendation [16]. Recently, some advanced methods connect GNNs with neural ODEs. GODE [14] generalizes the concept of continuous-depth models to graphs and parameterizes the derivative of hidden node states with GNNs. Inspired by the outstanding performance of ODEs in dynamic systems, we introduce ODEs to model the continuous dynamics in the diffusion process.

## 3 PRELIMINARY

### 3.1 Problem Formulation

Normally, an information diffusion process is recorded as a cascade  $c_m = \{(u_1^m, t_1^m), (u_2^m, t_2^m), \dots, (u_{L_m}^m, t_{L_m}^m)\}$  in chronological order, where element  $(u_i^m, t_i^m)$  denotes that user  $u_i^m$  performs an action to participation  $c_m$  at time  $t_i^m$ , e.g., forwarding a Twitter message.  $L_m$  is the maximum cascade length. The cascade  $c_m$  can be further divided into user sequence  $c_m^u = \{u_1^m, u_2^m, \dots, u_{L_m}^m\}$  or timestamp sequence  $c_m^t = \{t_1^m, t_2^m, \dots, t_{L_m}^m\}$ , if we focus only on the orders of users or the temporal information in the cascade  $c_m$ . We collect all historical cascades and users in  $C = \{c_1, c_2, \dots, c_{|C|}\}$  and  $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ , respectively. Moreover, describing and quantifying various user relations in the diffusion process is essential to the information diffusion prediction task. We then introduce graph structures used in the paper: social graph  $\mathcal{G}_s$ , diffusion graph  $\mathcal{G}_d$ , and bipartite graph  $\mathcal{G}_b$ . They are shown in the right of Figure 2. The social graph  $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$  is a directed graph that describes The social connections among users.  $\mathcal{V}_s$  is the set of nodes representing social users.  $\mathcal{E}_s$  is the set representing users' social relations. If the following relation exists from user  $u_i$  to user  $u_j$ , a directed edge  $u_i \rightarrow u_j$  will be added to edge set  $\mathcal{E}_s$ . Similarly, the diffusion

graph  $\mathcal{G}_d = \{\mathcal{V}_d, \mathcal{E}_d\}$  is a directed graph formed by users' diffusion connections.  $\mathcal{V}_d$  is the node set representing users in the historical cascades.  $\mathcal{E}_d$  is the edge set representing diffusion actions. If we observe user  $u_i$  forward information from user  $u_j$ , there is a directed edge  $u_i \rightarrow u_j$  in the diffusion graph. The bipartite graph describes the connection between cascades and their corresponding users.  $\mathcal{V}_d$  is the node set that contains both user and cascade nodes.  $\mathcal{E}_d$  is the edge set representing the connection between users and cascades. We add a directed edge between each cascade node and its users, i.e.,  $u_i \rightarrow c_i, u_i \in c_i$ .

Based on the above introductions, we describe the task of **information diffusion prediction** as: given the set of user  $\mathcal{U}$ , the set of historical cascades  $C$ , and an observed cascade  $c_o = \{(u_i^o, t_i^o) | u_i^o \in \mathcal{U}, i < L_{c_o}\}$ .  $L_{c_o}$  refers to the maximum length of  $c_o$ . Our goal is to compute the conditional probability  $\hat{y}_j = p(u_j | c_o)$  to show how likely user  $u_j$  will participate in this cascade at the next timestamp.

## 4 METHOD

This section introduces our graph neural ordinary differential equation network (GODEN). The overall architecture of GODEN is shown in Figure 2, which has three major components: 1) User encoding module, which generates user embeddings by modeling the dynamic evolution of the diffusion process and capturing users' static correlations. 2) Cascade representation module, which represents the diffusion pattern based on the temporal and user contexts in the observed cascade. 3) Prediction module, which applies a multi-head attention module to calculate the infection probability of candidates.

### 4.1 User Encoding

**4.1.1 Dynamic Preference Encoding.** The information diffusion process is affected by different factors. To comprehensively describe the diffusion process, we first merge social graph  $\mathcal{G}_s$ , diffusion graph  $\mathcal{G}_d$ , and bipartite graph  $\mathcal{G}_b$  into heterogeneous graph  $\mathcal{G}_h = \{\mathcal{V}_h, \mathcal{E}_h, \mathcal{W}_h\}$ .  $\mathcal{V}_h = \{v_h | v_h \in \mathcal{V}_u \cup \mathcal{V}_b\}$  is the set of nodes constructed by both the user nodes and the cascade nodes.  $\mathcal{E}_h = \{\mathcal{E}_s \cup \mathcal{E}_d \cup \mathcal{E}_b\}$  is the edge set with three types of relations.  $\mathcal{W}_h = \{e_h | e_h \in \mathcal{E}_h\}$  is the set of edge weights to differentiate edge effects. We initialize the weight of all edges as 1.

To infer the users' dynamic preferences, we first encode both users and cascades into embeddings as their initial state in the diffusion process. We apply a GNN layer for user encoding and design different aggregation strategies for user and cascade nodes since they share different neighbor contexts in heterogeneous graph  $\mathcal{G}_h$ .

Specifically, for user node  $u_i$  at the  $(l+1)^{\text{th}}$  GNN layer, its neighbor contexts contain both user and cascade nodes. Therefore, we divide the neighbors by the edge types and separately aggregate

