Continuous Dynamic Modeling via Neural ODEs for Popularity Trajectory Prediction

Songbo Yang, Ziwei Zhao, Zihang Chen, Haotian Zhang, Tong Xu, and Mengxiao Zhu (✉)

University of Science and Technology of China,Hefei, China
{songboyang,zzw22222,czh1999,sosweetzhang}@mail.ustc.edu.cn
{ tongxu,mxzhu}@ustc.edu.cn

Abstract. Popularity prediction for information cascades is a fundamental and challenging task. While most existing methods consider this as a discrete problem, we argue that popularity trajectory prediction is more practical, as it aims to forecast the entire continuous trajectory of how popularity unfolds over arbitrary future time. However, traditional methods for popularity trajectory prediction primarily rely on specific diffusion mechanism assumptions, which may not align well with real-world dynamics, limiting their performance. To address this limitation, we propose NODEPT, a novel approach based on neural ordinary differential equations (ODEs) for popularity trajectory prediction. We first employ an encoder to initialize the latent state representations that capture the co-evolution structural characteristics and temporal patterns of cascades. More importantly, we then introduce an ODE-based generative module that learns the dynamics of the diffusion system in the latent space. Finally, a decoder transforms the latent state into the prediction of the future popularity trajectory. Our experimental results on three real-world datasets demonstrate the superiority and rationality of the proposed NODEPT method.

Keywords: Popularity Trajectory Prediction · Diffusion System · Neural ODEs.

1 Introduction

With the growing popularity of social media, more users are sharing and interacting online, fundamentally changing how people entertain and communicate. Consequently, accurately predicting the popularity of information cascades on social media is essential for various applications, including viral marketing [6], assessing scientific impact [8], and item recommendation [10, 17].

Early research on popularity prediction relied on diffusion hypotheses or manually engineered features [4, 14]. With advances in deep learning, recurrent neural networks (RNNs) were introduced to model cascade structures and temporal dynamics [1]. However, RNNs struggle to fully exploit structural information, prompting the adoption of graph neural networks (GNNs) [12, 7, 5, 15] to capture comprehensively learning cascade structures.

Despite progress, most models frame popularity prediction as a single-point forecasting task, predicting popularity at a fixed time Δt [7, 12, 5]. Predicting multiple timestamps requires retraining [13], increasing computational costs. We argue that popularity trajectory prediction, which forecasts an entire trajectory over arbitrary time points, provides richer insights into both instantaneous popularity and underlying dynamics. For instance, as shown in Figure 1, although cascade c_1 initially has higher popularity than c_2 and c_3, its growth rate slows over time, eventually leading to similar popularity levels across cascades. Understanding these dynamics enables better interventions, such as amplifying positive content or mitigating misinformation. Traditional trajectory prediction models often rely on predefined diffusion assumptions, such as Hawkes processes [13, 14], which may not fully align well with real-world dynamics. Therefore, a data-driven neural network approach is essential. By learning complex patterns directly from data, such methods capture intricate dynamics that traditional approaches may overlook, leading to more accurate predictions.

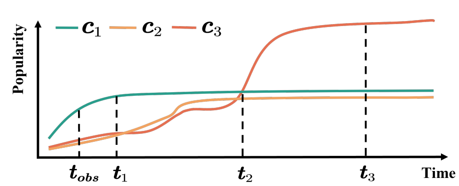

Fig. 1. Popularity trajectories of different information cascades.

To address the above limitation, we propose a novel neural ordinary differential equations (ODEs)-based method for popularity trajectory prediction, named NODEPT. NODEPT first employs a dynamic graph representation module to capture cascade co-evolution and a temporal sequence module to learn temporal patterns. These representations are fused to infer the initial latent state of the cascade. More importantly, we then propose an ODE-based generative module to capture the evolution of the diffusion system. The ODE function incorporates the latent state of cascades and their interaction representation. Finally, a decoder transforms the latent continuous trajectory produced by the ODE generative module into the future popularity trajectory.

Our main contributions can be summarized as follows. (1) Different from traditional methods, we explore a continuous data-driven network approach without predefined diffusion mechanisms, which captures intricate characteristics that might missed by traditional methods, offering more accurate predictions for continuous popularity trajectory. (2) To fully leverage neural networks for extracting latent information from data, we propose NODEPT, a novel framework based on neural ODEs for continuous popularity trajectory prediction. (3) Extensive experiments on multiple real-world datasets demonstrate the effectiveness of NODEPT in popularity trajectory prediction, validating its superior performance compared to traditional methods.

