

CasFT: Future Trend Modeling for Information Popularity Prediction with Dynamic Cues-Driven Diffusion Models

Xin Jing¹, Yichen Jing¹, Yuhuan Lu¹, Bangchao Deng¹, Xueqin Chen², Dingqi Yang^{1*}

¹The State Key Laboratory of Internet of Things for Smart City, University of Macau, China

²Kash Institute of Electronics and Information Industry

{yc27431, mc35259, yc17462, yc37980, dingqiyang}@um.edu.mo, nedchen0728@gmail.com

Abstract

The rapid spread of diverse information on online social platforms has prompted both academia and industry to realize the importance of predicting content popularity, which could benefit a wide range of applications, such as recommendation systems and strategic decision-making. Recent works mainly focused on extracting spatiotemporal patterns inherent in the information diffusion process within a given observation period so as to predict its popularity over a future period of time. However, these works often overlook the future popularity trend, as future popularity could either increase exponentially or stagnate, introducing uncertainties to the prediction performance. Additionally, how to transfer the preceding-term dynamics learned from the observed diffusion process into future-term trends remains an unexplored challenge. Against this background, we propose CasFT, which leverages observed information Cascades and dynamic cues extracted via neural ODEs as conditions to guide the generation of Future popularity-increasing Trends through a diffusion model. These generated trends are then combined with the spatiotemporal patterns in the observed information cascade to make the final popularity prediction. Extensive experiments conducted on three real-world datasets demonstrate that CasFT significantly improves the prediction accuracy compared to state-of-the-art approaches.

Introduction

The advent of the digital age has led to the emergence of various online social platforms (OSNs). Users on OSNs can freely post and share their interests, which can then be followed or re-posted by others, resulting in the widespread dissemination of content in the form of information cascades (Cheng et al. 2014). In such an era of information explosion, understanding future trends in information — predicting its future popularity — can aid social management (Shen et al. 2014) and benefit various applications, including recommendation systems (Wang et al. 2021), fake news detection (Lazer et al. 2018), social event organization (Yu et al. 2015), and personalized user experiences (Li et al. 2014). Specifically, this prediction task, known as information popularity prediction, aims at forecasting the future increase in the popularity of given content, such as predicting

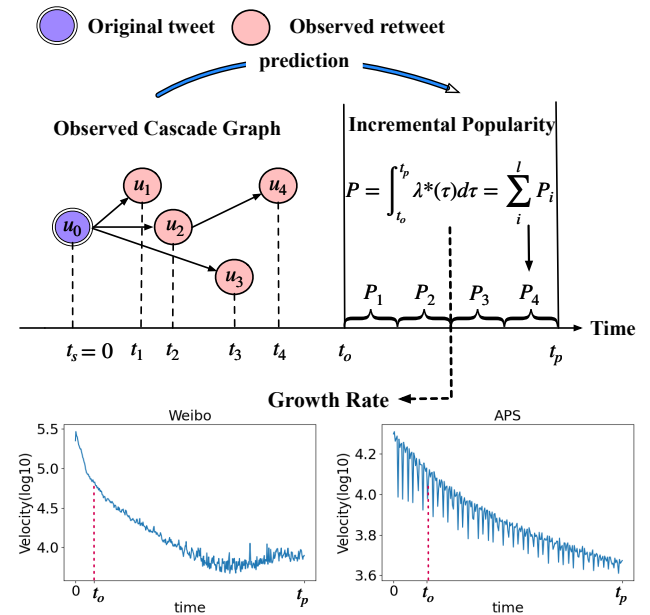


Figure 1: A toy example of information popularity prediction problem (top) and the variations of average growth rates after observation time in Weibo and APS datasets, respectively (bottom), where t_o represents the observation time and t_p represents the prediction time.

the number of retweets that may occur within a specified period in the future for a tweet.

State-of-the-art approaches mainly focus on capturing spatiotemporal patterns from observed cascades by adopting various deep learning techniques, such as Recurrent Neural Networks (RNNs) (Lu et al. 2023), attention mechanism (Yu et al. 2022) or Graph Neural Networks (GNNs) (Wang et al. 2022; Chen et al. 2019). However, existing approaches predominantly focus on modeling cascades before an observation time, neglecting their evolution of trends between the observation time and the prediction time, which is vital for popularity prediction. For instance, Figure 1 illustrates the average popularity growth rates between observation (t_o) and prediction (t_p) times in Twitter and APS datasets, where the growth rate represents the increase of popularity per unit of time. Notably, during this period, the

*Corresponding author

variation in growth rates fluctuates significantly across both two datasets, highlighting different cascade incremental patterns, which contributes to the uncertainty in depicting future diffusion trends. Subsequently, achieving robust popularity prediction necessitates accommodating such uncertainties by accounting for cascade evolution patterns after the observation period. Nevertheless, the actual dynamic changes in cascades during this period are invisible when performing real-time popularity prediction.

Against this background, we resort to generative models to simulate the cascade evolving patterns between the observation and prediction times for boosting popularity prediction performance. However, there exist two difficulties in characterizing the unique features in the patterns: 1) the incremental popularity equals the integral of the growth rate, which is highly fluctuated as evidenced by Figure 1; and 2) information diffusion is a complex propagation process that is susceptible to external factors, resulting in varying patterns of uncertainties that need to accommodate.