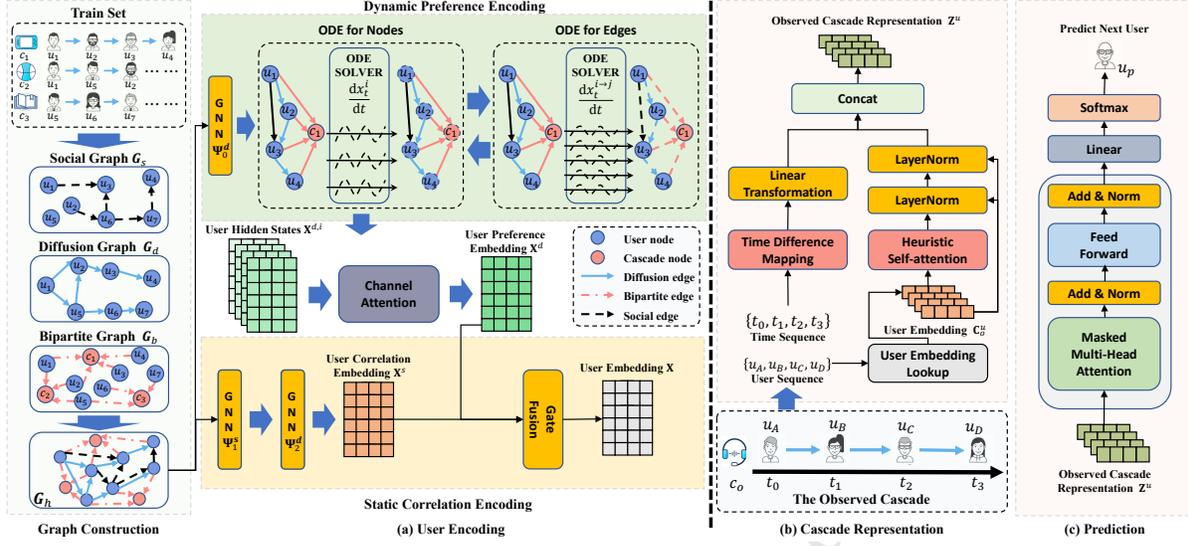


Figure 2: The overview architecture of GODEN.

their contextual information, which is formulated as,

$$\begin{aligned} \mathbf{a}_{\mathcal{N}_u}^{u_i(l+1)} &= f\left(\frac{e_h^{ij}}{|\mathcal{N}_u|} \sum_{u_j \in \mathcal{N}_u} \mathbf{W}_{u_j}^{u_i(l+1)} \mathbf{x}^{u_j(l)}\right), \\ \mathbf{a}_{\mathcal{N}_c}^{u_i(l+1)} &= f\left(\frac{e_h^{ik}}{|\mathcal{N}_c|} \sum_{c_k \in \mathcal{N}_c} \mathbf{W}_{c_k}^{u_i(l+1)} \mathbf{x}^{c_k(l)}\right), \\ \mathbf{x}^{u_i(l+1)} &= \text{MLP}\left[\mathbf{a}_{\mathcal{N}_u}^{u_i(l+1)}; \mathbf{a}_{\mathcal{N}_c}^{u_i(l+1)}; \mathbf{x}^{u_i(l)}\right], \end{aligned} \quad (2)$$

where  $\mathbf{x}_{u_j}^{(l)}$  and  $\mathbf{x}_{c_k}^{(l)}$  are the user embedding and cascade embedding from the last layer.  $\mathbf{W}_{u_j}^{u_i(l+1)}$  and  $\mathbf{W}_{c_k}^{u_i(l+1)}$  are learnable weight matrices to aggregate contextual features from different neighbors.  $\mathcal{N}_u$  and  $\mathcal{N}_c$  are set of user node and cascade node neighbors that shares edges with  $u_i$ .  $e_h^{ik}$  and  $e_h^{ij}$  is the corresponding edge weights.  $f(\cdot)$  means the activation and norm operations.

Similarly, we perform context aggregation operations for cascade nodes. We aggregate the user neighbors to construct cascade embeddings. For the cascade node  $c_i$ , the process is formulated as,

$$\begin{aligned} \mathbf{a}_{\mathcal{N}_u}^{c_i(l+1)} &= f\left(\frac{e_h^{ij}}{|\mathcal{N}_u|} \sum_{u_j \in \mathcal{N}_u} \mathbf{W}_{u_j}^{c_i(l+1)} \mathbf{x}^{u_j(l)}\right), \\ \mathbf{x}^{c_i(l+1)} &= \text{MLP}\left[\mathbf{a}_{\mathcal{N}_u}^{c_i(l+1)}; \mathbf{x}^{c_i(l)}\right], \end{aligned} \quad (3)$$

where  $\mathbf{x}_{u_j}^{(l)}$  is the user node embedding from the last layer.  $\mathbf{W}_{u_j}^{c_i(l+1)}$  is a learnable weight matrix to aggregate contextual features from user neighbors.  $\mathcal{N}_u$  is a set of user nodes that share bipartite relations with cascade node  $c_i$ .

As our model leverages the same GNN layer at different modules to aggregate structural contexts, we denote the above GNN layer as function  $\Psi(\mathcal{G}, \mathbf{X})$ , where graph  $\mathcal{G}$  and node embeddings  $\mathbf{X}$  is the input of GNN layers. Thus, the GNN layer to get node initial state is  $\Psi_0^d(\mathcal{G}_h, \mathbf{X}^{d(0)})$ , where the input node embedding matrix

$\mathbf{X}^{d(0)} \in \mathbb{R}(|\mathcal{U}|+|\mathcal{C}|) \times d$  is generated from a normal distribution [7]. The embedding matrix to represent the initial state of nodes is  $\mathbf{X}_0^d$ .

After obtaining the initial states for nodes, we characterize the continuous dynamics of the diffusion process with ODE functions. In diffusion processes, the states of users and their relations are deeply correlated and could affect each other. For example, if a user is interested in a piece of information and forwards it to his social friends, the relationship between these users will be closer. Similarly, if a user shares tight relationships with others, he is more likely to be influenced by them and obtain information from them, which will affect his preferences. Therefore, we propose to infer the future state of users and their relations with different but coupled ODE functions.

Typically, the state of edges is determined by the user it connects and their initial attributes. We concatenate the node's initial state to represent the initial attributes of the edges and leverage the current node state to infer edge states. Based on the current state of edges, we could further update the graph structures to obtain future states of users. Thus, the ODE for edges is defined as,

$$\begin{aligned} \mathbf{x}_0^{i \rightarrow j} &= [\mathbf{x}_0^{u_i, d}; \mathbf{x}_0^{u_j, d}], \\ \frac{d\mathbf{x}_t^{i \rightarrow j}}{dt} &= \text{MLP}_e\left(\left[\mathbf{x}_t^i \parallel \mathbf{x}_t^j\right]\right) + \text{MLP}_{\text{init}}\left(\mathbf{x}_0^{i \rightarrow j}\right), \\ e_t^{ij} &= \text{MLP}_{\text{weight}}\left(\mathbf{x}_t^{i \rightarrow j}\right), \end{aligned} \quad (4)$$

where  $\mathbf{x}_0^{i \rightarrow j}$  is the initial edge attributes.  $[\cdot; \cdot]$  means concatenation operation.  $\mathbf{x}_t^{i \rightarrow j}$  represents the edge states at timestamp  $t$ .  $e_t^{ij}$  is the new edge weights at timestamp  $t$ . The graph structure at timestamp  $t$  is  $\mathcal{G}_h^t = \{\mathcal{U}_h, \mathcal{E}_h, \mathcal{W}_h^t\}$ .