2 Preliminaries

Cascade. Given a set of users $\mathcal{U}$, a cascade c records the diffusion process of a information among the users $\mathcal{U}$. Specifically, we use a chronological sequence $g^c(t) = \{(u_j^c, v_j^c, t_j^c)\}_{j=1,\ldots,|g^c(t)|}$ to represent the growth process of cascade c until time t, where (u_j^c, v_j^c, t_j^c) indicates that u_j^c shared information to v_j^c at time t_j^c (or equivalently, that user v_j^c participates in cascade c due to user u_j^c at t_j).

Diffusion Graph. A diffusion graph is defined as a chronological sequence of diffusion behavior $\mathcal{G}^t = \{(u_j, v_j, c_j, t_j)|t_j < t\}$ to denote the diffusion process of all cascades until t, where (u_j, v_j, c_j, t_j) represents user v_j participates in cascade c_j due to user u_j at t_j.

Popularity Trajectory Prediction. Given a cascade c begins at t_0^c, after observing it up to time t_{obs}^c , our goal is to predict the trajectory of incremental popularity $\hat{P}_c^t = |g^c(t)| - |g^c(t_{obs}^c)|$ of c in the future, where $t > t_{obs}$. In particular, we learn a function $f : \mathcal{G}^{t_{obs}^c} \to \hat{P}_c^t$.

3 Method

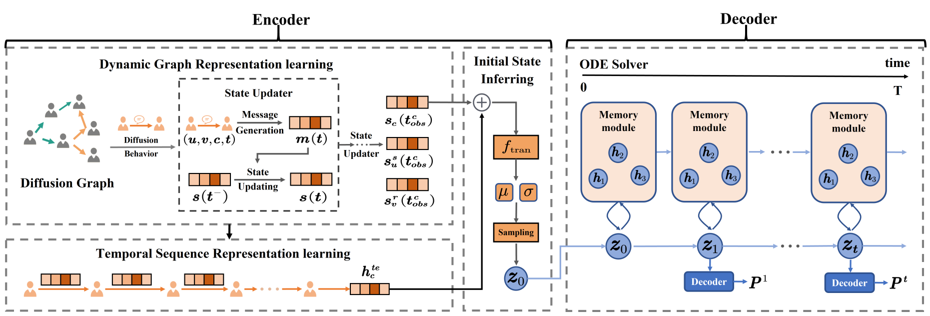

Fig. 2. Framework of our proposed NODEPT.

The overall NODEPT framework is illustrated in Figure 2. Following the variational autoencoder (VAE) paradigm, NODEPT contains the following modules.

3.1 Initial State Encoder

Dynamic Graph Representation Learning In this module, we dynamically learn the diffusion behaviors within the diffusion graph $\mathcal{G}^{t_{obs}^c}$ as they occur. To model the evolutionary patterns of users and cascades, we maintain dynamic states for both. For each user u, we maintain two types of state $\boldsymbol{s}_u^s(t)$ and $\boldsymbol{s}_u^r(t)$ that represent their roles as senders and receivers in the diffusion process. Furthermore, for each cascade c, we maintain a dynamic cascade state $\boldsymbol{s}_c(t)$.

To effectively integrate information from previous states and current diffusion behavior, we first encode the diffusion behavior (u, v, c, t) into message representations $\boldsymbol{m}_u(t)$, $\boldsymbol{m}_v(t)$, and $\boldsymbol{m}_c(t)$ for the users and cascade, respectively. Taking the cascade state $\boldsymbol{s}_c(t)$ as an example, let $\boldsymbol{s}_c(t^-)$ denote the cascade state just before time t. We generate the message representation for cascade c using the following mechanism:

$$\boldsymbol{m}_c(t) = \sigma(\boldsymbol{W}^m[\boldsymbol{s}_u^s(t^-)||\boldsymbol{s}_v^r(t^-)||\boldsymbol{s}_c(t^-)||\boldsymbol{f}_c^t] + \boldsymbol{b}^m), \tag{1}$$

where $||$ is the concatenation operation and $\boldsymbol{W}^m \in \mathbb{R}^{d\times 3d}, \boldsymbol{b}^m \in \mathbb{R}^d$ are trainable parameters, $\boldsymbol{f}_c^t$ denotes the temporal feature generated from timestamp t . Inspired by [11, 16], $\boldsymbol{f}_c^t$ takes the following form,