To address these challenges, we propose CasFT, a novel information popularity prediction technique that aims to capture the evolving patterns of information Cascades over time, in particular, the Future propagation Trend. CasFT leverages neural Ordinary Differential Equations (ODEs) (Chen et al. 2018) to model the growth rate based on corresponding graph structure and sequential event information under observation, propagate the growth rate from the observation time to the prediction time, and calculates the cumulative popularity during several periods by the integration of the growth rate. We take the historical information diffusion representation and the future dynamic cumulative popularity as conditions, adopt diffusion models to generate the future trend of information popularity, and fuse the cascade representation with the generated cues for prediction. In summary, our contributions are as follows:

- We revisit the existing cascade popularity prediction methods and identify the importance of modeling future trends of popularity after observation which contributes to the final prediction.
- We propose CasFT, which is designed to capture the evolving dynamic patterns of both the information cascades and growth rate with neural ODEs. We leverage diffusion models to generate future trends of popularity and concatenate them with dynamic cascade representations together for prediction.
- We conduct extensive experiments to evaluate CasFT on several real-world information cascade datasets. Results show that CasFT achieves superior performance compared to a sizeable collection of state-of-the-art baselines on the information popularity prediction task, yielding 2.2%-19.3% improvement over the best-performing baselines across different datasets.

Related Work

In the current literature, cascade popularity prediction techniques can be roughly classified into three categories.

Feature-based approaches: Feature-based approaches extract observed features manually which include informa-

tion about the content, such as tags (Ma, Sun, and Cong 2013), topic features (Martin et al. 2016), and graph structures (Gao, Ma, and Chen 2014; Wang et al. 2016), temporal features such as publication time (Petrovic, Osborne, and Lavrenko 2011; Wu et al. 2016), observation time (Cheng et al. 2014; Yang and Leskovec 2011), first participation time (Zaman, Fox, and Bradlow 2014), etc. However, these works heavily rely on the quality of hand-crafted features and thus have limited generalizability.

Generative-based approaches: These methods use probabilistic models to describe the propagation, including epidemic models (Matsubara et al. 2012), survival analysis (Lee, Moon, and Salamatian 2012), and various stochastic point processes, e.g., Poisson process (Bao et al. 2015), Hawkes process (Zhao et al. 2015), (Zhang, Aravamudan, and Anagnostopoulos 2022). However, the selection of underlying probabilistic models can lead to significant differences in performance.

Deep-learning-based approaches: The recent rapid development of deep learning has also spurred the emergence of numerous information cascade models based on deep neural networks. DeepHawkes (Cao et al. 2017) constructs an end-to-end network by Hawkes processes. DeepCas (Li et al. 2017) is the first cascade graph representation learning method, capturing both structural and temporal information. VaCas (Zhou et al. 2020a) and CasFlow (Xu et al. 2021) propose a hierarchical graph learning method using variational autoencoders or normalizing flows to model uncertainty in cascade graphs, and a bidirectional GRU to capture temporal dynamics. CasSeqGCN (Wang et al. 2022) aggregates cascade node representations using a dynamic routing mechanism. CTCP (Lu et al. 2023) integrates multiple cascades into a diffusion graph to further capture the correlation between cascades and users. However, existing methods only consider modeling cascades within observation. We argue that CasFT incorporates the future trends of popularity, thereby improving the prediction performance.

Preliminaries

In this section, we present the definitions of key concepts in our work and give a brief introduction to the concepts of neural ODEs and diffusion models.

Problem Definition

Suppose that a Twitter user u_0 posts a tweet I at time $t_s = 0$ (we set the start time to 0) and other users can interact with tweet I through various actions such as liking, retweeting, and commenting. Here we focus on the “retweet” behavior, while our work can also be generalized to other actions. Specifically, when a user u_1 retweets I posted by user u_0 at time t_1 , we can define a triplet (u_0, u_1, t_1) to present the diffusion of information I from user u_0 to user u_1 . Subsequently, a series of such triplets constitutes a retweet cascade \mathcal{C} . Specifically, within a given observation time t_o , the retweet cascade $\mathcal{C}(t_o)$ can be defined as the set of all relevant triplets, i.e., $\mathcal{C}(t_o) = \{(u_{i_1}, u_{i_2}, t_i)\}_{i \in N}$, where N represents the total number of triplets involved in the retweet process during the observation period and $t_i \leq t_o$.

Cascade Graph: Given a tweet I and an observation time t_o , the cascade graph can be defined as $G(t_o) = (\mathcal{V}_c, \mathcal{E}_c)$, where \mathcal{V}_c represents the set of users involved in $\mathcal{C}(t_o)$, and \mathcal{E}_c represents the set of edges, each denoting a retweet relationship between two users (nodes).

Cascade Sequence: Different from the cascade graph, the cascade sequence $S(t_o)$ for cascade $\mathcal{C}(t_o)$ is a collection of users, each arranged in chronological order based on their timestamps, i.e., $S(t_o) = \{u_0, u_1, \dots, u_N\}$.

Global Graph: Given all the retweet cascades under the observation time t_o , we define the global graph as $\mathcal{G} = \{G_I(t_o) | I \in \mathcal{I}\}$, where the edge in \mathcal{E}_g represents the node relationship from cascading, such as the follower/followee relationship in the social network.

Cascade Popularity Prediction: Given a cascade $\mathcal{C}(t_o)$ and a duration t_o , we predict its incremental popularity P over the time interval from t_o to t_p , where $t_p \gg t_o$, and t_p is the prediction time. P represents the number of triplets occurring during the time period (t_o, t_p) .

Neural Ordinary Differential Equations

A neural Ordinary Differential Equation (ODE) (Chen et al. 2018) describes the continuous-time evolution of variables. It represents a transformation of variables over time, where the initial state at time t_0 , denoted as $\mathbf{h}(t_0)$, is integrated forward using an ODE to determine the transformed state at any subsequent time t_i .