Intuitively, the state of users is determined by their neighbors and diffusion preference bias at the current timestamp. Moreover, their preference bias at the initial state also deeply affects their preferences in the future. Thus, we define the ODE function for

nodes as follows,

$$\frac{d\mathbf{X}_t^d}{dt} = \Psi_d(\mathcal{G}_h^t, \mathbf{X}_t^d) - \mathbf{X}_t^d + \mathbf{X}_0^d \quad (5)$$

where  $\mathbf{X}_t^d \in \mathbb{R}^{(|\mathcal{U}|+|C|) \times d}$  denotes the matrix representing all node states at timestamp  $t$ .  $\mathcal{G}_h^t$  is the heterogeneous graph with new edge weights at timestamp  $t$ .

Since we have modeled the dynamics of nodes and edges in the diffusion process with Eq. 4 and Eq. 5. Given a continuous time  $t$ , the value of nodes and edges can then be solved by a designated ODE solver:

$$\begin{aligned} \mathbf{X}_t^d &= \text{ODESolver} \left( \frac{d\mathbf{X}_t^d}{dt}, \mathbf{X}_0^d, t \right) \\ \mathbf{x}_t^{i \rightarrow j} &= \text{ODESolver} \left( \frac{d\mathbf{x}_t^{i \rightarrow j}}{dt}, \mathbf{x}_0^{i \rightarrow j}, t \right) \end{aligned} \quad (6)$$

To accurately and comprehensively describe users' dynamic preferences, we take  $N$  solving steps for the ODE solver and obtain user hidden states  $\{\mathbf{X}_1^d, \mathbf{X}_2^d, \dots, \mathbf{X}_N^d\}$  at different timestamps. We develop a channel attention mechanism to integrate users' hidden state to infer users' dynamic preferences. Formally, the users' dynamic preference embedding can be computed as,

$$\alpha^i = \frac{\exp(\mathbf{a} \cdot \mathbf{W}_a \mathbf{X}_i^d)}{\sum_{j \in N} \exp(\mathbf{a} \cdot \mathbf{W}_a \mathbf{X}_j^d)}, \mathbf{X}^d = \sum_{i \in N} \alpha^i \mathbf{X}_i^d, \quad (7)$$

where  $\mathbf{a} \in \mathbb{R}^d$  and  $\mathbf{W}_a \in \mathbb{R}^{d \times d}$  are trainable parameters. User dynamic preference embedding matrix is denoted as  $\mathbf{X}^d \in \mathbb{R}^{|\mathcal{U}| \times d}$ .

**4.1.2 Static Correlation Encoding.** In the diffusion process, user behaviors do not strictly abide by their preferences. Instead, they may also follow specific interaction patterns. For example, although social advertisers may show preferences for different products at different times, they tend only to forward information to potential buyers and reject sharing other information. Therefore, we extract static user correlations from a global perspective to complement users' dynamic preferences.

Since heterogeneous graph  $\mathcal{G}_h$  contains all historical user interactions and social relations, we directly apply two layers of GNN to extract users' correlation from the heterogeneous graph, which is,

$$\mathbf{X}^{s(2)} = \Psi_2^s(\mathcal{G}_h, \Psi_1^s(\mathcal{G}_h, \mathbf{X}^{s(0)})), \quad (8)$$

where input node embedding matrix  $\mathbf{X}^{s(0)} \in \mathbb{R}^{(|\mathcal{U}|+|C|) \times d}$  is randomly initialize with normal distribution [7]. As cascade embeddings hardly provide help for prediction, we only collect the user embedding from  $\mathbf{X}^{s(2)}$  as static correlation embedding matrix  $\mathbf{X}^s \in \mathbb{R}^{|\mathcal{U}| \times d}$ .

We introduce a gated fusion strategy to integrate users' dynamic preferences with static correlations. For user  $u_j$ , his embeddings  $\mathbf{x}_{u_j}$  is derived from the following procedure,

$$\begin{aligned} g_j &= \sigma(\mathbf{x}^{u_j, d} \mathbf{W}_d + \mathbf{x}^{u_j, s} \mathbf{W}_s), \\ \mathbf{x}^{u_j} &= g_j \mathbf{x}^{u_j, d} + (1 - g_j) \mathbf{x}^{u_j, s}, \end{aligned} \quad (9)$$

where  $\mathbf{W}_s$  and  $\mathbf{W}_d$  are trainable parameters. We perform the same operation for each user and obtain the user embedding matrix  $\mathbf{X}^u$ .

## 4.2 Cascade Representation

The future diffusion path of a cascade is affected by its diffusion pattern, which can be reflected through the order of previously infected users and infected timestamps in the cascade, *i.e.*, user context and temporal context. In this section, we represent the diffusion pattern in the observed cascade according to its user and temporal contexts to assist prediction.

**4.2.1 Temporal Context Representation.** Empirical studies [2, 10] have shown that the influence of diffusion cascades decreases over time, which is known as the time-decay effect. They consider the diffusion process as a temporal point process and estimate the activation timestamp through the time difference between each user interaction. Inspired by them, we propose to capture the temporal context by mapping the time difference into vector space with a neural function.

For timestamp sequence  $c_o^t = \{t_1^o, t_2^o, \dots, t_{L_o}^o\}$  of observed cascade  $c_o$ , we characterize its temporal context by the time difference between each diffusion behavior, which is represented as  $\Delta t_j^o = t_{i+1} - t_i, 1 < i < L_o$ . We map each time difference  $\Delta t_i^o$  into a new time label based on the constructed time intervals. The time label is calculated by following the function,

$$\lambda_i^o = \left\lfloor \frac{\Delta t_i^o - t_{min}}{\lceil (t_{max} - t_{min}) / l_t \rceil} \right\rfloor, \quad (10)$$

where  $t_{max}, t_{min}$  are predefined maximum and minimum time differences.  $l_t$  is the number of time slots used to discretize the time difference in the observed cascade. We transform  $\lambda_i^o$  into one-hot embedding  $\mathbf{t}_i^o \in \mathbb{R}^{l_t}$ . Each element in  $\mathbf{t}_i^o$  is set to 0, except for the element in position  $\lambda_i^o$  is set to 1. Finally, we generate temporal context encoding by capturing the time-decay pattern via an MLP layer,

$$\mathbf{z}_i^{o, t} = \mathbf{tanh}(\mathbf{W}_t \mathbf{t}_i^o + \mathbf{b}_t), \quad (11)$$

where  $\mathbf{W}_t \in \mathbb{R}^{l_t \times d}$  and  $\mathbf{b}_t \in \mathbb{R}^d$  are learnable parameters.  $\mathbf{tanh}(\cdot)$  is the activation function.