$$\boldsymbol{f}_c^t = [\cos w_1^r \Delta t_c, \cos w_2^r \Delta t_c, ..., \cos w_n^r \Delta t_c], \tag{2}$$

where Δt_c is the time interval since the last updating of cascade c (i.e., $\Delta t_c = t - t_c^-$ and t_c^- is the last time when c was updated), $\boldsymbol{w} = [w_1, ..., w_n] \in \mathbb{R}^d$ are trainable parameters. After generating message representations, we fuse it with the previous cascade state $\boldsymbol{s}_c(t^-)$ using a Gated Recurrent Unit (GRU) to capture the diffusion history and evolutionary patterns as follows.

$$\boldsymbol{s}_c(t) = GRU\left(\boldsymbol{s}_c(t^-), \boldsymbol{m}_c(t)\right). \tag{3}$$

The strengths of GRU in capturing long-range temporal dependencies make it well-suited for this continuous state evolution modeling.

The user sender $\boldsymbol{s}_u^s(t)$ and receiver $\boldsymbol{s}_u^r(t)$ states are updated similarly, fusing their respective messages $\boldsymbol{m}_u(t)$ and $\boldsymbol{m}_v(t)$ with previous states using GRU.

Temporal Sequence Representation Learning In this module, we learn the temporal patterns of a cascade by aggregating user dynamic states over time based on the diffusion sequence. Given a cascade c, we first organize cascade c as a diffusion sequence of user-time pairs $\{(u_1^c, t_1^c), (u_2^c, t_2^c), ..., (u_n^c, t_n^c)\}$, where u_j^c is the user and t_j^c is the time when user u_j^c shared the information and participate in the cascade c. To enhance the model's ability to capture the positional information of user shares, for each user u_i^c in the sequence, we compute their temporal user embedding $\boldsymbol{h}_i^c$ by first adding position embedding to the user' dynamic state representation:

$$\boldsymbol{h}_i^c = \boldsymbol{s}_u^i + \boldsymbol{e}_i^p, \tag{4}$$

where $\boldsymbol{s}_u^i$ is the dynamic state $\boldsymbol{s}_u^s(t_{obs}^c)$ of user u_i^c and $\boldsymbol{e}_i^p$ is position embedding.

We then feed the sequence of temporal user embeddings $\{\boldsymbol{h}_1^c,\ \boldsymbol{h}_2^c, ..., \boldsymbol{h}_n^c\}$ into a Long Short-Term Memory (LSTM) network to produce the temporal representation $\boldsymbol{h}_c^{te}$ of cascade c.

$$\boldsymbol{h}_c^{te} = LSTM([\boldsymbol{h}_1^c, \boldsymbol{h}_2^c, ..., \boldsymbol{h}_n^c]). \tag{5}$$

Initial State Inferring Whenever the observed time t^c_{obs} of cascade c is reached, we first compute the dynamic representation of the cascade. To capture the characteristic of the publishing time t^c_0 of the cascade, we then divide the overall observation period $[0, T_{obs}]$ into N equal time slots. For each time interval $[k\frac{T_{obs}}{N}, (k+1)\frac{T_{obs}}{N})$, we have a learnable embedding $\boldsymbol{e}^{dy}_k$ to distinguish cascade published in different time intervals. Next, we obtain the dynamic representation of the cascade as follows.

$$\boldsymbol{h}^{dy}_c = \boldsymbol{s}_c(t^c_{obs}) + \boldsymbol{e}^{dy}_{[t^c_0]}, \tag{6}$$

where $\boldsymbol{s}_c(t^c_{obs})$ is the dynamic state of cascade c at the observation time t^c_{obs}, and $\boldsymbol{e}^{dy}_{[t^c_0]}$ is time slot embedding that encodes the publishing time t^c_0 of the cascade.

The dynamic representation $\boldsymbol{h}^{dy}_c$ and temporal representation $\boldsymbol{h}^{te}_c$ explore the rich cascade information from different perspectives. To generate the overall cascade representation $\boldsymbol{h}_c$ for initializing the latent state, we concatenate these two representations and then feed them through a multi-layer perceptron (MLP):

$$\boldsymbol{h}_c = \sigma\left(\boldsymbol{W^i}\left[\boldsymbol{h}^{dy}_c || \boldsymbol{h}^{te}_c\right]\right), \tag{7}$$

where $\boldsymbol{W^i} \in \mathbb{R}^{d\times 2d}$ are learnable parameters and $\sigma(\cdot)$ is a non-linear activation function to provide non-linearity.