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t; \theta) \quad \text{where} \quad \mathbf{h}(0) = \mathbf{h}_0 \quad (1)$$

$$\mathbf{h}(t_i) = \mathbf{h}(t_0) + \int_{t_0}^{t_i} \frac{d\mathbf{h}(t)}{dt} dt, \quad (2)$$

where f denotes a neural network, such as a feed-forward or convolutional network.

Diffusion Models

Diffusion models (Ho, Jain, and Abbeel 2020) are used to generate high-quality samples from complex data distributions through a bi-directional process.

The forward process of diffusion model is a Markov Chain that gradually adds Gaussian noise to x^0 :

$$q(x^{1:T} | x^0) = \prod_{t=1}^T q(x^t | x^{t-1}), \quad (3)$$

$$q(x^t | x^{t-1}) = \mathcal{N}(x^t; \sqrt{1 - \beta_t} x^{t-1}, \beta_t I), \quad (4)$$

where T is the timesteps of adding noise and $\beta_t \in (0, 1)$ is the variance schedule.

For the denoising process, the goal is to learn a conditional probability distribution $p_\theta(x^{t-1} | x^t)$ that reverses the diffusion process:

$$p_\theta(x^{t-1} | x^t) = \mathcal{N}(x^{t-1}; \mu_\theta(x^t, t), \sigma_\theta(x^t, t)I), \quad (5)$$

where the parameter θ can be optimized by minimizing the negative log-likelihood via an Evidence Lower Bound (ELBO):

$$\min_{\theta} \mathbb{E}_{q(x^0)} \leq \min_{\theta} \mathbb{E}_{q(x^0:K)} L \quad (6)$$

$$L = -\log p(x^K) - \sum_{k=1}^K \log \frac{p_\theta(x^{k-1} | x^k)}{q(x^k | x^{k-1})} \quad (7)$$

Our Method: CasFT

Our proposed CasFT can be summarized as a three-step model: the first step involves extracting spatiotemporal patterns from the observed cascade, the second step simulates the future trends of popularity increment, and the third step makes the final prediction. The overall framework is shown in Fig. 2. We discuss the key components below.

Historical Spatiotemporal Modeling

Given an observed cascade $\mathcal{C}(t_o)$, CasFT first extracts the spatiotemporal features from it. Specifically, this step involves two components: (1) spatial structural pattern extraction from the observed cascade graph $G(t_o)$ and the global graph \mathcal{G} , and (2) temporal pattern extraction based on the cascade sequence. The details are outlined as follows.

Spatial Structural Pattern Extraction: In light of the distinct characteristics between the cascade and global graphs, we employ two different graph embedding methods to extract unique structural patterns from each, respectively. For the cascade graph, we use GraphWave (Donnat et al. 2018) to capture local structural information, as referenced in previous studies (Zhou et al. 2020b; Xu et al. 2021). For the large-scale global graph, we employ NetSMF (Qiu et al. 2019), which is designed for large-scale networks and can quickly learn interactions between all users, efficiently embedding these global structural patterns into node embeddings. Specifically, given the observed cascade graph $G(t_o)$ and the global graph \mathcal{G} , we calculate the local-view and global-view representations of users as follows:

$$E_c = \text{GraphWave}(G(t_o)), \quad (8)$$

$$E_g = \text{NetSMF}(\mathcal{G}), \quad (9)$$

where E_c and E_g represent local-view and global-view representations for users in $G(t_o)$ and \mathcal{G} , respectively. For a specific user u_i , we retrieve its representations from these two views according to its index, denoted as $E_c(u_i)$ and $E_g(u_i)$.

Temporal Pattern Extraction: In this phase, we split the cascade sequence $S(t_o)$ into multiple sub-sequences in chronological order, represented as $\mathcal{S} = \{S_j\}_{j \in N} = \{(u_0), (u_0, u_1), \dots, (u_0, u_1, \dots, u_N)\}$, and employ the self-attention mechanism (Vaswani et al. 2017) to each sub-sequence to capture its internal interdependencies, thereby accounting for the long-term history information. Firstly, for each timestamp t_i , we define a temporal encoding procedure by trigonometric functions:

$$[\mathbf{z}(t_i)]_j = \begin{cases} \cos(t_i/10000^{\frac{j-1}{B}}), & \text{if } j \text{ is odd,} \\ \sin(t_i/10000^{\frac{j}{B}}), & \text{if } j \text{ is even.} \end{cases} \quad (10)$$

We deterministically compute $\mathbf{z}(t_i) \in \mathbb{R}^B$, where B is the dimension of encoding and get \mathbf{Z}_t which is the set of $\mathbf{z}(t_i)$.