**4.2.2 User Context Representation.** Although we have encoded users into embeddings, these embeddings contain massive contextual information from different relations and structures, which hardly reflects the specific diffusion pattern in the observed cascade. Thus, we incorporate a heuristic self-attention mechanism to filter out relevant information among participants and aggregate them to represent specific user contexts.

For the user sequence  $c_o^u = \{u_1^o, u_2^o, \dots, u_{L_o}^o\}$  of the observed cascade, we first look up the user embedding matrix  $\mathbf{X}^u$  to transform it into the user embedding sequence  $\mathbf{z}_o^u = [\mathbf{x}_1^o, \mathbf{x}_2^o, \dots, \mathbf{x}_{L_o}^o]$ . Then, we compute relevant scores between each user  $u_j \in c_o^u$  and his context user  $u_k \in \{u_1, \dots, u_{j-1}\}$  and apply the weighted attention score sum to aggregate the relevant information. Specifically, the context-enhanced embedding  $\hat{\mathbf{x}}_j^o$  for  $u_j$ , is calculated as,

$$\begin{aligned} \beta_{kj} &= \frac{\exp(\sigma(\mathbf{W}_k \mathbf{x}_k^o) \odot \sigma(\mathbf{W}_j \mathbf{x}_j^o))}{\sum_{r=1}^{j-1} \exp(\sigma(\mathbf{W}_r \mathbf{x}_r^o) \odot \sigma(\mathbf{W}_j \mathbf{x}_j^o))}, \\ \hat{\mathbf{x}}_j^o &= \sum_{k=1}^{j-1} \beta_{kj} \mathbf{x}_k^o, \end{aligned} \quad (12)$$

where  $\beta_{kj}$  denotes the relevance score between user  $u_j$  and  $u_k$ .  $\mathbf{W}_k^d, \mathbf{W}_j^d \in \mathbb{R}^{d \times d}$  are transformation matrices to map the user embeddings into the different linear spaces to measure their correlation.  $\sigma(\cdot)$  is the activation function.  $\odot$  denotes the Hadamard product operation for user embeddings.

Moreover, information diffusion is a stochastic process driven by multiple factors [36], which does not always abide by the diffusion patterns learned from previously infected users. To comprehensively characterize the user context in the observed cascade, we apply a recursive residual connection layer to fuse the context-enhanced embedding sequence with the user embedding sequence to extend the contextual information for each user. Specifically, for user  $u_j$ , the user context representation is derived from the following procedure,

$$\mathbf{z}_j^{o,u} = \text{LN}(\mathbf{x}_j^o + \text{LN}(\mathbf{x}_j^o + \hat{\mathbf{x}}_j^o)) \quad (13)$$

where  $\text{LN}(\cdot)$  means the layer normalization.

Finally, we concatenate temporal and user contexts to represent the observed cascade. For user  $u_j$ , his representation  $\mathbf{z}_j^o$  is computed as,

$$\mathbf{z}_j^o = [\mathbf{z}_j^{o,u}; \mathbf{z}_j^{o,t}], \quad (14)$$

where  $[\cdot]$  is the concatenate operation. The observed cascade  $c_o$  is represented as  $\mathbf{Z}^o = [\mathbf{z}_1^o, \mathbf{z}_2^o, \dots, \mathbf{z}_{L_{c_o}}^o] \in \mathbb{R}^{L_{c_o} \times d}$ .

### 4.3 Prediction

Although we have represented the observed cascade as  $\mathbf{Z}^o$  by jointly exploring its user and temporal contexts, the context-dependence between each user is still unclear to achieve the prediction. Therefore, we apply a multi-head decoding layer to attend to different contextual information in the observed cascade efficiently. The process could be formulated as follows:

$$\begin{aligned} \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_h}} + \mathbf{M}\right) \mathbf{V}, \\ \mathbf{o}_i^d &= \text{Attention}\left(\mathbf{Z}^o \mathbf{W}_i^Q, \mathbf{Z}^o \mathbf{W}_i^K, \mathbf{Z}^o \mathbf{W}_i^V\right), \\ \mathbf{Z}^h &= [\mathbf{o}_1^d; \mathbf{o}_2^d; \dots; \mathbf{o}_H^d] \mathbf{W}^O, \end{aligned} \quad (15)$$

where  $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d \times d_h}$ , and  $\mathbf{W}^O \in \mathbb{R}^{H \times d_h \times d}$  are learnable parameters.  $H$  is the number of heads in the multi-head self-attention module.  $d_h$  is the scaling factor.  $\mathbf{M} \in \mathbb{R}^{L_{c_o} \times L_{c_o}}$  is a matrix to mask out future users to avoid label leakage, which is denoted as,

$$\mathbf{M}_{ij} = \begin{cases} 0 & \text{otherwise,} \\ -\infty & i \geq j. \end{cases} \quad (16)$$

Then, we apply two layers of fully connected neural networks to obtain attentive cascade representation  $\mathbf{Z}^p$ :

$$\mathbf{Z}^p = \sigma(\mathbf{Z}^h \mathbf{W}_1^h + \mathbf{b}_1) \mathbf{W}_2^h + \mathbf{b}_2 \quad (17)$$

where  $\mathbf{W}_1^h, \mathbf{W}_2^h$  are all learnable transformation matrices.  $\mathbf{b}_1, \mathbf{b}_2$  are bias parameters.