Afterward, we utilize $\boldsymbol{h}_c$ to infer the approximate posterior distribution $q_\phi\left(\boldsymbol{z}^0_c \mid \mathcal{G}^{t^c_{obs}}\right)$ from which we sample the initial latent state $\boldsymbol{z}^0_c$ for the ODE generative model.

$$\begin{aligned} \boldsymbol{\mu}_{z^0_c}, \boldsymbol{\sigma}_{z^0_c} &= f_{\text{trans}}\left(\boldsymbol{h}_c\right), \\ q_\phi\left(\boldsymbol{z}^0_c \mid \mathcal{G}^{t^c_{obs}}\right) &= \mathcal{N}\left(\boldsymbol{\mu}_{z^0_c}, \boldsymbol{\sigma}_{z^0_c}\right), \end{aligned} \tag{8}$$

where $\mathcal{N}$ denotes a normal distribution, f_{trans} is a simple MLP. We then employ the reparametrization trick to sample from the posterior distribution $\boldsymbol{z}^0_c$.

$$\boldsymbol{z}^0_c \sim p(\boldsymbol{z}^0_c) \approx q_\phi\left(\boldsymbol{z}^0_c | \mathcal{G}^{t^c_{obs}}\right). \tag{9}$$

Here $p(z_0)$ is a prior distribution used as a regularization. We minimize the KL divergence between $q_\phi\left(\boldsymbol{z}^0_c | \mathcal{G}^{t^c_{obs}}\right)$ and the prior distribution $p(\boldsymbol{z}_0)$ during training.

3.2 ODE Generative Module and Decoder

ODE Generative Module. After establishing the initial state encoder, we now define the neural ODE function that drives the diffusion system forward. The latent state of each cascade is influenced by two key factors: (1) its self-evolution dynamics and (2) interactions with other cascades. Thus, our ODE function comprises two components—one modeling the cascade's intrinsic evolution and the other capturing external interactions.

In real-world diffusion, new information and behaviors continuously emerge, making it challenging to identify relevant interacting cascades. Since influence decays over time, older cascades are unlikely to interact with the current one. Thus, we propose an external memory module to store representations of the most recent historical cascades up to t_{obs}^c. This module maintains a memory matrix $\mathbf{M}_c \in \mathbb{R}^{N_m \times d}$, where N_m is the memory module size and d is the representation dimension. We consider the cascades in $\mathbf{M}_c$ as potential interactors with the current cascade c and dynamically update the memory module as new information emerges. To model these interactions effectively, we employ an attention mechanism that adaptively aggregates relevant information from $\mathbf{M}_c$. In formulation, we have:

$$\begin{aligned} \mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \mathrm{softmax}\left(\mathbf{Q}\mathbf{K}^T\right)\mathbf{V}, \\ h_c^{t,in} &= \mathrm{Att}\left(\boldsymbol{z}_c^t W^Q, \mathbf{M}_c W^K, \mathbf{M}_c W^V\right), \end{aligned} \tag{10}$$

where z_c^t is the latent state of cascades, $W^Q \in \mathbb{R}^{d \times 2d}, W^K \in \mathbb{R}^{d \times 2d}, W^V \in \mathbb{R}^{d \times d}$ are trainable parameters and $h_c^{t,in}$ is the interaction representation of cascade c with other cascades. This mechanism can be viewed as a function $f_a \colon (\boldsymbol{z}_c^t, \mathbf{M}_c) \rightarrow h_c^{t,in}$ that maps the latent state $\boldsymbol{z}_c^t$ of cascade c and the matrix $\mathbf{M}_c$ to their interaction representation.

Furthermore, we define the overall neural ODE function as follows:

$$\begin{aligned} \boldsymbol{h}_c^{t,in} &= f_a\left(\boldsymbol{z}_c^t, \mathbf{M}_c\right), \\ \frac{d\boldsymbol{z}_c^t}{dt} &= f_g\left(\boldsymbol{z}_c^t, \mathbf{M}_c\right) = \sigma\left(\boldsymbol{W}^{\boldsymbol{f}}\left[\boldsymbol{h}_c^{t,in} \| \boldsymbol{z}_c^t\right]\right), \end{aligned} \tag{11}$$

where $\boldsymbol{W}^{\boldsymbol{f}} \in \mathbb{R}^{d \times d} \in$ is trainable parameters. Here, $\boldsymbol{z}_c^t$ models the self-evolution of c itself, while $\boldsymbol{h}_c^{t,in}$ captures the interaction between c and other cascades.