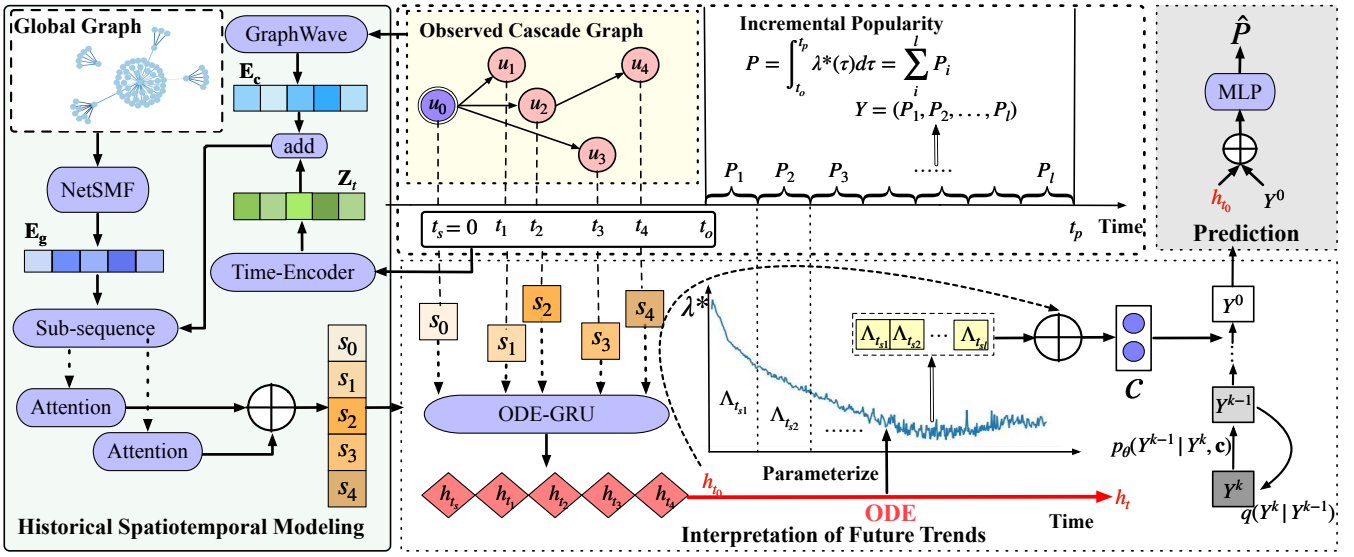


Figure 2: An overview of our proposed model CasFT.

The initial embedding of each sub-sequence S_j is specified by:

$$\mathbf{x}(i) = \mathbf{z}(t_i) + E_c(u_i), \quad (11)$$

$$E_c^j = \{\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(i)\}, \quad (12)$$

$$E_g^j = \{E_g(u_0), E_g(u_1), \dots, E_g(u_i)\}, \quad (13)$$

where $u_i \in S_j$, and t_i is its associated time stamp. We then encode E_c^j and E_g^j via two independent self-attention layers. Specifically, the scaled dot-product attention (Vaswani et al. 2017) is defined as:

$$\mathbf{s} = \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad (14)$$

where Q, K, V represent queries, keys, and values. In our case, the self-attention operation takes E_c^j or E_g^j as input and converts them into different groups of Q, K , and V through separate linear transformations. By applying the above self-attention operation to all sub-sequences, we obtain sets of hidden representations $\mathbf{s}_c = \{\mathbf{s}_c^0, \mathbf{s}_c^1, \dots, \mathbf{s}_c^N\}$ and $\mathbf{s}_g = \{\mathbf{s}_g^0, \mathbf{s}_g^1, \dots, \mathbf{s}_g^N\}$ for the local-view and global-view, respectively. Finally, we concatenate these two views of hidden representations at the sub-sequence level to form the final historical spatiotemporal features of \mathcal{S} , denoted by $\{\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_N\}$.

Interpretation of Future Trends

As we discussed in the Introduction section, the real dynamics of cascade evolution between the observation and prediction periods are invisible and difficult to depict due to their fluctuations. To address this challenge, we designed a dynamic cues-driven diffusion model to simulate future trends based solely on the observed cascades. Specifically, this model consists of two components: (1) a dynamic cues learning module and (2) a diffusion model-based future trends simulation module. To acquire the dynamic cues, we are inspired by ODEs, which are used to model quantity

changes over time in dynamic systems. We introduce a neural ODEs to model the continuous-time growth rate during the observation period, capturing its evolving tendency. Afterward, the future growth rate between the observation and prediction periods can be autoregressively predicted using this neural ODEs. Finally, the dynamic cues of popularity increment are derived through an integration operation based on these predicted growth rates. However, one limitation of ODEs is that they are constrained by the initial state of the dynamic system, leading to popularity predictions based on the estimated growth rate that are biased and lack realistic variability. Accordingly, we further divide the estimated period into several time spans and employ diffusion models to generate incremental popularity in each time span, conditioned on the growth rate.

Dynamic Cues Learning with Neural ODEs: We leverage neural ODEs to model the continuous-time growth rate dynamics with a vector representation \mathbf{h}_{t_i} at every timestamp t_i . Here we parameterize the growth rate using hidden state dynamics:

$$\lambda^*(t) = f_1(\mathbf{h}_t), \quad (15)$$

where f is a linear neural network with the softplus function to ensure the output is positive. As for the continuous-time hidden state \mathbf{h}_{t_i} , we use a multi-layer fully connected neural network f_2 to model the continuous change in the form of an ODE. When a new retweeting action occurs at time t_i , we use a GRU function g to model instantaneous changes triggered by a newly observed event:

$$\frac{d\mathbf{h}_{t_i}}{dt} = f_2(t, \mathbf{h}_{t_{i-1}}), \quad (16)$$

$$\mathbf{h}'_{t_i} = \text{ODESolve}(f_2, \mathbf{h}_{t_{i-1}}, (t_{i-1}, t_i)), \quad (17)$$

$$\mathbf{h}_{t_i} = g(\mathbf{h}'_{t_i}, \mathbf{s}_i), \quad (18)$$

and the popularity at time t_i can be calculated as:

$$P_{t_i} = \int_0^{t_i} \lambda^*(\tau) d\tau = \Lambda_{t_i}. \quad (19)$$

Then we can get the hidden state, growth rate, and cumulative popularity at an arbitrary time by the above equations, even after observation time.