Finally, we use the predicted cascade representations  $\mathbf{Z}_p$  to calculate infected probabilities  $\hat{\mathbf{y}}_{ij} \in \mathbb{R}^{L_{c_o} \times |\mathcal{U}|}$  for all users, i.e.,

$$\hat{\mathbf{y}}_{ij} = \text{softmax}(\mathbf{W}^p \mathbf{Z}_p + \text{Mask}) \quad (18)$$

where  $\mathbf{W}_p \in \mathbb{R}^{d \times d}$  is a learnable parameter to calculate the infect probability for each user. We utilize  $\text{Mask} \in \mathbb{R}^{|\mathcal{U}| \times L_{c_o}}$  matrix to mask users who have already been activated in the observed cascade sequence. We adopt cross-entropy loss as the objective to optimize the information diffusion prediction task:

$$\mathcal{L}(\theta) = - \sum_{i=2}^{c_o} \sum_{j=1}^{|\mathcal{U}|} \mathbf{y}_{ij} \log(\hat{\mathbf{y}}_{ij}), \quad (19)$$

where  $\theta$  represents all parameters that need to be learned in the model  $\mathbf{y}_{ij} = 1$  denotes that the predicted user  $u_j$  is infected at timestamp  $t_i^o$ , otherwise  $\mathbf{y}_{ij} = 0$ .

## 5 EXPERIMENT

### 5.1 Experimental Setups

**5.1.1 Datasets.** We incorporate four publicly available real-world datasets to evaluate the performance of our model. The detailed statistics of the datasets are presented in Table 2. (1) **Twitter** [9] records the tweets with URLs during October 2010 on Twitter and its diffusion paths. The social relations are pre-defined by the following relation on Twitter. (2) **Douban** [19] is collected from a Chinese social website named Douban, where people can share their book reading statuses. The co-occurrence connection of users is interpreted as their social relations. (3) **Android** [18] is collected from Stack-Exchanges, a community Q&A website. Cascade refers to a series of chronologically ordered posts associated with the tag "Android". The social relation is pre-defined by user interactions, e.g., answering or commenting on the same post. (4) **Meme-tracker**<sup>1</sup> [12] tracks the migration of frequent quotes and phrases, i.e. memes. Each URL is treated as a user in the dataset.

**5.1.2 Baselines.** To evaluate the performance of GODEN, we select eight information diffusion prediction models as baselines for comparison. We preserve the original parameter settings for each model. The baseline models are (1) **NDM** [32] utilizes the self-attention mechanism and convolution modules to attend to long-term user correlation in cascade sequences. (2) **FOREST** [33] is a recurrent model that employs GRU to learn sequential features and extracts network structure information via GCN. (3) **CEGCN** [22] utilizes GNNs to exploit collaborative patterns from other cascades for prediction. (4) **DyHGAT** [35] applies GCN to learn users' dynamic preferences by discretising the diffusion process into heterogeneous subgraphs (6) **MS-HGAT** [19] constructs a series of hyper-graphs to model user interactions and integrate them with static social relations to depict interaction dependencies among users. (7) **DisenIDP** [5] leverages two hyper GCN to learn users' intents in the diffusion process and designs a self-supervised disentanglement task to assist the procedure. (8) **RotDiff** [15] maps the users into the hyperbolic representation space based on the social relations and diffusion paths, which achieves state-of-the-art performances.

**5.1.3 Implementation Details.** We implement our model in PyTorch and conduct our experiments on an Ubuntu server equipped with two 32 GB Nvidia V100 GPUs. GODEN is trained based on the Adam optimizer with parameters  $\beta_1$  and  $\beta_2$  set to 0.90 and 0.99, respectively. The learning rate is set as 0.001. The batch size in

<sup>1</sup><http://memetracker.org/>

**Table 1: Experimental results on HITS score over four datasets (%).**

Models	Android			Memetracker			Twitter			Douban		
	H@10	H@50	H@100									
NDM	0.0339	0.0953	0.1572	0.2083	0.3663	0.4583	0.1934	0.2941	0.3573	0.1013	0.2123	0.3125
FOREST	0.0700	0.1514	0.2237	0.2963	0.4780	0.5786	0.2552	0.3850	0.4607	0.1868	0.3084	0.3857
CEGCN	0.1075	0.2109	0.2842	0.2951	0.5021	0.6112	0.3381	0.5140	0.5987	0.2078	0.3483	0.4267
DyHGNCN	0.0842	0.1915	0.2679	0.2952	0.4864	0.5848	0.2901	0.4688	0.5719	0.1987	0.3289	0.3942
MS-HGAT	0.1049	0.1987	0.2781	0.2843	0.4966	0.6047	0.2996	0.4654	0.5735	0.2065	0.3504	0.4136
DisenIDP	0.0946	0.1916	0.2684	0.3074	0.5199	0.6280	0.3273	0.4799	0.5540	0.2059	0.3545	0.4284
RotDiff	<u>0.1144</u>	<u>0.2304</u>	<u>0.3130</u>	<u>0.3066</u>	<u>0.5170</u>	<u>0.6206</u>	<u>0.3590</u>	<u>0.5246</u>	<u>0.6121</u>	<u>0.2216</u>	<u>0.3823</u>	<u>0.4637</u>
GODEN	<b>0.1201</b>	<b>0.2401</b>	<b>0.3269</b>	<b>0.3379</b>	<b>0.5430</b>	<b>0.6399</b>	<b>0.3811</b>	<b>0.5578</b>	<b>0.6475</b>	<b>0.2490</b>	<b>0.3926</b>	<b>0.4729</b>
Improve.(%)	4.98	4.21	4.44	10.21	5.03	3.11	6.16	6.33	5.78	12.36	2.69	1.98

**Table 2: Statistics of the datasets.**

Datasets	Twitter	Douban	Android	Memetracker
# Users	12,627	12,232	9,958	4,709
# Social Links	309,631	348,280	48,573	-
# Cascades	3,442	3,475	679	12,661
Avg. Repost	10.74	6.18	2.27	44.00
Avg. Length	32.60	21.76	33.3	16.24

the training set is 16. The dimension of user embeddings is all set to  $d = 64$ . For user dynamic preference embeddings, we set the hidden dimensions of GNN  $\Psi_0^d(\cdot)$  as 128. The hidden state of edge and nodes in ODEs are 256 and 128, respectively. We use Runge-Kutta-4(RK-4) as the solver of our coupled ODE. For user static correlation embeddings, we set the hidden dimensions of GNN  $\Psi_1^s(\cdot)$  and  $\Psi_1^c(\cdot)$  as 128 and 64, respectively. The time interval  $l_t$  to map the time difference in the observed cascades into vector space is set to 5000. The dimensionality of temporal context encoding  $d_t$  is set to 8. The number of heads  $H$  in a multi-head attention module is chosen from  $\{8, 10, 12, 16\}$  and set to 8 after comparison. For all four datasets, we randomly sample 80% of cascades for training and split the remaining 20% evenly for validation and testing. The maximum cascade length is set to 200 for all datasets. Since information diffusion prediction aims to predict user participation by ranking all uninfected users according to their infection probabilities, Following the evaluation protocol of previous works [19, 35], we consider the task as an information retrieval task and evaluate the performance of information diffusion prediction models with two ranking metrics, *i.e.*, Mean Average Precision on top  $K$  and HITS scores on top  $K$ , with  $K = [10, 50, 100]$ . We abbreviate them as MAP@K ( $M@K$ ) and Hits@K ( $H@K$ ), respectively.