Decoder Given the ODE function f_g and the initial state $\boldsymbol{z}_c^0$ of cascade c, the popularity trajectory of c is determined, which can be solved via a black-box ODE solver. Finally, a decoder generates the predicted trajectory based on the decoding probability $p(P_c^t|\boldsymbol{z}_c^t)$ computed from the decoding function f_{dec} as shown below.

$$\hat{P}_c^t \sim p(P_c^t|\boldsymbol{z}_c^t) = f_{\mathrm{dec}}(\boldsymbol{z}_c^t). \tag{12}$$

We implement f_{dec} as a two-layer MLP with nonlinear activation. It outputs the mean of the normal distribution $p(P_c^t|\boldsymbol{z}_c^t)$, which we treat as the predicted popularity value for cascade c at time t.

3.3 Training

Joint training The standard VAE framework is trained to maximize the evidence lower bound (ELBO). We first derived the ELBO loss for NODEPT as shown in Equation 13. The first term is the reconstruction loss for the predicted trajectory and the second term is the KL divergence regularizing the inferred posteriors towards a prior $p(\boldsymbol{z}_0^c)$.

$$\mathcal{L}_{ELBO(\theta,\phi)} = \frac{1}{N}\sum_{c}\mathbb{E}[\log p_\theta\left(P_c^{t^c_{obs}:\,t^c_{pre}}\right)] - \mathrm{KL}\left[q_\phi\left(\boldsymbol{z}_c^0|\mathcal{G}^{t^c_{obs}}\right)\|p\left(\boldsymbol{z}_c^0\right)\right]. \tag{13}$$

The reconstruction loss is estimated by $-\sum_t \frac{\|P_c^t - \hat{P}_c^t\|^2}{2\sigma_2^2}$ where the constant σ is the standard derivation of each prior distribution.

Since the reconstruction loss only constrains popularity at discrete time points, we argue that an effective prediction should also maintain realistic increments between these points, reflecting the true growth pattern of the trajectory. To achieve this, we introduce an increment loss, defined in Equation 14, which quantifies the discrepancy between the increments of adjacent time points in the true and predicted trajectories.

$$\mathcal{PI} = \hat{P_c^{t+1}} - \hat{P}_c^t, \mathcal{TI} = P_c^{t+1} - P_c^t, \mathcal{IL} = \frac{1}{N}\sum_c\sum_t |\mathcal{PI} - \mathcal{TI}|. \tag{14}$$

Finally, we combined the two losses mentioned above to obtain the overall model loss and jointly trained the encoder, ODE generative model, and decoder in an end-to-end manner.

$$\mathcal{L} = \lambda_1\mathcal{IL} - \mathcal{L}_{ELBO(\theta,\phi)}. \tag{15}$$

4 Experiments

4.1 Experimental Settings

Datasets. The experiments are conducted on three real-world datasets, including Twitter [9], Weibo [1], and APS [1] that have been commonly used in previous related works [7, 12, 5] for evaluating cascade popularity prediction.

Baselines. To assess the effectiveness of our method in single-point popularity prediction task, we compare our model with a variety of baselines, including **Xgboost** [2], **DeepHawkes** [1], **CasCN** [3], **CasFlow** [12], **CTCP** [7], **CasDO** [5]. Meanwhile, We compare NODEPT with **SEISMIC** [14] and **CASPER** [13] for the popularity trajectory prediction task.

Implementation Details. We randomly divide the cascades into three sets: 70% for training, 15% for validation, and 15% for testing purposes. We set the observation length of a cascade to 2 days, 2 hours, and 2 years on Twitter, Weibo, and APS. Correspondingly, the prediction length of a cascade is set to 13 days, 13 hours, and 13 years for Twitter, Weibo, and APS datasets, respectively. We evaluate model performance using Mean Squared Logarithmic Error (MSLE) and Mean Absolute Percentage Error (MAPE).