Future Trends Simulation with Diffusion Model: We first divide the time between the observation and prediction time into l uniform time intervals, denoted as $\{(t_o, t_{s1}), \dots, (t_{si-1}, t_{si}), \dots, (t_{sl-1}, t_{sl})\}$, where $i \in l$. The start time is t_o and the end time t_{sl} equals to t_p . We aim to model the incremental popularity in each time interval and define $Y = (P_1, P_2, \dots, P_l)$ as the sequential increase process of popularity. y_i represents the popularity during the time interval (t_{i-1}, t_i) .

We compute the hidden state \mathbf{h}_{t_o} at the exact observation time, and the dynamic cues $\{\Lambda_{t_{s1}}, \dots, \Lambda_{t_{sl}}\}$ by using Eq. (16) and (19), respectively. We then concatenate them together to form a new guide vector \mathbf{c} . Conditioned on \mathbf{c} , we utilize the diffusion model to approximate the real distribution of the incremental popularity sequence Y . Specifically, we characterize the diffusion operation of Y as a Markov process (Y^0, Y^1, \dots, Y^K) to model the potential changes of incremental popularity over the prediction window, with T denoting the number of diffusion steps. In the forward process, Gaussian noise is added to Y^0 step by step until it is corrupted into pure Gaussian noise, which is formulated as follows:

$$q(Y^k|Y^{k-1}) = \mathcal{N}(Y^k; \sqrt{1 - \beta_k}Y^{k-1}, \beta_k I). \quad (20)$$

During the reverse diffusion process, we iteratively reconstruct the incremental popularity sequence Y^0 for the final prediction. Specifically, the denoising process does not only depend on the representation obtained in the previous step but also on the significant cue \mathbf{c} serving as a condition, which is formulated as follows:

$$p_\theta(Y^{0:K}|\mathbf{c}) := p(Y^K) \prod_{k=1}^K p_\theta(Y^{k-1}|Y^k, \mathbf{c}), \quad (21)$$

In our implementation, we adopt a fully connected layer to parameterize the denoising process and use Denoising Diffusion Implicit Models (DDIM) (Song, Meng, and Ermon 2020) to generate segmented popularity sequence Y .

Prediction & Optimization

Prediction. After obtaining both the hidden state \mathbf{h}_{t_o} at observation time t_o and the generated segmented popularity sequence Y^0 , we concatenate and feed them into an MLP to make the final cascade popularity prediction:

$$\hat{P} = \text{Softplus}(\text{MLP}([Y^0, \mathbf{h}_{t_o}])) \quad (22)$$

where the softplus activation function is to ensure the predicted popularity is positive.

Optimization. The training objective of CasFT can be decomposed into a regression loss and a generative loss. As for the regression part, the loss function is defined as:

$$\mathcal{L}_1 = \frac{1}{M} \sum_{k=1}^M (\log_2(P + 1) - \log_2(\hat{P} + 1))^2, \quad (23)$$

Dataset	Twitter	APS	Weibo
Cascades	86,764	207,685	119,313
Avg. popularity	94	51	240
<i>Number of cascades in two observation settings</i>			
Train(1d/3y/0.5h)	7,308	18,511	21,463
Val(1d/3y/0.5h)	1,566	3,967	4,599
Test(1d/3y/0.5h)	1,566	3,966	4,599
Train(2d/5y/1h)	10,983	32,102	29,908
Val(2d/5y/1h)	2,353	6,879	6,409
Test(2d/5y/1h)	2,353	6,879	6,408

Table 1: Statistics of the three datasets.

where P and \hat{P} is the ground-truth and the predicted popularity, respectively. For the generative part, we minimize the following negative log-likelihood:

$$\mathcal{L}_2 = - \sum_{k=1}^M \log p_\theta(Y_k^0|c_k). \quad (24)$$

To sum up, the final loss is defined as:

$$\mathcal{L} = \mathcal{L}_1 + \gamma \mathcal{L}_2, \quad (25)$$

where γ is a hyperparameter used to adjust the trade-off between two losses.

Experiments

In this section, we first present our benchmark datasets and then evaluate our model CasFT¹ against state-of-the-art baselines in information cascade popularity prediction task to answer the following questions:

RQ1: Compared to state-of-the-art baselines, can our approach achieve a more accurate prediction of cascade popularity?

RQ2: What are the benefits of employing neural ODEs to model the growth rate? How much does it contribute to performance improvement?

RQ3: Why do we need diffusion models for enhancing the modeling of future trend dynamics? In contrast, how does the utilization of simpler models capture this characteristic?

RQ4: What is the impact of the hyperparameters, including the number of segmented periods, the choice of ODE-solver, the hidden dimension, and the diffusion steps?

Datasets and Preprocessing

We conducted experiments on three real-world datasets, including Twitter, APS (Shen et al. 2014), and Sina Weibo (Cao et al. 2017). The details of these datasets are as follows:

- **Twitter:** We collected tweets posted between March 1 and April 15, 2022, constructing the cascades based on the hashtag, where original tweets are posted between March 1 and March 31.
- **American Physical Society (APS):** This dataset includes papers published in APS journals between 1893 and 1997, where each paper and its citations form a citation cascade.