## 5.2 Experimental Results

The experimental results of the information diffusion prediction task are shown in Table 1 and Table 3. Numbers in bold denote the best results among all models and the underlined ones denote the second best results. Improvements in GODEN are statistically significant with  $p < 0.01$  on paired  $t$ -test. With the result, we have the following observations, (O1) GODEN consistently and significantly outperforms all state-of-the-art baselines on all four datasets under

different evaluation metrics. The relative improvements over the best-performing baseline are at least 6.18% (O2) Generally, methods applying graph data to explore user correlations beyond cascade sequence perform well. Instead, NDM focuses on learning user correlation in the sequence with attention mechanism gets limited performance. FOREST, CEGCN, and DisenIDP extend user relations with graph structure to improve prediction performance. DyHGNCN and MS-HGAT achieve relatively high performance by creating a series of graph snapshots to describe the diffusion process. RotDiff represents users in hyperbolic space to predict infection probability, which achieves better performance than most methods based on Euclidean space. (O3) GODEN achieves significant improvements compared to all baselines. We attribute the improvement to two reasons. For one, we leverage coupled ODE to model the continuous dynamics of users and relations in the diffusion process, which allows GODEN to infer users' preferences accurately. For another, we represent the diffusion pattern based on user and temporal context in the observed cascade. This specific context information could assist GODEN in retrieving users with similar preferences and filtering out irrelevant users.

## 5.3 Ablation Study Results

To validate the contribution of each component in GODEN, we design six variants for our model, which are 1) GODEN<sub>DP</sub> removes user dynamic preference embedding, which only utilizes the user static correlation embedding for prediction. 2) GODEN<sub>SC</sub> removes the static correlation embedding of the user, which only utilizes the dynamic preference embedding of the user for prediction. 3) GODEN<sub>CA</sub> removes channel attention to fuse multiple user preferences embeddings and only solves coupled ODEs in one step. 4) GODEN<sub>ODE</sub> removes the ODE solver and leverages 3 layers of GNN and MLP layer to model the dynamics of nodes and edges. 5) GODEN<sub>EdgeODE</sub> removes edge ODE function and solves Eq. 5 with fixed graph structure. 6) GODEN<sub>UC</sub> removes the user context embedding in the cascade representation module and directly fuses user embedding and temporal context for prediction.

The results of these variants are shown in Table 4. By analyzing the results, we have the following observations: (1) All variants suffer performance drops compared with GODEN, which shows that each component is essential for prediction. Moreover, GODEN<sub>UC</sub>

**Table 3: Experimental results on MAP score over four datasets.**

Models	Android			Memetracker			Twitter			Douban		
	M@10	M@50	M@100									
NDM	0.0160	0.0202	0.0210	0.0931	0.1031	0.1048	0.1169	0.1243	0.1256	0.0581	0.0651	0.0663
FOREST	0.0381	0.0416	0.0426	0.1553	0.1637	0.1751	0.1733	0.1790	0.1801	0.1086	0.1146	0.1183
CEGCN	0.0706	0.0750	0.0760	0.1654	0.1749	0.1765	0.2107	0.2189	0.2202	0.1202	0.1267	0.1278
DyHGCN	0.0633	0.0675	0.0685	0.1542	0.1641	0.1657	0.1880	0.1951	0.1965	0.1122	0.1187	0.1198
MS-HGAT	0.0458	0.0503	0.0514	0.1611	0.1623	0.1725	0.1751	0.1832	0.1847	0.1048	0.1114	0.1148
DisenIDP	0.0582	0.0623	0.0634	0.1624	0.1722	0.1737	0.2159	0.2228	0.2239	0.1041	0.1110	0.1121
RotDiff	<u>0.0696</u>	<u>0.0745</u>	<u>0.0756</u>	<u>0.1653</u>	<u>0.1691</u>	<u>0.1766</u>	<u>0.2406</u>	<u>0.2482</u>	<u>0.2495</u>	<u>0.1170</u>	<u>0.1254</u>	<u>0.1266</u>
GODEN	<b>0.0739</b>	<b>0.0792</b>	<b>0.0804</b>	<b>0.1939</b>	<b>0.2036</b>	<b>0.2050</b>	<b>0.2480</b>	<b>0.2562</b>	<b>0.2575</b>	<b>0.1486</b>	<b>0.1553</b>	<b>0.1564</b>
Improve.(%)	6.18	6.31	6.35	17.3	20.4	16.08	3.08	3.22	3.21	27.01	23.84	23.54

**Table 4: Ablation study on three datasets.**

Models	Twitter				Douban				Android			
	H@50	H@100	M@50	M@100	H@50	H@100	M@50	M@100	H@50	H@100	M@50	M@100
GODEN	<b>0.5578</b>	<b>0.6475</b>	<b>0.2480</b>	<b>0.2562</b>	<b>0.3926</b>	<b>0.4729</b>	0.1553	0.1564	<b>0.2401</b>	<b>0.3269</b>	<b>0.0739</b>	<b>0.0792</b>
GODEN <sub>DP</sub>	0.5328	0.6275	0.2327	0.2340	0.3822	0.4575	0.1467	0.1478	0.2042	0.2873	0.0708	0.0719
GODEN <sub>SC</sub>	0.5161	0.6202	0.2209	0.2224	0.3545	0.4340	0.1393	0.1404	0.1957	0.2911	0.0716	0.0730
GODEN <sub>CA</sub>	0.5254	0.6403	0.1822	0.1839	0.3921	0.4712	<b>0.1600</b>	<b>0.1612</b>	0.2352	0.3137	0.0752	0.0764
GODEN <sub>ODE</sub>	0.5176	0.6280	0.2001	0.2016	0.3434	0.4206	0.1211	0.1222	0.2089	0.2849	0.0689	0.070
GODEN <sub>EdgeODE</sub>	0.5305	0.6244	0.2480	0.2494	0.3755	0.4527	0.1480	0.1491	0.2112	0.2888	0.0728	0.0749
GODEN <sub>UC</sub>	0.5366	0.6139	0.2405	0.2506	0.3440	0.4234	0.1195	0.1206	0.2065	0.2888	0.0703	0.0715