[1] https://journals.aps.org/datasets

Table 1. Model Performance of single-point prediction On Twitter, Weibo, APS datasets in terms of MSLE, MAPE.

Model	Twitter				Weibo				APS			
	5 Days		15 Days		5 Hours		15 Hours		5 Years		15 Years	
	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
Xgboost	8.3209	0.7719	10.1375	0.8478	3.1755	0.3261	4.0548	0.4139	2.2523	0.3314	4.2351	0.5863
CasODE	2.4529	0.2637	4.1891	0.3484	1.7659	0.2434	2.1098	0.2349	1.0852	0.2267	1.7357	0.2934
DeepHawkes	3.6782	0.3511	5.3521	0.4719	2.4531	0.2934	3.3905	0.3591	2.1412	0.2919	2.8276	0.3199
CasCN	2.8390	0.2914	4.2041	0.3397	2.1431	0.2852	2.8417	0.3038	1.7607	0.3076	2.6881	0.3593
CasFlow	2.6368	0.2630	4.3256	0.3432	1.8548	0.2458	2.1132	0.2372	0.9726	0.2201	1.7537	0.2837
CTCP	2.3624	0.2799	3.8968	0.3593	1.9860	0.2703	2.3111	0.2830	1.2864	0.2556	1.9875	0.2760
CasDO	2.2258	0.2673	3.6911	0.3187	1.7286	0.2374	**2.0549**	**0.2317**	1.0286	0.2219	1.6935	0.2338
NODEPT	**1.7629**	**0.2635**	**3.2276**	**0.2976**	**1.6573**	**0.2251**	2.2842	0.2576	**0.9274**	**0.2157**	**1.6792**	**0.2183**

Table 2. Trajectory prediction performance on all datasets

Model	Twitter		Weibo		APS	
	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
SEISMIC	4.4345	0.3818	4.8685	0.4053	3.9436	0.3538
CASPER	3.9408	0.3549	4.0468	0.3505	3.4609	0.3376
NODEPT	**2.3024**	**0.2645**	**1.9147**	**0.2575**	**1.1493**	**0.2291**

4.2 Performance Comparison

Single-point prediction To comprehensively verify the effectiveness of NODEPT in single-point prediction task, we assess the model's performance at two distinct time points for each dataset to gauge both short-term and long-term predictive capabilities separately. The results are shown in Table 1, from which the following conclusions can be drawn.

Firstly, NODEPT consistently outperforms other models on the Twitter and APS datasets, validating the effectiveness of our proposed method. Furthermore, NODEPT shows a significant improvement over the state-of-the-art baseline, CasDO, due to its superior ability to capture the dynamics of the underlying diffusion system. On the Weibo dataset, however, NODEPT's advantage is less pronounced compared to CasDO at the 15-hour mark. This discrepancy may be attributed to the relatively short observation period on Weibo, where user interests and the diffusion system undergo less significant evolution.

Secondly, CTCP and CasFlow achieved better performance compared to DeepHawkes and CasCN, indicating that graph-based methods can effectively capture the structural features of cascades and deliver superior results. Meanwhile, CadDO achieve relatively better performance by considering the temporal irregularity of cascade events.

Overall trajectory prediction Previous trajectory prediction models were evaluated using single-point prediction tasks [13]. To assess performance, we

Table 3. Ablation study results on all datasets

Model	Twitter		Weibo		APS	
	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
w/o IR	3.2887	0.2989	2.7946	0.2858	2.1846	0.2566
w/o IL	2.4741	0.2771	2.0982	0.2736	1.1287	2.2386
NODEPT	2.3024	0.2645	1.9147	0.2575	1.0657	0.2291

initially computed the model's MSLE and MAPE at all integer discrete points within the prediction interval and averaged these values. The results are presented in Table 2, leading to the following conclusions.

NODEPT consistently outperforms both SEISMIC and CASPER on MSLE and MAPE metrics, highlighting its superior ability in overall trajectory prediction. CASPER, a state-of-the-art baseline for traditional popularity trajectory prediction, assumes that the information diffusion process follows marked Hawkes point processes, but it fails to fully capture the underlying dynamics driving popularity growth. In contrast, NODEPT leverages neural ODEs to automatically learn complex patterns from data, resulting in excellent predictive performance.