¹<https://github.com/UM-Data-Intelligence-Lab/CasFT>

Method	Twitter				APS				Weibo			
	1 day		2 days		3 years		5 years		0.5 hours		1 hour	
	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
Feature-based	7.8268	0.7073	6.5154	0.6514	1.9881	0.3085	1.9696	0.3193	4.0788	0.4094	3.6380	0.4268
SEISMIC	10.687	0.9689	8.1851	0.8147	2.0583	0.3013	2.3013	0.4320	5.0300	0.4819	4.0594	0.5003
DeepCas	6.3297	0.6566	5.7146	0.6671	2.1051	0.2869	1.9260	0.3458	4.6460	0.3258	3.5532	0.3532
DeepHawkes	5.9341	0.5017	4.8489	0.5189	1.9142	0.2823	1.8145	0.3368	2.8741	0.3041	2.7434	0.3346
CasCN	5.8742	0.4894	4.7154	0.4974	1.8930	0.2763	1.7494	0.3208	2.7931	0.2940	2.6831	0.3255
VaCas	5.5124	0.4796	4.2147	0.4871	1.7764	0.2697	1.6945	0.3012	2.5246	0.2847	2.3451	0.2997
CasFlow	4.7799	0.4150	3.6888	0.4222	1.4370	0.2401	1.3346	0.2624	2.3370	0.2665	2.2232	0.2949
CTCP	5.3991	0.3757	3.6016	0.3773	1.7676	0.3054	1.3751	0.2908	2.5572	0.3056	2.2968	0.3010
CasFT	3.8546	0.3674	3.4496	0.3605	1.2468	0.2282	1.1748	0.2561	2.1728	0.2448	2.0655	0.2695
(Improve)	19.36%↑	2.21%↑	4.22%↑	4.45%↑	13.24%↑	4.96%↑	11.97%↑	2.40%↑	7.02%↑	8.14%↑	10.07%↑	8.61%↑

Table 2: Performance comparison between baselines and CasFT on three datasets across different observation times measured by MSLE, MAPE (lower is better).

Method	Twitter				APS				Weibo			
	1 day		2 days		3 years		5 years		0.5 hours		1 hour	
	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
CasFT-w/o FT	4.8641	0.3964	4.1154	0.4531	1.4147	0.2504	1.3544	0.2805	2.3964	0.2745	2.3412	0.2933
CasFT-w/o ODE	4.3115	0.3902	3.7468	0.4174	1.3241	0.2473	1.2366	0.2778	2.3054	0.2601	2.2247	0.2794
CasFT-w/o Diffusion	4.2645	0.3887	3.6618	0.4037	1.3258	0.2466	1.2364	0.2776	2.3312	0.2631	2.2644	0.2819
CasFT-FM	4.5334	0.3875	4.0047	0.4329	1.3847	0.2495	1.3017	0.2798	2.3645	0.2681	2.3015	0.2884
CasFT	3.8546	0.3674	3.4496	0.3605	1.2468	0.2282	1.1748	0.2561	2.1728	0.2448	2.0655	0.2695

Table 3: Performance comparison between CasFT and CasFT-variants on three datasets under two observation times measured by MSLE, MAPE (lower is better).

- *Sina Weibo*: Weibo is the largest Chinese microblogging platform, where each original microblog post and subsequent reposts can form a repost cascade.

Following previous works (Xu et al. 2021; Lu et al. 2023), we set observation time as 1 day and 2 days for the Twitter dataset, and 3 years and 5 years for the APS dataset, for Weibo, the observation times are set to 0.5 hours and 1 hour. Furthermore, prediction time is set to 15 days for Twitter, 20 years for APS, and 24 hours for Weibo. In addition, we filter out cascades with fewer than 10 participants during the observation periods. For all datasets, 70% of the data is used for training, 15 % for validation, and 15% for testing. Detailed information about the three datasets is provided in Table 1.

Baselines

We compare CasFT against a sizeable collection of state-of-the-art baselines: **Feature-based** methods extract key features (cascade size, temporal intervals, etc.) and use an MLP model for prediction. **SEISMIC** (Zhao et al. 2015) designs a statistical model based on the theory of self-excited point processes. **DeepCas** (Li et al. 2017) represents cascades as random walk paths and uses a bi-directional GRU with attention for effective modeling and prediction. **DeepHawkes** (Cao et al. 2017) integrates the Hawkes process and deep learning, focusing on user impact, self-excitation, and time decay for cascade modeling. **CasCN** (Chen et al. 2019) adopts a novel multi-directional/dynamic GNN. **VaCas** (Zhou et al. 2020b) devises a hierarchical graph learning method and leverages a variational autoencoder for modeling the uncertainty. **CasFlow** (Xu et al. 2021) mainly considers the effects of local and global graphs to represent user be-

havior for predicting popularity. **CTCP** (Lu et al. 2023) proposes an evolution learning module that updates the states of users and cascades in real time as diffusion behaviors occur.

Evaluation Metrics

Two commonly used metrics, mean squared logarithmic error (MSLE) and mean absolute percentage error (MAPE), are employed to evaluate the performance of models:

$$MSLE = \frac{1}{M} \sum_{k=1}^M (\log_2(P+1) - \log_2(\hat{P}+1))^2, \quad (26)$$

$$MAPE = \frac{1}{M} \sum_{k=1}^M \frac{|\log_2(P+2) - \log_2(\hat{P}+2)|}{\log_2(P+2)}, \quad (27)$$

where P represents the true popularity, \hat{P} denotes the predicted popularity, and M represents the number of cascades. The operations '+1' and '+2' are used for scaling to avoid potential zero values in the denominator or logarithmic operations.

Performance Comparison (RQ1)

Table 2 shows the overall performance of the three datasets. We observe that our method CasFT significantly outperforms all baselines on MSLE and MAPE. For example, compared to the best-performing baselines, our CasFT achieves a 4.2%-19.3% improvement of MSLE, and a 2.2%-8.6% improvement of MAPE, on the three datasets with two different observation times. We see that CasFT demonstrates notable enhancement, especially on the Twitter dataset. This can be

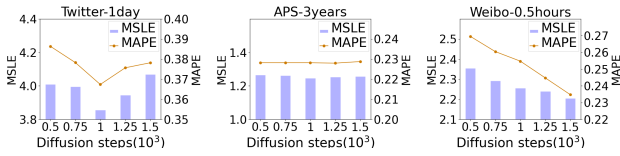


Figure 3: Impact of the diffusion steps.