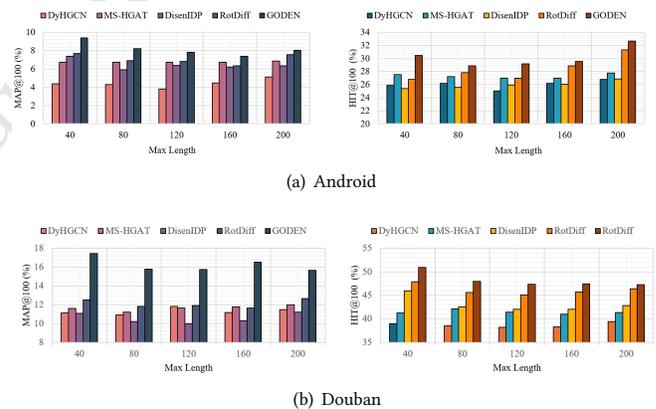
suffers a significant performance drop in HIT scores, which suggests that it is important to consider specific diffusion patterns in the cascade to retrieve potential users. (2) Compared with GODEN, the performance of variants GODEN<sub>DP</sub> and GODEN<sub>SC</sub> shows that removing any type of user encoding mechanism would lead to performance degradation. These results indicate that users' dynamic preferences and static correlations are both key factors affecting the diffusion process. (3) When we remove the ODE function on edges(GODEN<sub>EdgeODE</sub>), the model suffers from certain drops. This result verifies that the hidden dynamic of users in the diffusion process is influenced by their relations. When we remove the ODE solver and estimate diffusion dynamics with only neural networks(GODEN<sub>ODE</sub>), the model shows worse performance, highlighting the importance of estimating hidden dynamics with ODE functions instead of discrete neural networks.

## 5.4 Parameter Analyze Results

In this subsection, we conduct comparative experiments on the Douban and Android datasets and further analyze the effect of maximum cascade length. The result in Figure 3 shows that our model could outperform other models in any cascade length, illustrating its stability and effectiveness. We contribute its remarkable performance to our coupled ODE module, which could model the hidden dynamics of the diffusion process and infer users' dynamic preferences accurately regardless of cascade length and duration.

## 6 CONCLUSION

In this work, we propose a novel graph ordinary differential equation network (GODEN) for information diffusion prediction, which

**Figure 3: Impact of maximum cascade length.**

models the continuous dynamics of the diffusion process. With the coupled ODEs to characterize the co-evolution dynamics of users and their relations, GODEN could accurately infer users' dynamic preferences. Moreover, we extract static user correlation from the heterogeneous graph to complement users' dynamic preferences. To predict the future infection probability, we first represent the diffusion pattern in the observed cascade based on its temporal and user contexts. Then, we leverage a multi-head attention mechanism to attend to different contexts. The experimental results demonstrate the effectiveness of GODEN.

## REFERENCES

- [1] 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, San Francisco, CA, USA, February 22-25, 2016*, Paul N. Bennett, Vanja Josifovski, Jennifer Neville, and Filip Radlinski (Eds.). ACM, 573–582.
- [2] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. 2017. DeepHawkes: Bridging the Gap between Prediction and Understanding of Information Cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, Ee-Peng Lim, Marianne Winslett, Mark Sanderson, Ada Wai-Chee Fu, Jimeng Sun, J. Shane Culpepper, Eric Lo, Joyce C. Ho, Debora Donato, Rakesh Agrawal, Yu Zheng, Carlos Castillo, Aixin Sun, Vincent S. Tseng, and Chenliang Li (Eds.). ACM, 1149–1158. <https://doi.org/10.1145/3132847.3132973>
- [3] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 6572–6583. <https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html>
- [4] Zeyuan Chen, Wei Zhang, Junchi Yan, Gang Wang, and Jianyong Wang. 2021. Learning Dual Dynamic Representations on Time-Sliced User-Item Interaction Graphs for Sequential Recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 231–240. <https://doi.org/10.1145/3459637.3482443>
- [5] Zhangtao Cheng, Wenxue Ye, Leyuan Liu, Wenxin Tai, and Fan Zhou. 2023. Enhancing Information Diffusion Prediction with Self-Supervised Disentangled User and Cascade Representations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, 3808–3812. <https://doi.org/10.1145/3583780.3615230>
- [6] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 364–373. <https://doi.org/10.1145/3447548.3467430>
- [7] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010 (JMLR Proceedings, Vol. 9)*, Yee Whye Teh and D. Mike Titterton (Eds.). JMLR.org, 249–256.
- [8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
- [9] Nathan Hodas and Kristina Lerman. 2014. The Simple Rules of Social Contagion. *Scientific reports* 4 (2014).
- [10] Mohammad Raihanul Islam, Sathappan Muthiah, Bijaya Adhikari, B. Aditya Prakash, and Naren Ramakrishnan. 2018. DeepDiffuse: Predicting the 'Who' and 'When' in Cascades. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 1055–1060.
- [11] David Kempe, Jon M. 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, Washington, DC, USA, August 24 - 27, 2003*, Lise Getoor, Ted E. Senator, Pedro M. Domingos, and Christos Faloutsos (Eds.). ACM, 137–146.
- [12] Jure Leskovec, Lars Backstrom, and Jon M. 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, Paris, France, June 28 - July 1, 2009*. ACM, 497–506.
- [13] Yong Liu, Susen Yang, Yinan Zhang, Chunyan Miao, Zaiqing Nie, and Juyong Zhang. 2023. Learning Hierarchical Review Graph Representations for Recommendation. *IEEE Trans. Knowl. Data Eng.* 35, 1 (2023), 658–671. <https://doi.org/10.1109/TKDE.2021.3075052>
- [14] Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. 2019. Graph Neural Ordinary Differential Equations. *CoRR abs/1911.07532* (2019). <http://arxiv.org/abs/1911.07532>
- [15] Hongliang Qiao, Shanshan Feng, Xutao Li, Huiwei Lin, Han Fu, 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 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, 2065–2074. <https://doi.org/10.1145/3583780.3615041>
- [16] Yifang Qin, Wei Ju, Hongjun Wu, Xiao Luo, and Ming Zhang. 2024. Learning Graph ODE for Continuous-Time Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2024), 1–14. <https://doi.org/10.1109/TKDE.2024.3349397>
- [17] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 519–527. <https://doi.org/10.1145/3336191.3371845>
- [18] 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 *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 510–518. <https://doi.org/10.1145/3336191.3371811>
- [19] 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 *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 4156–4164. <https://ojs.aaai.org/index.php/AAAI/article/view/20334>
- [20] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjit Biswal, and Hongyuan Zha. 2019. DyRep: Learning Representations over Dynamic Graphs. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. <https://openreview.net/forum?id=HyePrhR5KX>
- [21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008.
- [22] Ding Wang, Lingwei Wei, Chunyuan Yuan, Yinan Bao, Wei Zhou, Xian Zhu, and Songlin Hu. 2022. Cascade-Enhanced Graph Convolutional Network for Information Diffusion Prediction. In *Database Systems for Advanced Applications - 27th International Conference, DASFAA 2022, Virtual Event, April 11-14, 2022, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 13245)*, Arnab Bhattacharya, Janice Lee, Mong Li, Divyakant Agrawal, P. Krishna Reddy, Mukesh K. Mohania, Anirban Mondal, Vikram Goyal, and Rage Uday Kiran (Eds.). Springer, 615–631. [https://doi.org/10.1007/978-3-031-00123-9\\_50](https://doi.org/10.1007/978-3-031-00123-9_50)
- [23] 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 2017, New Orleans, LA, USA, November 18-21, 2017*, Vijay Raghavan, Srinivas Aluru, George Karypis, Lucio Miele, and Xindong Wu (Eds.). IEEE Computer Society, 475–484.
- [24] Ruijie Wang, Zijie Huang, Shengzhong Liu, Huajie Shao, Dongxin Liu, Jinyang Li, Tianshi Wang, Dachun Sun, Shuochoao Yao, and Tarek F. Abdelzaher. 2021. DyDiff-VAE: A Dynamic Variational Framework for Information Diffusion Prediction. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 163–172. <https://doi.org/10.1145/3404835.3462934>
- [25] Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, and Zhenyu Guo. 2021. APAN: Asynchronous Propagation Attention Network for Real-time Temporal Graph Embedding. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*, Guoliang Li, Zhanhui Li, Stratos Idreos, and Divesh Srivastava (Eds.). ACM, 2628–2638. <https://doi.org/10.1145/3448016.3457564>
- [26] Zhitao Wang, Chengyao Chen, and Wenjie Li. 2018. Attention Network for Information Diffusion Prediction. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 65–66. <https://doi.org/10.1145/3184558.3186931>
- [27] 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, CIKM 2018, Torino, Italy, October 22-26, 2018*. ACM, 1795–1798.
- [28] Zhitao Wang and Wenjie Li. 2019. Hierarchical Diffusion Attention Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 3828–3834.
- [29] Lingwei Wei, Dou Hu, Wei Zhou, Zhaojuan Yue, and Songlin Hu. 2021. Towards Propagation Uncertainty: Edge-enhanced Bayesian Graph Convolutional Networks for Rumor Detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and