4.3 Ablation Study

To investigate the effectiveness of the submodule of NODEPT, we compare it with the following variations. (1) **w/o IR** removes the interaction representation in the ODE generative model. (2) **w/o IL** removes the increment loss in the Equation 15. To thoroughly evaluate the effectiveness of each module, we computed the evaluation metrics for overall trajectory prediction. The results are presented in Table 3, with the observations detailed as follows. (1) The w/o IR variant underperforms NODEPT, suggesting that the interaction representation can explicitly capture intricate interactions between cascades through attention mechanism, which is important for tracking the evolution of the underly diffusion system. (2) NODEPT outperforms the w/o IL, indicating that the increment loss can effectively capture the growth pattern of the popularity trajectory.

5 Conclusion

In this paper, we propose a novel approach to model the continuous dynamics of diffusion system using neural ODEs for continuous popularity trajectory prediction. We first adopt an encoder which consists of two representation learning modules for initializing latent state representations of cascades. More importantly, we propose an ODE generative module to track the evolution of the underlying diffusion system which effectively models both the self-evolution of individual cascades as well as their intricate interactions with other cascades.

This method addresses the limitation of traditional approaches that rely on fixed diffusion mechanism assumptions, providing more accurate predictions. Extensive experiments on three real-world datasets demonstrate the effectiveness and rationality of our proposed NODEPT approach.

Acknowledgments. This work was supported by the Supercomputing Center, University of Science and Technology of China.

References

1. Cao, Q., Shen, H., Cen, K., Ouyang, W., Cheng, X.: DeepHawkes: Bridging the Gap between Prediction and Understanding of Information Cascades. In: CIKM. pp. 1149–1158 (2017)
2. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: KDD. pp. 785–794 (2016)
3. Chen, X., Zhou, F., Zhang, K., Trajcevski, T., Zhang, F.: Information Diffusion Prediction via Recurrent Cascades Convolution. In: ICDE. pp. 770–781 (2019)
4. Cheng, J., Adamic, L., Dow, P.A., Kleinberg, J.M., Leskovec, J.: Can cascades be predicted? In: WWW. pp. 925–936 (2014)
5. Cheng, Z., Zhou, F., Xu, X., Zhang, K., Trajcevski, G., Zhong, P.S.: Information cascade popularity prediction via probabilistic diffusion. TKDE pp. 1–14 (2024)
6. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. ACM Transactions on the Web pp. 5–es (2007)
7. Lu, X., Ji, S., Yu, L., Sun, L., Du, B., Zhu, T.: Continuous-time graph learning for cascade popularity prediction. In: IJCAI. pp. 2224–2232 (2023)
8. Wang, D., Song, C., Barabási, A.L.: Quantifying Long-Term Scientific Impact. Science pp. 127–132 (2013)
9. Weng, L., Menczer, F., Ahn, Y.Y.: Virality prediction and community structure in social networks. Scientific Reports (2013)
10. Wu, Q., Gao, Y., Gao, X., Weng, P., Chen, G.: Dual Sequential Prediction Models Linking Sequential Recommendation and Information Dissemination. In: KDD. pp. 447–457 (2019)
11. Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., Achan, K.: Inductive Representation Learning on Temporal Graphs (Feb 2020)
12. Xu, X., Zhou, F., Zhang, K., Liu, S., Trajcevski, G.: CasFlow: Exploring Hierarchical Structures and Propagation Uncertainty for Cascade Prediction. TKDE pp. 3484–3499 (Apr 2023)
13. Zhang, X., Aravamudan, A., Anagnostopoulos, G.C.: Anytime Information Cascade Popularity Prediction via Self-Exciting Processes. In: ICML (2022)
14. Zhao, Q., Erdogdu, M.A., He, H.Y., Rajaraman, A., Leskovec, J.: SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In: KDD. pp. 1513–1522 (2015)
15. Zhao, Z., Lin, F., Zhu, X., Zheng, Z., Xu, T., Shen, S., Li, X., Yin, Z., Chen, E.: Dynllm: When large language models meet dynamic graph recommendation. arXiv (2024)
16. Zhao, Z., Yang, Y., Yin, Z., Xu, T., Zhu, X., Lin, F., Li, X., Chen, E.: Adversarial attack and defense on discrete time dynamic graphs. TKDE (2024)
17. Zhao, Z., Zhu, X., Xu, T., Lizhiyu, A., Yu, Y., Li, X., Yin, Z., Chen, E.: Time-interval aware share recommendation via bi-directional continuous time dynamic graphs. In: SIGIR. pp. 822–831 (2023)