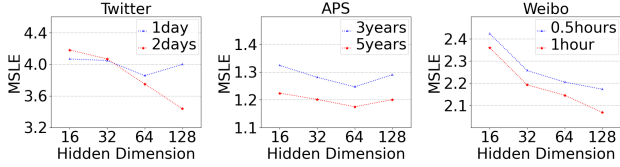


Figure 4: Impact of the hidden dimension.

attributed to the fact that cascades on Twitter often lack intricate multi-level forwarding information, resulting in a relatively simpler structure of the cascade graph compared to the Weibo and APS datasets while previous research predominantly concentrates on the evolution of dynamic graphs, underscoring the versatility of our proposed method, CasFT, in effectively handling various types of cascade graphs.

Ablation Study (RQ2 & RQ3)

To answer RQ2 and RQ3, we have conducted a series of experiments where we introduced different variants of our CasFT model. We conduct the ablation study to investigate the contribution of each component and develop four variants: 1) **CasFT-w/o FT**. We remove the future trend modeling module and only use the spatiotemporal features \mathcal{S} for prediction; 2) **CasFT-w/o ODE**. We take the spatiotemporal features \mathcal{S} as the condition of the later diffusion models, without parameterizing the growth rate; 3) **CasFT-w/o Diffusion**. We remove the diffusion block and directly input the condition c into an MLP; and 4) **CasFT-FM**. We just use an MLP to replace our future trend modules, predict both the segmented popularity and the incremental popularity during (t_o, t_p) , and also take the predicted segmented popularity sequence as a significant cue.

The results and comparison of these variants are shown in Table 3. The comparison of CasFT over CasFT-w/o FT validates the necessity of modeling the future trend of cascade with an improvement of up to 20.75%. Through a comparative analysis of CasFT-w/o FT, CasFT-w/o ODE, and CasFT-w/o Diffusion, it becomes evident that modeling the growth rate and segmented popularity generation both facilitate prediction accuracy, achieving improvements of 20.75%, 13.63%, and 10.70% respectively. We also designed a variant CasFT-FM which outperforms CasFT-w/o FT but is worse than CasFT, showing the usefulness of our proposed future trend block.

Hyperparameters (RQ4)

We investigate the influence of hyperparameters including the hidden dimension of h_t , the choice of ODEsolvers, the number of the time interval l , and the diffusion steps. Firstly, the diffusion steps serve as a crucial parameter for diffusion models. To assess the influence of diffusion steps, we train

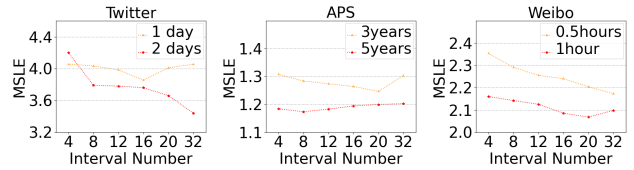


Figure 5: Impact of the interval number.

ODE_Solver	Twitter (1 day)		APS (3 years)		Weibo (0.5 hours)	
	MSLE	MAPE	MSLE	MAPE	MSLE	MAPE
bosh3	4.0933	0.3764	1.2701	0.2292	2.1792	0.2509
adaptive_heun	4.2215	0.3995	1.2702	0.2291	2.1786	0.2578
euler	4.2487	0.3989	1.2837	0.2304	2.2188	0.2598
rk4	4.2115	0.3720	1.2644	0.2263	2.1780	0.2537
implicit_adams	4.1503	0.3722	1.2644	0.2263	2.1780	0.2537
midpoint	4.0760	0.3697	1.2622	0.2280	2.1873	0.2457
dopri5	3.8546	0.3679	1.2468	0.2282	2.1728	0.2448

Table 4: Impact of different types of ODE.Solver on the performance of CasFT across the three datasets.

CasFT using varying steps, ranging from 500 to 1500. The results are shown in Figure 3 and we find that the number of diffusion steps has an impact on the Twitter dataset while the MAPE in APS and Weibo datasets tend to be stable. Additionally, among the ODEsolver options shown in Table 4, euler yields the worst performance because the error of euler’s method usually decreases as the step size decreases, while dopri5 demonstrates relatively superior predictive capabilities and is used across all of our experiments.

Moreover, the prognostication of CasFT relies on the dynamic hidden state h_t and the generated segmented popularity sequence Y^0 , prompting an investigation into the influence of the dimensionality of h_t and the interval number l of Y^0 on the performance, shown in Figure 4 and Figure 5. It is observed that both the dimension of the hidden state and the number of intervals affect the ultimate forecasting outcomes.

Conclusion

In this work, we propose CasFT, which leverages observed information Cascades and dynamic cues modeled via neural ODEs as conditions to guide the generation of Future popularity-increasing Trends through a diffusion model. The generated trends are integrated with the spatiotemporal patterns present in the observed information cascades to enhance the accuracy of popularity predictions. Experiments on three real-world information cascade datasets demonstrate the superior performance of CasFT compared to a sizeable collection of state-of-the-art baselines. CasFT significantly outperforms all the baselines, with 2.2%-19.3% improvement over the best-performing baseline methods across various datasets.

Acknowledgements

This project has received funding from the Science and Technology Development Fund, Macau SAR (0047/2022/A1, 001/2024/SKL), and Taihu Innovation Scheme. This work was performed in part at SICC which is supported by SKL-IOTSC, University of Macau.