1045	Roberto Navigli (Eds.). Association for Computational Linguistics, 3845–3854. <a href="https://doi.org/10.18653/v1/2021.acl-long.297">https://doi.org/10.18653/v1/2021.acl-long.297</a>	4033–4039.	1103
1046	[30] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement Knowledge Graph Reasoning for Explainable Recommendation. In <i>Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019</i> , Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 285–294.	[34] Yang Yang, Jie Tang, Cane Wing-ki Leung, Yizhou Sun, Qicong Chen, Juanzi Li, and Qiang Yang. 2015. RAIN: Social Role-Aware Information Diffusion. In <i>Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA</i> , Blai Bonet and Sven Koenig (Eds.). AAAI Press, 367–373. <a href="https://doi.org/10.1609/aaai.v29i1.9164">https://doi.org/10.1609/aaai.v29i1.9164</a>	1104
1047		[35] 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. 12459 (2020), 347–363.	1105
1048		[36] Yuan Zhang, Tianshu Lyu, and Yan Zhang. 2018. COSINE: Community-Preserving Social Network Embedding From Information Diffusion Cascades. In <i>Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018</i> , Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 2620–2627.	1106
1049	[31] Shaokai Xu, Lihua Zhou, Jianfeng Xu, Lizhen Wang, and Hongmei Chen. 2022. MSIDP: Multi-scale Information Diffusion Prediction with Timestamp Information and Wide Dispersion. In <i>International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022</i> . IEEE, 1–10. <a href="https://doi.org/10.1109/IJCNN55064.2022.9892786">https://doi.org/10.1109/IJCNN55064.2022.9892786</a>	[37] Fan Zhou, Xovee Xu, Goce Trajcevski, and Kunpeng Zhang. 2022. A Survey of Information Cascade Analysis: Models, Predictions, and Recent Advances. <i>ACM Comput. Surv.</i> 54, 2 (2022), 27:1–27:36. <a href="https://doi.org/10.1145/3433000">https://doi.org/10.1145/3433000</a>	1107
1050			1108
1051			1109
1052			1110
1053	[32] Cheng Yang, Maosong Sun, Haoran Liu, Shiyi Han, Zhiyuan Liu, and Huanbo Luan. 2021. Neural Diffusion Model for Microscopic Cascade Study. <i>IEEE Trans. Knowl. Data Eng.</i> 33, 3 (2021), 1128–1139.		1111
1054			1112
1055			1113
1056			1114
1057	[33] Cheng Yang, Jian Tang, Maosong Sun, Ganqu Cui, and Zhiyuan Liu. 2019. Multi-scale Information Diffusion Prediction with Reinforced Recurrent Networks. In <i>Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019</i> , Sarit Kraus (Ed.). ijcai.org,		1115
1058			1116
1059			1117
1060			1118
1061			1119
1062			1120
1063			1121
1064			1122
1065			1123
1066			1124
1067			1125
1068			1126
1069			1127
1070			1128
1071			1129
1072			1130
1073			1131
1074			1132
1075			1133
1076			1134
1077			1135
1078			1136
1079			1137
1080			1138
1081			1139
1082			1140
1083			1141
1084			1142
1085			1143
1086			1144
1087			1145
1088			1146
1089			1147
1090			1148
1091			1149
1092			1150
1093			1151
1094			1152
1095			1153
1096			1154
1097			1155
1098			1156
1099			1157
1100			1158
1101			1159
1102			1160