References

- Bao, P.; Shen, H.-W.; Jin, X.; and Cheng, X.-Q. 2015. Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes. In *WWW*, 9–10.
- Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; and Cheng, X. 2017. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *CIKM*, 1149–1158.
- Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.
- Chen, X.; Zhou, F.; Zhang, K.; Trajcevski, G.; Zhong, T.; and Zhang, F. 2019. Information Diffusion Prediction via Recurrent Cascades Convolution. In *ICDE*, 770–781. IEEE.
- Cheng, J.; Adamic, L.; Dow, P. A.; Kleinberg, J. M.; and Leskovec, J. 2014. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, 925–936.
- Donnat, C.; Zitnik, M.; Hallac, D.; and Leskovec, J. 2018. Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1320–1329.
- Gao, S.; Ma, J.; and Chen, Z. 2014. Effective and effortless features for popularity prediction in microblogging network. In *WWW*, 269–270.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.
- Lazer, D. M.; Baum, M. A.; Benkler, Y.; Berinsky, A. J.; Greenhill, K. M.; Menczer, F.; Metzger, M. J.; Nyhan, B.; Pennycook, G.; Rothschild, D.; et al. 2018. The science of fake news. *Science*, 359(6380): 1094–1096.
- Lee, J. G.; Moon, S.; and Salamatian, K. 2012. Modeling and predicting the popularity of online contents with Cox proportional hazard regression model. *Neurocomputing*, 76(1): 134–145.
- Li, C.; Ma, J.; Guo, X.; and Mei, Q. 2017. Deepcas: An end-to-end predictor of information cascades. In *WWW*, 577–586.
- Li, G.; Chen, S.; Feng, J.; Tan, K.-l.; and Li, W.-s. 2014. Efficient location-aware influence maximization. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 87–98.
- Lu, X.; Ji, S.; Yu, L.; Sun, L.; Du, B.; and Zhu, T. 2023. Continuous-Time Graph Learning for Cascade Popularity Prediction. *arXiv preprint arXiv:2306.03756*.
- Ma, Z.; Sun, A.; and Cong, G. 2013. On predicting the popularity of newly emerging hashtags in twitter. *Journal of the American Society for Information Science and Technology*, 64(7): 1399–1410.
- Martin, T.; Hofman, J. M.; Sharma, A.; Anderson, A.; and Watts, D. J. 2016. Exploring limits to prediction in complex social systems. In *WWW*, 683–694.
- Matsubara, Y.; Sakurai, Y.; Prakash, B. A.; Li, L.; and Faloutsos, C. 2012. Rise and fall patterns of information diffusion: model and implications. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 6–14.
- Petrovic, S.; Osborne, M.; and Lavrenko, V. 2011. Rt to win! predicting message propagation in twitter. In *Proceedings of the international AAAI conference on web and social media*, volume 5, 586–589.
- Qiu, J.; Dong, Y.; Ma, H.; Li, J.; Wang, C.; Wang, K.; and Tang, J. 2019. Netsmf: Large-scale network embedding as sparse matrix factorization. In *The World Wide Web Conference*, 1509–1520.
- Shen, H.; Wang, D.; Song, C.; and Barabási, A.-L. 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.
- Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, R.; Huang, Z.; Liu, S.; Shao, H.; Liu, D.; Li, J.; Wang, T.; Sun, D.; Yao, S.; and Abdelzaher, T. 2021. Dydiff-vae: A dynamic variational framework for information diffusion prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 163–172.
- Wang, S.; Yan, Z.; Hu, X.; Yu, P. S.; Li, Z.; and Wang, B. 2016. CPB: a classification-based approach for burst time prediction in cascades. *Knowledge and Information Systems*, 49: 243–271.
- Wang, Y.; Wang, X.; Ran, Y.; Michalski, R.; and Jia, T. 2022. CasSeqGCN: Combining network structure and temporal sequence to predict information cascades. *Expert Systems with Applications*, 206: 117693.
- Wu, B.; Mei, T.; Cheng, W.-H.; and Zhang, Y. 2016. Unfolding temporal dynamics: Predicting social media popularity using multi-scale temporal decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Xu, X.; Zhou, F.; Zhang, K.; Liu, S.; and Trajcevski, G. 2021. Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction. *IEEE Transactions on Knowledge and Data Engineering*.
- Yang, J.; and Leskovec, J. 2011. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 177–186.
- Yu, L.; Xu, X.; Trajcevski, G.; and Zhou, F. 2022. Transformer-enhanced Hawkes process with decoupling training for information cascade prediction. *Knowledge-Based Systems*, 255: 109740.
- Yu, Z.; Zhang, D.; Yu, Z.; and Yang, D. 2015. Participant selection for offline event marketing leveraging location-based social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(6): 853–864.

- Zaman, T.; Fox, E. B.; and Bradlow, E. T. 2014. A Bayesian approach for predicting the popularity of tweets.
- Zhang, X.; Aravamudan, A.; and Anagnostopoulos, G. C. 2022. Anytime Information Cascade Popularity Prediction via Self-Exciting Processes. In *International Conference on Machine Learning*, 26028–26047. PMLR.
- Zhao, Q.; Erdogdu, M. A.; He, H. Y.; Rajaraman, A.; and Leskovec, J. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1513–1522.
- Zhou, F.; Jing, X.; Xu, X.; Zhong, T.; Trajcevski, G.; and Wu, J. 2020a. Continual information cascade learning. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, 1–6. IEEE.
- Zhou, F.; Xu, X.; Zhang, K.; Trajcevski, G.; and Zhong, T. 2020b. Variational information diffusion for probabilistic cascades prediction. In *IEEE INFOCOM 2020-IEEE conference on computer communications*, 1618–1627. IEEE.